On the Metatheory of Linguistics

Christian Wurm

Doktorarbeit zur Promotion

an der Universiät Bielefeld

Gutachter: Marcus Kracht, Jens Michaelis, Gregory Kobele

December 18, 2013

Contents

Chapter 1

Introduction

1.1 What is Metalinguistics?

If we should define the goal of the science of language concisely in a sentence, most scholars would say something like: it is the study of our implicit, unconscious knowledge of language. We will see that this statement is very problematic; and most of this first chapter will be devoted to show why it is problematic. We start with some basic, uncritical observations.[1]

The subject of linguistics consists the study of languages and of language. Study of languages means that linguists have to look at languages, which are their primary object of study. Study of language means that what is interesting to linguists are in particular general properties of all languages, rather than properties of particular languages. In any way, linguistics is based on the observation of ourselves as a species, because after all we are interested in the "verbal behavior" of humans. This is maybe the main point which distinguishes it from a science like physics. What distinguishes it from a science like psychology is mostly the following: linguistics is typically not about what we actually say in given circumstances, but about what we *can* say. It does not describe our actual behavior, but rather our possible behavior. There might not be a complete agreement on this point, still it seems to guide theoretical linguistics in its current practice.[2] So in the sequel, I take it for granted that if we are to describe language, what we essentially do is to provide rules for well-formed utterances, rather than providing rules which prescribe what we have to say in a given circumstance. Linguistic rules are thus rules for possible behaviors, not for actual behavior. Having said this, it is exactly this *intensional* character which distinguishes linguistics from most of psychology, though of course not from all of it.

A next distinguishing hallmark is the fact that language is one of the main fields of human *creativity*: there is no upper bound to the number of utterances we can make. There are two main arguments for this claim: firstly, the old Chomskyan argument that given any sentence (say the presumably longest sentence of my finite language of English), I can construct a new, longer sentence (say by means of conjunction), which is again English. This is quite convincing, though not strictly empirical, because it already presupposes an abstract notion of "any sentence", which is not an empirical notion or object. The second argument[3] is based on the frequency distribution of our observations: as a matter of fact, most sentences we observe, we observe only once. If we would have observed a considerable portion of the language in question, this would be an extremely improbable distribution; but it is very plausible under the assumption that we

[1] What is to follow can be read as the outlines of a theory of the science of language. It falls in this sense under the general field of theory of science, as exemplified e.g. by Kevin Kelly, [33]. However, as I lay out in the sequel, the peculiarities of linguistics seem to outweigh the common ground with the general theory of science, at least for the aspects I am focussing on.

[2] This is surely not the place for a complete discussion of this point. So let me just say: this position is not necessarily the correct one, but it seems to me the "working assumption" of theoretical linguistics in the canonical sense. Moreover, to me it seems to be generally unclear what theoretical linguistics would actually look like if we would think of it as a science predicting verbal behavior in given circumstances. In my view, this depends on a lot of things: for example, if we want to reconstruct canonical semantics in this view, I guess we first need a good theory of communication, in particular a notion of what successful communication means. As this and similar questions are complex and mostly open, I will just assume the standard "working assumption" of linguistics being on *possible* rather than actual behavior.

[3] Put forward by Alexander Clark, as far as I know.

have only observed a small fragment of the complete language. As there is not the slightest evidence that this changes with a growing number of observations we make, this points us towards the fact that the number of possible observations is infinite. This argument is more empirical than the Chomskyan; yet it does not exclude the fact that languages are extremely large yet finite.

Despite these problems, by now all linguists agree that linguistic descriptions have to account for linguistic creativity. This is the main ingredient of our problem, which is the following: we can of course observe linguistic behavior in humans, but as languages are infinite objects, we can only observe finite fragments. From a formal language-theoretic point of view, finite fragments tell us *a priori* (that is, without further assumption) quite little about infinite languages, namely exactly nothing beyond the fact that the language has a certain finite subset. So we quickly come to the conclusion (we will also make this argument in much more detail in the sequel) that the infinite dimension of language remains subject to our stipulation. So as linguists, we first have to construct an infinite language before we can describe it. This is a very conscious process, because as linguists, it is not our implicit knowledge of language that counts but rather the explicit, conscious knowledge. But this obviously conflicts with the claim that our main goal is to describe an implicit, unconscious knowledge. It is important to be clear about this point: even if I (implicitly) know an infinite language as a speaker, as a linguist I do *not* know it in a way such that I can describe it. I can take parts of my implicit linguistic knowledge and make them explicit by using my intuition, but these parts will always be finite! As a linguist, I will never find an infinite language as a given, empirical object, I always have to construct it. How can we know that our conscious construction of language coincides with the implicit knowledge? Well, we simply do not know. So the commitment to describe linguistic knowledge and creativity conflicts with the claim that it should be unconscious and implicit.

As is easy to see, there is some similarity between the child learning a language, and the linguist constructing it: both construct an infinite language from a finite amount of data. But whereas the child can in the end claim to know the language (implicitly)[4] by the very definition of what language is, for the linguist, even if he knows the language implicitly, this is of no use, because what counts is what he knows *explicitly*; and for his explicit knowledge, there is no way to ever tell whether he has constructed the correct language! So in this sense, his situation is worse.[5] Anyway, here and in the sequel, if we talk about observation etc., we *always* take the perspective of the linguist rather than the child learner, and we urge the reader to be aware of this rather unusual perspective.

Now of course, in linguistics, nobody would ever think that the finite fragments of language we observe (that is, the linguistic observations we have made *as linguists*) are uninformative about the language as an infinite object, and this for several reasons: maybe the strongest argument against this view is that the child learner has to learn from the data he observes, and he will learn the language in a way which is determined by his observations. Nonetheless, there is no *a priori* reason we cannot claim a thing like: "The true pattern of language does only reveal itself in sentences with more than 300million words; everything below

[4] Although in fact there are good arguments why even this is problematic, see [16]

[5] In another sense, as we will see, his situation is also much better.

is quite arbitrary." This makes perfect sense from the point of view of formal language theory.[6] But from the point of view of a linguist, this would seem to make the entire enterprise of linguistics ridiculous. The reason is: in linguistics we always construct language on the basis of the finite fragments we observe; if we do not rely on *them*, on what are we supposed to rely? But that is not a linguistic argument, but rather a methodological, metalinguistic one. And regarding the argument of learnability which has to be ensured, there is a simple, well-known answer: it is only below the mentioned threshold of 300million words that children even need to learn - beyond this threshold everything is innate!

So linguists have to rely on finite fragments they observe; they are bound to the commitment that these fragments reveal the nature of language both in a positive and negative sense: positively: the fragments we observe are informative about the infinitary nature of language, and negatively: all that is informative about the infinitary nature is in the fragment we observe. So the infinite languages linguists construct are interpretations or *projections* of the fragments they observe. But of course, finite fragments can be interpreted in many ways, and the projection we perform depends heavily on the theoretical devices we use, more bluntly: the shape of infinite language, as we construe it, depends on linguistic tools and theories we use.

But if language as an object depends on the theory we make of it, in how far can we make valid general statements on the formal properties of language? These seem to be circular by necessity! This is in a word the problem I will address in this work, and in fact I will argue that *a priori* we can hardly make any claims on the nature of language just by observation, but only by making in addition some pre-theoretical assumptions. Still, there are good arguments which favor some pre-theoretical assumptions over others, and the matter turns out to be rich and interesting. This is what I will study in this work, under the label *linguistic metatheory*.

What is linguistic metatheory, or the metatheory of language? In a word, we can say that in the same way as metamathematics is the theory of mathematical reasoning, metalinguistics is the theory of linguistic reasoning. What is linguistic reasoning? In a word, it is the thinking about what is part of our language (semantically: which utterance has which meaning), beyond our immediate intuition. By way of analogy, mathematical reasoning consists in deriving certain consequences from premises; that is, infer the truth of a statement from the truth of other statements. Linguistic reasoning, as we conceive of it, is inferring a certain infinite language from a given finite set of data. The inferred language will for theoretical reasons always be infinite; the dataset given to us will for practical reasons always be finite. This is not a matter of learning or corpus linguistics at all: even the most armchair linguist, trying to write a (fragment of a) grammar for a language he is a native speaker of, will always only consider only a finite set of utterances, before he can write a grammar; but the grammar will have to cover infinitely many sentences, otherwise the (armchair) linguist will consider his work to be idle. So linguistic reasoning consists in inferences of the following form: we have some sentences in our language L, more formally, we have a set of premises

[6] And in fact, Chomsky himself takes a related point of view when he says that the "perfect regularity" of language is visible only when we look beyond the language we use, moving towards the infinite, see [5]

$$\{\vdash \vec{w} \in L : \vec{w} \text{ is in our dataset}\}; \tag{1.1}$$

and from this we make inferences roughly of the form

$$\frac{\vdash \vec{w} \in L \quad \vec{w} \text{ is similar to } \vec{v}}{\vdash \vec{v} \in L}. \tag{1.2}$$

So we have some means of deducing linguistic judgments from linguistic judgments. The precise form of linguistic inferences we will consider later on in much more detail. The resulting language will then simply be the closure under deduction of the dataset we have. This is all very general: but at a certain point, the linguist will have to decide how the infinite language should look like, given the data he has considered. This is what we call linguistic reasoning; and this is what we will study here.

In which way shall we study linguistic reasoning in this work? Again, a look at mathematics might be helpful. Every profane mathematician, doing profane mathematics as calculus, uses mathematical logic, even though mostly implicitly. For example, he might say: "I can show that 'p and q' is true; therefore, in particular p is true". This is a line of reasoning which seems to be unsuspicious. he might as well say: "I can show that 'either p or q' is true; and I can show that p is not true. Therefore, q must be true." This is a line of reasoning which will also seem unsuspicious to many mathematicians. What is it that the metamathematician will do? He might say: your first line of reasoning is fine, this seems to be pure logic; however, your second line is not wrong, but depends on certain metaphysical assumptions you make: for example, let us look at the quantum universe, where "p is true" means as much as "p can be verified in some physical system". Now there are cases, where we can verify: "'p or q' is true", for example in the following case: assume we look at a photon ϕ moving towards a surface, and p means: "ϕ will cross the surface in the square interval α_1", q means "ϕ will cross the surface in the square interval α_2". Choosing α_1, α_2 appropriately, we might be able to establish that "p or q is true", that is, we can verify it in the system under observation.This is because *one* measurement can confirm that one of p or q must be true. Furthermore, having chosen α_1, α_2 appropriately, it might happen that we *cannot* verify whether "p is true", nor can we verify whether "q is true", because they already are so small that the path of ϕ cannot be determined as exactly by the uncertainty principle. So setting up negation appropriately, we have "p or q are true", yet "p is not true" and "q is not true". We can now ask again: given that "p or q is true", and "p is not true", does it follow that "q is true"? In the quantum universe obviously not, because it might well be that neither we can verify q! So the line of reasoning which the mathematician doing calculus applied is not valid for the quantum universe. His reasoning is valid under certain ontological assumptions, but not under all. So, what the metamathematician does is: he uncovers the implicit metaphysical assumptions, which have to be made in order to allow for certain inferences and certain methods of mathematical reasoning.

In this way, the profane mathematician applies many different arguments he considers to be valid. In the next step, the metamathematician will try to make them fully explicit by finding find some enumeration of all valid arguments. And as a third point, having made this explicit characterization, he will try to point out how different ways of mathematical reasoning affect actual mathematics.

This is roughly what mathematical logic is about. Its goal is as this: we want to find a position which 1. is both well-founded from a metaphysical and ontological point of view, which can 2. be sufficiently formalized, that is, allows an enumeration of all valid arguments, and which 3. in addition is working well from a practical point of view: our valid arguments should give rise to a rich and interesting mathematics.

What does the metalinguist do? First, we will give some examples for linguistic reasoning, and show why some lines of reasoning are more problematic than others. Let us consider the linguist writing some grammar fragment. For example, he might say: "I can say: Peter is in love with Sally. But I can also say: Peter is in love with Sally and Mary., and that is as good. Also: Peter is in love with Sally and Mary and Gina." The linguist might do this up to a certain point, and conclude: "If my language contains Peter is in love with X., where X is any conjunction of names, then it also contains Peter is in love with X and Y., where Y is a name. This is still problematic, as the linguist presupposes to know what a name is, but if we grant him this knowledge, we should grant him the conclusion. His main argument is as follows: "In principle, there is no upper bound to the examples I can consider; for any example I can think of, I clearly judge it to be in the language. Therefore, only practical restrictions prevent me from effectively proving the infinity which I have to stipulate."

We grant him this; but most probably not all of his inferences will be as neat. Just consider the following line of reasoning: "I can say People see. I can also say People people see, see. [Now things get tricky!] In principle, I could also say: People people people see, see, see.; the fact that I do not say nor understand it under normal circumstances is due to my restricted amount of working memory, not to my knowledge of language." Now, if we grant him this point, he will be able to make the same argument as before, which we recognized to be valid. The question is: should we also accept the other argument, that his incapability to utter and understand a sentence like People people people see, see, see. has nothing to do with his language in a proper sense? Linguists usually do, but the metalinguist has to ask: well, but on what grounds? Obviously, this has to do with the fact that it is possible to extend the structure at least once; but note that this is a much weaker criterion than the first, where we could extend it arbitrarily.

Let us consider a third inference. Our linguist might argue: "War in Vietnam or no war in Vietnam, my son is gonna join the army. Now, is it possible to say: War in Vietnam or no war in Vietnam or no war in Vietnam or no war in Vietnam, my son is gonna join the army.? (For more examples and discussion, see [35], [48]) It is doubtful whether this sentence would be accepted by any speaker. Pondering about this sentence, we can somehow make sense of this, at least syntactically. The fact that we probably cannot assign any meaningful interpretation to it should not bother us, for the same holds for a sentence as At night it is colder than outside. So, is the fact that we do not understand the former sentence a restriction which is essential to knowledge or language, or due to language external factors? In the last inference, we said that there is a pattern which applies at least twice; in this case, we do not have this argument. Should we allow the inference nonetheless? We do not know, and in this case linguists seem to be generally unsure as well.

We thus acknowledge basic facts on natural language, which we will consider in much more detail later on: 1. given some finite dataset, it is generally unclear

how the corresponding infinite language does look like; 2. not all constructions, which *can* be projected to the infinite, have the same status with respect to projections; that is, projection has to be done according to different criteria in different cases. 3. whether a linguist accepts a certain sentence or not seems to depend on his *trying to make sense of it*, that is, his thinking about the sentence. This is important, as the construction of language depends on the judgments we have; but the judgments themselves are already influenced by our reasoning. 4. The same seems to hold also for normal speakers.

Note that these observations strongly contradict two essential assumptions of what we can call *naive linguistics*: the first one is: if we look at enough data, then it is entirely clear how language looks like. The second one is: there is such an object as language, which is completely determined in any regard; it is usually situated in the mind of the speaker, and all thinking and reasoning about language only spoils this *mythical* natural language. The first part of this work will be mainly dedicated to showing why naive linguistics is inadequate.

If the object of study of metalinguistics is linguistic reasoning, what are the goals of metalinguistics? We might say its main goal is to provide an explicit, formal foundation for what naive linguistics takes for granted: the existence of infinitary language. In doing so, it has to fulfill five main requirements:

1. It has to be based on datasets in a formally rigid manner; that is, we have to think of it as a computable function from finite languages to languages.

2. It should have a good mathematical and linguistic motivation for projecting certain patterns into the infinite.

3. Given the datasets we usually have in linguistics - with the usual restrictions - it has to provide languages sufficiently rich and well-structured for a satisfying linguistic theory.

4. It has to be strictly finitary in its methods. It is its goal to justify and provide the infinite objects which linguistic theory requires; but it must not take for granted the existence of any infinitary objects or methods. Finally,

5. it has to be based on reasonable philosophical assumptions on the relation of datasets and languages, and on the nature of linguistic judgments.

1.2 A Note on Syntax and Semantics

Note that we have to distinguish a purely syntactic conception of language from a more comprehensive syntactic and semantic conception. In the syntactic perspective, language consists of simple objects (less naively, languages are sets of strings or trees); in the semantic perspective, it consists of pairs (less naively, is a relation). The latter is surely more adequate a conception. Nonetheless, we will mostly stick to the former, because syntax is so much more simpler than semantics as regards both its primary objects and their decomposition. Whereas for syntax, we can modulo some idealization easily think of the basic, given objects as strings, for semantics, nothing the like seems to be at hand. Whereas for strings, the possible decompositions are trivially given, for semantic objects they are extremely unclear. So we will throughout this work simply

take the syntactic perspective, ignoring any other (semantic, phonological etc.) component of language, and moreover take for granted that the decompositions for our objects - strings - into the combinatorily relevant units - letters - are given. This is by no means mandatory, but still a reasonable assumption.

Whereas it is quite unclear what are meanings, and what are the decompositions of meanings, it nonetheless seems clear that linguistic reasoning as we conceive it applies equally well to semantics. For example, it is clear that the sentence Every boy loves some girl has two readings. But how many readings has the sentence Every girl thinks that some boy thinks that every girl thinks that some boy thinks that she is stupid?

Before we can for this sentence devise the quantifier meaning, we have to make up our mind on how many readings/meanings this sentence has (and many other sentences); otherwise, we have no means of deciding its adequacy. But in the latter example, this is clearly not a matter of intuition; there is no intuition of the form: "this sentence has 16 readings". So what we usually do is: we conversely take our primitive quantifier interpretation, in order to find an answer to the question how many readings the sentence has. But this is precisely the same problem we encountered in syntax: we need the theory in order to properly determine the data (or more clearly: to fix the data), and so all the problems from syntax come also for semantics.

So metatheory of language is *not* the metatheory of syntax. It is the metatheory of all infinite domains of language. If I mostly treat it as if it only concerned language as a syntactic object, this is because I think that in order to do otherwise, we would need much more elaborate methods and much more space.

1.3 The Goal of This Work

The goal of this work is a mathematical formalization and philosophical critique of linguistic reasoning. It goes without saying that I can only give a broad outline of this huge enterprise, and show results which only amount to showing that some things are possible in principle.

The main goal is this to give an outline of the discipline of metalinguistics, the subject of which is the construction of the subject of linguistics. It is this crucial distinction between the level of describing (infinite) languages – which is linguistics – and constructing infinite languages – which is metalinguistics – that has to be kept in mind throughout this work. Moreover, I try to show that this neat separation is not only possible, but useful and necessary if we want to avoid some pervasive problems at the very foundations of modern linguistics. I work out three different approaches to metalinguistics, which are based on different philosophical assumptions. I show that they can be motivated, rigidly formalized and give rise to interesting questions both of linguistic and metalinguistic nature. I want to stress that if one wants to do linguistics, there is no way to avoid some kind of metalinguistics. Usually, the procedure is kept implicit and is blurred with linguistics itself, but that does of course not mean that we have avoided (or even solved) the problems of metalinguistics. So what I do is twofold: I make something which is usually done implicitly explicit, pointing out possible choices, assumptions and consequences; and I make something, which is usually done in a sketchy, intuitive manner mathematically precise.

I do not want to change linguistics fundamentally, and not even in principle

endeavor to tell linguists how they should project languages to the infinite: rather, I want to show certain possibilities of formalizing the procedure. In practice, I can only give the rough outlines of the main problems and some solutions to them; I will also show that for certain problems, there are no satisfying solutions. So the main work is to give the rough outlines of a field which has been entirely neglected in linguistics so far, though in my view, it is of crucial importance for all formal approaches to natural language.

If the results of this work remain rather modest as compared to the huge endeavor or metalinguistics, I think there is one important goal which has been achieved in this work: there *are* ways to formalize and justify linguistic reasoning, and linguistic metatheory *can* be studied properly. This is of fundamental importance for formal linguistics: because it means that its arguments on natural language need not always rest on a vague and unclear notion of projection. If the reader will agree with me on that point after reading my work, then I can be content with it.

In the next chapter we will consider a bit more closely the main ingredients of the problem; these are the "creative commitment" of modern linguistics, and the "epistemologic burden", which any approach in this commitment has to carry. We will also present some presumably linguistic problems which turn out to be strongly entangled with metalinguistics. But first I will give some contextualization of my work.

1.4 The Philosophical Context...

Linguists of all schools of thought have claimed that there is a distinction between linguistics in the Cartesian tradition, assuming a mind (language faculty) which is richly structured before all experience; and linguistics in the empiricist tradition, assuming that the mind (language faculty) has the minimal structure and knowledge *a priori*, and becomes rich only through experience. Cartesian linguistics is usually identified with Chomskyan linguistics, whereas empiricists are identified with people who do not think there is a (rich) language-specific innate module. As has been pointed out by [10], this identification is not entirely correct and even misleading, as the position of empiricists and Cartesianists concerns the possibility of knowledge in the first place, which is a thing which none of the linguists arguing on innateness ever calls into question. And in fact, the old dispute between Cartesianists and empiricists seems to be more closely related to my work than to the old debate on innateness and universal grammar, because my work is concerned with a properly epistemic question: namely what can we even know about language, the object we study? But the duality which I consider most important is another one. One could say that most (almost all) of approaches to language have been based on a *metaphysical* point of view: one argues about the true nature of language, whether it is in the mind, an abstract object or whatever else. The point which I promote could be said to be *epistemological*: I am quite agnostic about the true nature of language; the question which is important to me is rather the following: what can we even know about language? So the duality which is crucial for my approach is the one between epistemology and metaphysics. One major presupposition which underlies this work is the following principle: epistemological questions have priority over metaphysical questions. This is my fundamental commitment. Of

course, one does not have to share it; still: I do not see in how far it matters whether language is in the mind or an abstract object, if I do not know how it looks like. I will elaborate on this later on, and just mention it here to contextualize my work.

1.5 ...and the Context of Learning Theory

One might say that in the end, my problem is one of learning: the linguist, as the speaker, just has to learn the language he wants to describe. Sure, learning is different in the sense that the linguist learns explicit knowledge, the speaker implicit knowledge, but the mechanisms are the same, so we can just apply learning theory to our problem. I will quickly explain why this is not the case.

The first main difference is the following: the speaker learns his language *effectively* and by definition; at some point, he knows it – under any standard definition of language. For the linguist, this does not obtain: he can always be wrong about the true nature of language. This is not a matter of empirical observation or mathematics, it is a matter of definition. This has an important consequence: whereas for the speaker, we can quietly assume some limiting procedure, which then terminates by linguistic definition at some point (when the speaker has learned his language), for the linguist that does not make sense: he never knows whether he has constructed his language correctly. So a limiting procedure to him is completely useless: he wants to do linguistics at a certain point, and therefore, he wants to terminate his metalinguistic procedure at a certain point. As this excludes all limiting procedures, we can just take the following stance: the linguist takes some finite datasets, and wants to map it onto an infinite dataset. Then he can immediately begin his proper work of grammar writing.

The second difference is the following: for the learner, it is a big open question in how far he has access to negative data (see [10]). For the linguist, things are much better: he can elicit judgments on any sentence, and can gather negative data in abundance. His problem is another one: he will get *way too much* negative data. This requires some explanation. We usually agree that there is a difference between acceptability and grammaticality, the former being an empirical notion, the latter a theoretical notion. The former is what we are (partially) given; the latter is what we want to construct. Now, the central problem is: there are many sentences we usually consider grammatical, which are *not* acceptable. This means: we will get more negative judgments than we want! So we either have to discard the notion of negative data altogether, giving them no importance whatsoever, or we have to distinguish between different sorts of negative judgments. We will take the latter road, as *some* negative data will be necessary for our purposes. This leads us to the third difference.

As we have said, a speaker learning a language succeeds by definition, whereas this does not hold for the linguist reconstructing it. Consequently, for the speaker learning from data, once he has learned his language, there is no meaningful question of the form: "did I learn the correct language?" For the metalinguist, this is a very important question: given the data, did we reconstruct the correct language? We cannot know for sure, of course, and this is very problematic: because our construction should *not* be completely arbitrary – in that case it could as well be skipped. Though we can never know which reconstruction of

language is correct, we should make our constructions at least *falsifiable*. This can be achieved as follows: given that we have some (distinguished) negative data, we can use this data to falsify the entire process of construction of the language. If the reconstructed language contains any of the (distinguished) negative data, we require that the language be constructed in another way. But this of course presupposes that the process of the construction of language itself does not have access to the negative data, otherwise we can trivially avoid any falsification! So the construction of language has to be based on positive data alone, if we want to aboid arbitrariness.

There is another consequence of the fact that the metalinguistic construction *can* be wrong, contrary to the learner. This time, it concerns the general mathematical paradigm. In learning theory, one generally departs from a class of languages and look whether it is learnable. So in a sense, we always take for granted that we *know* what is learned; and learning without this presupposition seems to be a trivial thing (actually, things are a bit more complicated, cf. [10], pp.89–95, but that is only of minor importance to us). The linguist constructing a language, on the other side, never knows whether he is correct in his construction. So for him, this process is in a sense open-ended: something should come out, but there are few criteria to decide whether the outcome is satisfying or not. This has some important consequences for this work. The first one is exactly that we need at least some negative data: if we construct an infinite language from a finite (positive) sample, then there should be a way to tell whether the construction is complete nonsense, and for this, we need some sample of utterances which should not be part of the constructed language. The second consequence is the following: to my knowledge, there have not been any studies focussing on questions on the form: given a learning algorithm A, what is the class of languages on which A converges? So one usually departs from a given class and check whether A converges for all members of this class. We will here exactly focus on questions on the former kind: given a map π from finite languages to languages, what is the class of languages π induces? What are its properties, and what are its properties given only a certain kind of input? The reason is that it is exactly this kind of question which our approach makes us ask. So not only do we differ in the techniques we use, but also in the focus of our study, just because it is the basic presupposition of this work that we really know nothing about how language really looks like, except for a finite fragment thereof.

So whereas learning theorists depart from assumptions like: "natural languages are not context-free", for us this does not hold: these statements can only be made given some projection of our data into the infinite, and this is exactly what we want to provide in the first place. This does of course not mean that considerations of complexity and expressivity are irrelevant to us: they are most interesting, as we want our formal machinery to agree with the linguists intuitions. But we are extremely open minded with respect to the outcome of our procedures!

Chapter 2

Fundamentals and Problems of Linguistic Metatheory

Summary of the Fundamental Problems

We first discuss the consequences of the commitment to describe language as an infinitary object. Whereas there is broad agreement on the commitment and no doubt about the fact that we only observe finite datasets, there is little awareness of the fundamental problem of constructing infinite languages from finite ones. We illustrate that this is not a secondary or trivial matter, by showing some invalid conclusions on the nature of language, which have arisen due to the confusion of theory and data in the projection problem. Furthermore, we show some deep problems around the projection problem, suggesting that the projection problem cannot be solved in straightforward fashion, and that questions of projection are strongly entangled with central questions of linguistic theory. If there is a reason why there is such a strong convergence on projections, then it is the habit of a certain methodology, which is very widespread, but which is questionable on a number of points. Finally, we give an outline of how, given the ontology of meta-linguistics, we can construct different adequate ontologies for linguistics.

2.1 The Creative Commitment

2.1.1 What and Where is Language?

We have already mentioned the fact that we have to assume that languages are infinite. As this is of central importance for us, we will look at it in a bit more detail.

The first question we have to address is: what is language? This is not as trivial as it seems at the first glance: in (American) structuralism, there was rather broad agreement that language is an abstraction of the collection of all utterances we observe, where abstraction is meant in the rather narrow sense of abstracting features from certain oppositions. This is so to speak an extensional definition: language is a collection of physical objects in the real world. This view, though it was never really unchallenged, was successfully attacked by Chomsky with his famous infinity argument: as there is no upper bound to the length of sentences speakers can utter, and consequently no upper bound to the number of sentences, so there are infinitely many. Any account of the extensional language, that is, the utterances we observe, will therefore be defective. In particular, it will not only be inadequate as it will not account for new, unuttered sentences, but it will be inadequate as it completely fails to capture the central aspect of language: that it is infinite. And it will also fail to capture the central aspect of the structure of language: this is only revealed if we consider that there is a finite specification of the infinite set of utterances, which is such that speakers learn their language in a finite amount of time.

The creative commitment is much weaker than Chomsky's assumptions about language, and we want to stay with this weaker, more fundamental claim: whereas Chomsky goes on to say, that linguistics needs to describe speaker's knowledge of language, and thus a cognitive capacity, we only want to use the argument to make sure: linguistics has to describe an *intensional object*; that is, the relation of language and the observable reality of what we consider part of language is one of *possibility*. So language might well be a cognitive capacity, but

does not need to be. Other than a cognitive capacity, we can think of language as the set of possible utterances, still in a physical sense (this is Michael Devitt's position, see [16]), or we can think of it as an abstract object such as is Peano arithmetics (this used to be Jerrold Katz' position, see [32],[31]). All these positions are fine with the creative commitment.

Anyway, it is not the task of linguistic to account for what has been uttered at some place, but what *can* be uttered; its range is not the actual, but the possible. In an idealized setting, the possible would be a superset of the actual. Things are, however, a bit more complicated: we observe actual utterances which seem to be wrong according to our knowledge of language, and often even the speakers who uttered them recognize them to be wrong and correct themselves. So these utterances should not be among our possible utterances. There is a lot of discussion where we have to draw the line, and in fact we will talk about this at length later on; we just mention for completeness that there are actual utterances we should not account for.

We conclude that it is the task of linguistics to account for (the structure of) language as an intensional object, or to put it differently, to account for linguistic creativity. An approach which aims at covering only the utterances found in the British National Corpus, or only covering the utterances with less than 20 words, we would not consider as satisfying or even relevant from the point of view of linguistic theory - though these approaches might be useful in many applications.

We will call this the *creative commitment*, which any serious linguistic theory has to make. Note that this mostly concerns syntax and semantics, but not exclusively: one might construct infinity arguments of the syntactic kind as well for phonetics and phonology, though they might not be as convincing, or even for pragmatics.

So there is broad agreement on this creative commitment, even among those who consider language as an abstract object, and linguistics not to be entangled with psychology. So one can adhere to the creative commitment without making a "cognitive commitment", which is to consider linguistics as part of psychology, and consider language interesting only as a capacity of the mind/brain. On the other hand, the cognitive commitment of the generative school and many others just seems to be a particular stream within this creative commitment.

2.1.2 The Mathematics of the Creative Commitment

Usually, we lose the broad consensus as soon as we look for more particular commitments, which are more concrete in a mathematical sense. However, among formal approaches to natural language, there seems to be a broad consensus how to fulfill this creative commitment: we treat languages as infinite sets (of strings, trees, pairs, triples...), which are finitely characterized. This is what we will call the classical conception. It is important to underline at this point: subscribing to the creative commitment by no means is the same as subscribing to the classical conception. The classical conception is maybe the most simple and straightforward, but comes with its own problems, as we will see.

In this dissertation, I will look at justifications of the classical conception, but also try work out alternative approaches. These will maybe be less simple and clear, but as I argue, provide ways to avoid some fundamental problems of

the classical conception, and maybe also allow for more adequate descriptions of some phenomena. Our fundamental problem is very briefly the following:

(1) The infinity of language cannot be conceived independently of the reasoning/theorizing subject.

That is, the languages we describe as linguists are *never* primary data. It is always an object constructed according to some pre-theory, which lets us interpret the finite fragment we observe in a certain way. In an interesting way, this makes linguists very similar to speakers: speakers as learners make languages infinite only in as far they construct theories around the language they observe in the sense of learning; but we might think of it also differently: they derive new utterances by *explicitly thinking* about the ones they already know, and this is exactly what linguists do. This leads to an important ontological distinction: for linguists, we must distinguish between actual (observable) and constructed data. We might argue - and will argue later on - that the same holds also for speakers: we must distinguish between the *immediate* use of utterances, where speakers use utterances they have heard and know in advance, and *creative* use. This distinction contradicts the usual assumption that all linguistic knowledge has the same ontological status, which comes necessarily with the classical conception of language as a set.

So we have, beforehand, an epistemic distinction for the linguist. This, however, also corresponds very roughly with a cognitive distinction of the speakers. Later on we will see how we might take this into consideration when construing a non-classical linguistic ontology.

2.2 The Epistemic Foundations of Linguistics

There is yet another perspective on our problem, namely the one of epistemology and ontology. It seems to be a basic issue in philosophy of science whether we depart from the question of what is given as our subject, or the question: what can we know of it? For example, in classical logic, we construct our ontology regardless of the question what we actually (can) know, whereas in intuitionism, this epistemic aspect is quite important. The same seems to hold for other sciences, and in particular, for linguistics. To my impression, there is lots of work on the ontology of linguistics, concerned with what language really is; but there is very little work on the question: what can we even know about language? This question is one of the guiding questions of this dissertation, so we look at language from the perspective of epistemology.

Before I proceed, I want to argue that linguistics is *always* maculate with epistemic concerns, no matter which perspective we take on it, and that for this reason, firstly, the epistemical approach to language has priority over the ontological one, and secondly, there is no way to get around linguistic metatheory in our sense. It seems to be important to me to stress this point simply because there seems to be so much ignorance of it. For example, Chomskyan scholars tend to do away with epistemic arguments on language with the objection: I-language has physical reality in the brain, so what we study is as real and tangible as any physical object (for a recent example, see [47]). This might be even true - but it does not change the point that we have no clue what it looks like - contrary to many other physical objects.

Ever since its very foundation, linguistics is struggling to become a science. It is not entirely clear, though, which one. One the one hand, the generative school insists on the objective reality of language in the mind/brain, which is the subject of linguistics proper. This is to say, first of all, linguistics is a branch of psychology, as it aims at describing something which has psychological reality in the mind of the speaker. But in a second step, it is even something like physics, as it ultimately aims at describing something which has a physical reality in the brain of the speaker. Note that one can go the first step without going the second.

On the other side, it has been claimed that linguistics is similar to mathematics. This claim is implicit in Montague's seminal work ("I do not believe there is any important difference between natural and formal languages", see [52], "English as a Formal Language"), and has been given a more explicit philosophical underpinning in the work of Jerrold Katz (see [31]). The argument goes roughly as follows: the logical structure of language is much more rich than our cognition. In particular, natural language semantics seems to contain very powerful logics, and we cannot say that these kinds of logics are part of psychology, because in the end because they are more complicated than what we can effectively handle. We therefore have to think of language as of arithmetics, an abstract object which exists independently of us.

We will now review both positions, and quickly show that in either approach, we fall back on the same limitations, and that in none of them, linguistics will become a true and proper science. Note, by the way, that the position we try to formulate here, seems to be that of Saussure, as the founder of modern linguistics, as he repeatedly claimed that in the science of language, it is the point of view which creates the subject, and that there is no way to think of the subject of linguistics independently from a certain perspective on it (see the fragments appeared in [15]).

We use this section to show that either way, linguistics in the creative commitment always carries an epistemic burden with it, and there is simply no way out of this. We show this first for the psychological conception of linguistics, then for the "abstract" conception" (nominalist, platonist conception).

2.2.1 The Epistemic Burden of Linguistics as Psychology

A commitment much stronger than the creative commitment is the *cognitive commitment*, which can be phrased as follows:

(2) It is the goal of linguistic theory to describe the native speaker's knowledge of language.

We ignore for the moment that this is heavily underspecified in many regards, and just focus on the consequence that it is the ultimate goal of linguistics to specify a cognitive capacity. This entails the following: (i) linguistics is a branch of psychology, and (ii) it is its ultimate goal to describe something which has a physical reality in the mind/brain. Regarding (i), we simply comment on the fact that usually psychology is exactly not intensional in the way linguistics is: it tries to account for concrete behaviors in concrete situations. This accounts for the fact that linguistics looks so different from psychology, if not in theory, then in practice.

We focus, however, on (ii). It is one of the fundamental claims of the generative school, that one day we will be able to verify its theoretical claims by directly looking into the physical reality of the brain. The huge gap between physical reality and linguistic theory is mainly due to our lack of knowledge on how the two interact, which does not hold only for linguistics, but virtually any domain of psychology.

Chomskyan linguistics strongly ignores all epistemic arguments on language, that is, arguments which say that we cannot know certain things (cf. the reply to an argument of Quine, that weak generative capacity underspecifies strong generative capacity; the answer is: "yes, but generative grammarians directly look at strong generative capacity!" – as if the latter were accessible independently from weak generative capacity). A major point seems to be the following: generative linguists insist that all linguistic theory is preliminary in the sense, that at a certain point, we will be able to directly read within the language module of the brain, and thereby answer all question. So theories are preliminary constructs, which will become obsolete (or confirmed), as soon as we properly understand the brain. They are preliminary descriptions of what is hard-coded in the brain, and linguistics will ultimately be a study of a physical reality in the brain.

But now assume, one day we will bridge this gap between theories and "the brain": we will be able to verify the correctness of a theory by directly looking into the brain. Then the brain must somehow satisfy our requirement of the creative commitment: it must somehow specify language in the intensional sense. If we look at it that way, by the assumption that we can read it, then it is just some very weird notation. To understand this notation means that we can translate into a notation which is closer to our usual language of mathematics. Now assume we will one day manage to do so.

Then our process of understanding is a process of translation (whether explicit or not), as we have to translate the code of the brain into a "human readable" format. But now, in order to know whether a translation is correct or not, how are we supposed to decide? Obviously, whether a translation is correct or not, we can only decide when we have some notion of a meaning, in the most general sense of something which remains invariant under the translation. Now, what is the invariant for the translation of a (piece of) brain into a formalized theory, or vice versa? As we have agreed on, they are both intensional descriptions of language. If there is anything by which we can compare them, then it their extension, and this is nothing but the possible verbal behavior they predict.

This in turn is nothing but language in the infinitary sense. Our ignorance of what language really looks like makes us unable to say whether the translation is correct. If we want to translate a text say from English or German, and we have no idea what it means, then there is no way to say the translation is adequate or not, and so the old problem strikes in once again! Of course, we are not completely ignorant about language: we know finite fragments. But regarding the infinitary nature of language, we are as ignorant as before, as we have no idea what the brain actually "denotes" or specifies in the linguistic sense. This in turn means: the entire decifration of the brain as a code, whether possible or even meaningful or not, will not reveal to us anything new on the subject of linguistics without further assumptions on the infinitary nature of language. And so despite all efforts, the problems we have to face in the end are epistemical.

2.2.2 The Epistemic Burden of Linguistics as a Formal Science

So how about the approach to language as an abstract object: does this prevent our epistemic problems? We will sketch here while these problems also pursue us in the latter approach.

Can we think of language in the same way as of arithmetics, such that the study of language is like elementary number theory? Let us briefly review what they have in common. Arithmetics arises, when we put down formally our basic intuitions about numbers (any numbers!). This gives rise to an axiomatic system, which is completely specified, and the properties of which we can study without any appeal to our intuitions. So the important thing is: we appealed to our intuitions once and nevermore; we put them down and now can study the resulting system on purely formal accounts.

This analogy of language and Peano arithmetics has been worked out in [41]. So how is this in linguistics? In principle, things go in a similar way: we have some data, we fix the rules according to our intuition on the data, and the resulting object is language, which is subject to our study. The fact that the number of rules we need to fix a language is vastly larger than the number of axioms for (Peano) arithmetics should not bother us for the moment. Neither should the fact that in arithmetics, we speak of one sort of abstract object, namely numbers, whereas in language we have to assume many sorts of objects, as noun phrases, verb phrases, sentences etc. What should bother us is that there are many ways to think of language: it is not clear at all which rules we take, which abstract objects we assume etc. So language is heavily underdetermined by the data; or put different, given one set of data, there are many languages which I can construct thereof. And any linguist knows that this not only holds in principle, but also in practice.

A more serious objection is the following: arithmetics is not affected of how we actually calculate. If some people (in "primitive" cultures) do not use numbers beyond three, that does not affect arithmetics. We can think of it, in fact, as completely independent of any calculating person. So there is a point, where we can simply say: we are *not* interested in people's intuitions, but only study formal systems. The same applies, maybe more neatly maybe, to logic. Logic, originally conceived of as a single one (Frege), is supposed to be a formalization of human reasoning (in the domain of mathematics). It comes under the implicit assumption, that all propositions have the same status for logics, regardless of how complex they might be, in the same way as we have made assumptions for *all* natural numbers, regardless of their size in arithmetics. We thereby define a field, formal logic, which we can study; and so it is independent of the fact that maybe scholars fail to fully grasp some line of reasoning, or make a wrong one. This is because we have used our intuitions to *define* a field. Can we make this step in linguistics?

Let us put this question more precisely: can we formalize linguistics to an extent, such that speaker judgments are completely irrelevant to it? So if a speaker tells us: this sentence of your grammar is completely wrong in my opinion, can we answer him at some point: "look, we have formalized reasonable intuitions, and if you do not like this sentence, this is because you are not talking about language in our sense". Actually, this seems to be a common practice in some cases, as we refute certain judgments for performance reasons. But

to do so in a systematic way, this would make linguistics meaningless: after all, linguistics has to do with data - language - and if we say it does not, after a certain formalization, then we would surely get the reply by any common sense person that what we are doing is not the study of language. Otherwise, any linguist could claim to be studying his own language, disconsidering other linguist's data and theories, all within what we would call one and the same language. Again, this seems to take the very essence out of the enterprise of linguistics, which after all is an empirical science.

So we have two major points: the first is, there are many possible linguistics, as the field of language is not neatly bounded; and secondly, each of them is open; there is not a definite formalization of the language we want to study, as there is a definite formalization of arithmetics (which still comes with a lot of problems of its own, see [3]).

2.2.3 The Epistemic Burden of Linguistic Judgments

So we see there are many ways to do linguistics, but none of them is immaculate from epistemological concerns. We will always have to ask ourselves: how much can we know about language? And the answer will always be: we will never be able to know as much as we actually want to be covered by our theories. This is, in a nutshell, because of the creative commitment and because the data depend on our own judgments. We will now quickly review why even these judgments do not come without epistemic concerns.

Roman Jakobson once attempted to formalize criteria for poetic language. This, of course, was strongly connected to the attempt of giving formal criteria of what we find aesthetic about certain linguistic expressions. He made a very fundamental restriction to the attainable goals of this enterprise: we will never be able to go beyond a certain threshold in formalizing aesthetic judgments, because the more we formalize our judgments, the more our formalization will come to have an influence on our judgments themselves. So our judgments lose their innocence; if we are theorists of our own judgments, we cannot construct theories without affecting our own judgments.

The same line of reasoning can be applied to linguistics, and in fact it has been applied to linguistics: it is a frequent criticism of empirical linguists that nobody understands the grammaticality judgments of theoretical syntacticians; that these judgments are no longer "natural data", because they are spoiled by the theories which have been developed by the same people who make them.

Now, one could say (and empirical linguists often do say so:) there is a simple solution to that issue: we simply take the judgments of innocent speakers, by making experiments or looking at corpora. But in a sense, this would be the same as saying: we solve Jakobson's problem on aesthetic judgments, by letting the part of judging to people who have never thought a minute about poetry in all their life. But that is clearly not what Jakobson intended; the main point is: making an aesthetic judgment on a poem, and reason on what is a good poem, are intrinsically tied; even if the reasoning does not take place on the level of an explicit formal theory. And we just do not want to rely on judgments of people never having thought about poetry in their lives! Now, the same seems to hold for linguistics: what theorists call their object of study is most emphatically not what the naive speaker finds acceptable. We want many more utterances, namely, we want infinitely many; and on the other side, we probably do not want some,

which sound acceptable at the first hearing. If the judgments of trained linguists are problematic, so are the judgments of very naive speakers, who maybe a minute after making their judgment become aware that they have been wrong (see for example (16) and the discussion on intensional linguistics). So on the one side, (naive) speakers sometimes make mistakes which they themselves consider as such. On the other side, there is nothing which prevents the most naive speaker from reasoning about his language (with whatever outcome). Given this, we can ask: what does a "naive speaker" even mean, and is the naive speaker not as much a fictional person as the infamous "ideal speaker" of Chomsky? We will discuss these problems in much more detail in the section on intensional linguistics.

Linguistics, in the end, is a science over judgments of speakers. But a judgment itself presupposes a form of pre-theory. It is this inseparability of theory and pre-theory, which makes linguistics particular in the sciences (or more general, any science whose primary datum are human judgments). In these sciences, empiric facts always have an epistemic flavor.

2.3 Some Fundamental Concepts of Metalinguistics

We start with fixing some terminology. As we have said, the proper subject of linguistics are infinitary languages. We will call such an object "language". That is to say, "language" is something we construct from a finite dataset, and which is considered to be an adequate subject for linguistic theory. This is opposed to what we call the *observable fragment* of language, or simply o-language. This is the fragment of language we can observe in principle, that is, the set of utterances we *could* observe at some point. Importantly, there is no upper bound to the length of sentences we can observe, so the infinity argument as conceived for "language" also holds for o-language. Nonetheless, the two do usually not coincide: in the usual setting, linguists assume sentences to be grammatical, which are not accepted or uttered by speakers (cf. the introduction), and this is where "language" exceeds o-language. In principle, we could say that o-language is contained in "language"; this is the standard assumption, but is not necessarily the case: Haider ([22]) sketches a (meta-) theory in order to allow for sentences of o-language not to be contained in "language".

A third important concept is the one of an *observed language*. An observed language is the set of data a linguist considers. Contrary to the observable language, the observed language is always finite. The distinction between infinite o-language and finite observed language is of a central importance for us for two main reasons: first of all, as o-language is infinite, it is not clear what it looks like: the fact that, by definition, we could observe all utterances it contains does not entail that we know what these utterances look like. This might sound paradoxical, as we call it the observable language; but still it is well-known to the linguistic community that from time to time some new data come up, which, though being observable, simply have not been considered by any linguist so far (an important example of a "late" discovery are parasitic gaps, which were unknown to early syntactic theory). So we have no right to claim that we exactly know what o-language looks like, even though it consists only of utterances we

can get to know. On the contrary, an observed language is defined by the very fact that we know it.

But also apart from this, there is good reason to distinguish the two: as we have said, linguistic metatheory has to be strictly finitary in its methods. This entails that we cannot just simply project an infinite language. That is, in general there is no finitary procedure which ensures we get from o-language to "language" (that is, without further assumption). Therefore, even if o-language would be accessible, it would be a bad point of departure for linguistic metatheory, exactly *because* it is infinite.

This gives rise to other important concepts. The first one is, put set-theoretically, "language" *minus* o-language, where by "*minus*" we mean standard set-theoretic subtraction; that is, the subset of language we *cannot* observe in principle, or put differently, that will never be observed. This is what we might call the "dark zone", as it is the part of "language", which is not accessible to any empirical observations, and has to be distinguished from the part of "language" we have not yet observed, but might observe at some point, which is o-language *minus* observable language.

These are so to speak the main ingredients, the objects of the metalinguistic universe (though of course not the objects which are given to the metalinguist!). How does this relate to linguistic reality, that is, to natural languages as we usually think of them? Regarding a natural language such as German, we can think of it in two different ways: we can think of it of an empirical language, which thus is an o-language – the set of all German utterances we can possibly observe. The second way is to think of it as a theoretical language, as for example, the entity which is the extension of the knowledge of language of a German native speaker. Note that both conceptions presuppose considerable idealizations such as the fact that all German speakers agree on all judgments etc. But importantly, there are (infinitely) many observed languages with respect this this one o-language, which is observable German, the set of all German sentences we could hear. In fact, any finite fragment of o-language qualifies as an observed language, though obviously not every fragment qualifies equally well, as regards the projection: some fragments are presumably more informative on the infinitary nature of "language" than others. This is quite intuitive. A consequence which might be rather unexpected is that with respect to this one o-language of empirical German, there is also an infinite set of "languages", namely possible theoretical "languages" of German. The reason for this is: there are many ways of projecting observed languages to the infinite. And even if we might "exclude" some of them later on - we will discuss later how this might work - there are still infinitely many which only differ in the dark zone, such that there is no way to empirically distinguish between their adequacy. The unity of language in the empirical sense thus corresponds to the uniqueness of an o-language, not to "language"; and in our ontology, i tis better to think of a natural language as German or English as an o-language, not as a "language". But note that regarding a language as German, there is a difference between German as an *observable language*, which corresponds to an o-language, and German as a *theoretical language*, which qualifies as a "language", and which might be a model of what a German speaker knows. Note that both, the observable and the theoretical German belong supposedly to the real world, but the former as a datum, a source of evidence, the latter as a construction, which in some way or other has to exist (in the mind of speakers or elsewhere), but for which there is

no direct way of verification.

There is still one very important issue. From a metalinguistic point of view, we cannot give negative evidence the same status as positive evidence, for then there would be no space for a "dark zone". In our approach, we must say that there is positive evidence, and its absence. Still, this is somewhat unsatisfying: whereas for some strings such as

(3) a. What are the chances good I left without caring about?
b. The cat the dog the man let loose, chased ran all over the place.

we might expect that they might be okay given some linguistic reasoning - or more simply put: we simply want to remain agnostic about their grammaticality - for others we are pretty sure they are not, such as

(4) Peter John be yesterday.

What does pretty sure mean? For us it means: we are as sure as we can be, saying that if this sentence is okay, then just anything would be fine. Surely, this is not what we expect or want, because it would make the entire procedure of projection trivial: there would be no way of rejecting any projection as inadequate. Therefore we would reject a projection which contains these strings. This gives us a *negative language*, a set of strings we accept under no circumstances in our "language"; but not because we know that they are wrong – we cannot know such a thing – but because then we would have to accept that just anything is fine. From this it follows that negative evidence has a fundamentally weaker status than positive evidence: the fact that something is judged unacceptable is not enough to make it a negative evidence. And even if we agree on some negative evidence, we have to be careful in using it: if we use it in the construction of "language" itself, then we cannot use it for rejecting certain constructions anymore! This will play an important role in the sequel. Note that as there is no way to enumerate this negative language, we have to assume that it is finite. Moreover, to keep up a division of labor, I should add that though the negative language is not observed but rather constructed, the task of its construction is clearly a *linguistic* task, not a metalinguistic one, as the construction proceeds over linguistic intuitions.

2.4 The Projection Problem

The projection problem is usually the term for a problem of the language learner: he is exposed to a finite set of utterances, and has to learn an infinite language. Therefore, at some point, the learner has to *project* the finite language into the infinite. We use the term here and throughout this work in a different sense, which is however so similar, that I think it would be wrong to coin a new term for such a well-known concept. For us, the projection problem is the same problem, but not for the speaker, but rather for the linguist. The linguist looks at a finite dataset, and has to describe an infinite language. So at some point, he has to make the transition from a finite language to an infinite language. This is our version of the problem. Note that the problem poses itself in a slightly different way: whereas the speaker has to do the transition "automatically", that is, without a conscious decision, the (meta-)linguist can do the projection

consciously after having gathered relevant data. The main difference between the two is: the (meta-)linguist can reflect over the data and whether it is convenient for a projection. So he has the great advantage that he can be aware of the result of the projection, and depending on this, project or not project a dataset.

This difference means that the linguist is essentially in a much better position than the learner. This also is the reason that we will not approach the problem in an algorithmic fashion: we do not want to deprive the linguist of his choices and automatize the projection of an infinite language of which we observe growing, finite portions: because this is paramount to giving up our most important advantages, which we have with respect to the naive learner, out of hand. Another important difference is the following: our procedure is principally open-ended. In a learning setting, there is always something that has to be learned. But in our setting, we do not know what we are supposed to learn, because as meta-linguists there is nothing we can say *a priori* on the nature of language. So for us, the problem is essentially open ended: knowing nothing on the infinitary nature of language, it makes little sense to speak of learning certain classes, and make the question: "is the class C an adequate class for natural languages" depend on the question "is the class C learnable"? Rather, things work the other way round: knowing that projection always results in a class C, we know that natural language being contained in class C is an artefact of our meta-theory. This is, briefly, the outline of our version of the projection problem.

2.5 A Sketch of the History of the Problem...

Our projection problem is very closely related to another, even more fundamental question of linguistic theory:

(5) What is the proper subject of linguistic theory?

Despite many scholars have contributed to this problem, probably only on two names everyone will agree that they have truly defined (or re-defined) the matter of linguistics: Ferdinand de Saussure and Noam Chomsky. Interestingly, the problems we want to address are already present in Saussure's original work. A hundred years ago, the Cours de Linguistique Générale was published, giving for the first time a preliminary definition of what should be the proper subject of linguistics proper. Apart from some minor revisions, mostly done by Chomsky within what is now called the "cognitive turn", Saussures definitions remain valid up to today.[1] As Saussure said, the proper subject of linguistics should be *langue*, a concept which Chomsky renamed to *competence*, later *I-language*.[2]

The Cours de Linguistique Générale was published at a time when linguistics mainly consisted in the study of the historical development of language and its

[1] Note that Saussure's conception was not as antimentalistic as became the one of later structuralism.

[2] Note that Saussure's work is now judged to be a synthesis of what other theorists said, such as Humboldt and von der Gabelentz. Also for Chomsky, there is a long discussion which portion of his ideas have to be attributed to his supervisor Zellig Harris. We will however not touch upon problems of authorship and editorship; so everything we say on persons has to be taken *cum grano salis*: even if they had developed all their ideas on their own, they could not have been as successful if there would not have been a broad acceptance and consensus that their steps were steps to take.

geographical distribution[3] (we nowadays say: diachronic and diatopic linguistics). Saussure's position in this regard was, to put it shortly: linguistics proper should be interested in the system of a language, not in its history, not in its variations, not in its individual aspects. One of his maybe most capturing pictures was the one of the game of chess: studying chess proper, we should study the rules of the game; it does not matter whether we have figures of wood or of ebony; and it does not matter whether chess came from India through Arabia etc: that is all interesting, but it is not chess proper. I can know all this without knowing any chess, and I can know the game perfectly without knowing any of this.

Chomsky in turn started his work in the age of structuralism and behaviorism, when everyone was interested in language as an extensional phenomenon: the common picture of language was the one of a "structured inventory" of signs, and the goal was to give a distributional analysis of all utterances and parts of utterances. In particular, there was little attention paid to the fact that languages consist of infinite sets of utterances, because the focus was put on observable sets, which are always finite. The argument for infinity of languages has in fact to be based on linguistic intuition, and the appeal to intuition was ill-reputed as "mentalism". So there was no good way to access the "intensional" aspect of language, that is, the set of *possible* utterances.

Chomsky's work gave rise to the so-called cognitive turn: he emphasized that language has to be based on a mental capacity; the subject of the grammarian is to give a description of a speaker's knowledge of a language; and linguistics in general has to describe the underlying mental capacity which allows humans to learn a language. Chomsky's focussed on the following point: language consists of an infinite set of utterances, which speakers learn to master in a very short amount of time. This focus on learning contains a more general problem: the speaker has to get from a finite database (in a finite amount of time) to a grammar which generates infinitely many utterances.

Chomsky found himself in front of a dilemma, which already Saussure encountered: Saussure said we are not interested in the individual aspects of single speakers, but still in a psychological phenomenon. That led him to say that langue is a collective, social object, while being at the same time psychological (see [61],p.37). Also Chomsky was not interested at all into individual aspects of language use, but strongly into the psychological, or rather cognitive aspect. He escaped the paradox with the invention of the "ideal speaker". It was the grammarians task to describe the "competence/knowledge of language of the ideal speaker".[4]

2.6 ...and Why the Classical Solution does not Work

Chomsky's by now classical work "The Logical Structure of Linguistic Theory" (henceforth LSLT,[4]) prescribes also a solution for our version of the projection problem. By the way, it remains to my knowledge the only work explicitly addressing this problem, so we can dub this solution "classical". I also have the impression that if linguists think on our problem, they normally come up

[3] As it did in fact for Saussure during his lifetime.

[4] This ideal speaker remains a weak point in the conception; we will discuss this later on.

with a similar solution. It works roughly as follows: we look at a language, that is, observe a finite set of utterances of this language. We write a grammar, which covers the data we have seen so far. As new data comes in, we change the grammar, eventually make it describe an infinite language. We continue this process as new data keeps coming in. At some point, our grammar will *converge*, that is, for all new data we see, our grammar will cover it. This grammar is then descriptively adequate. If we aim for more, we might have additional criteria for evaluating grammars, which let us choose between the different possible grammars, and the best grammar according to the criteria will then be *explanatorily adequate*.

Either way, the resulting grammar, will describe an infinite language (otherwise it could not remain unchanged given new data), and this grammar will at the same time define "language", the proper subject of linguistics. The trick is obvious: it aims at making meta-linguistics part of linguistics, and solve the projection problem and the problem of grammar writing at the same time. Now why does this not work?

Apart from technical problems with this approach (all problems coming with classical learning in the limit have to be considered, [21]), it is inadequate for reasons of principle. The reason is as follows: the procedure and criterion of convergence is *not finitary*. That is, we do never know after a finite amount of time, whether our grammar converges or not. But we have said that linguistic metatheory has to be finitary in its means, for otherwise it is useless. On the other hand, assume we know at some finite point after a finite amount of steps that our grammar will converge. But then we have to know *independently* what "language" looks like, for otherwise, how could we know? So the whole procedure is pointless for the purposes of metalinguistics: as it presupposes that we know the shape of "language", it does not tell us anything new.[5] For completeness we should also add that this is not the only approach taken to the problem in LSLT; in fact, we find quite contradicting positions regarding the problem almost on the same page, see p.96. But this approach seems to have become canonical, not in practice, but as the theoretical solution to the problem.

In conclusion, the classical procedure from LSLT does not solve any of our problems, because it is either infinitary, or circular.

2.7 Questions Around the Projection Problem

2.7.1 Language is Not Designed for Usage

To show that the above considerations are of relevance for *linguistics proper*, we show some invalid arguments on language. As a first example, take the following Chomskyan line of reasoning: Chomsky notes that we only use a small fragment of "language" (as he defines it), and that this fragment is even quite messy (in the dark zone, things are supposed to be more clear). He concludes that this is strong evidence that language is not at all designed for usage, for then we would not expect to find similar properties (this is taken from [5]).

After all we said, we need not explain why this argument is entirely based

[5] Another fundamental problem of this procedure is overgeneration: as the data which comes in is positive, we can always write trivial grammars. So effectively, we would also need an infinite amount of negative data in order to make this procedure work.

on Chomsky's own untestable assumptions. Making different assumptions, we could make different arguments on the nature of language, no more and no less convincing and valid than the one above.

2.7.2 Insights by Descriptive Elegance

Another one is the argument that the generative program has achieved great insights by looking for descriptive elegance (which of course is an arguable notion, but let us accept it as it stands. If we just look at the data we have and descriptions we have, it is hard to see how there is a great elegance: in fact, tiny phenomena cause huge problems to strongly principled approaches to linguistics as mainstream generative grammar. But the statement becomes suggestive or true under two additional assumptions: there is a part of language which is interesting for linguistic theory (the "core"), and a part which is uninteresting ("periphery") (where the core is *defined* by descriptive elegance). Furthermore, only core grammar is projected into the infinite, whereas peripheral constructions are limited to some constant bound (and therefore uninteresting). Also this line of reasoning is self-fulfilling: the statement on language is rather a statement on the approach to language taken. Note that the distinction between core phenomena and periphery is a typical example of what is known as immunization in theory of science: it makes a theory impervious to criticism, because anything which goes against it, is almost per definition uninteresting (from the theory-internal point of view).

2.7.3 On Recursion

One usually argues that center embedding is unbounded *in principle*, though in performance we do not observe cases which would suggest that. The argument is that we find the same distribution for simple NPs and NPs with relative clauses which contain NPs. In fact, there has been a lengthy debate on whether there is recursion in a language, based on the observation whether there is a category of a certain kind embedded within the same category. However, on a more abstract level all the arguments seem to be circular: they already presupposes the infinite language which we are supposed to construe by recursive phrase structure rules.

The syntactic conception of distribution says that the distribution of two strings $\vec{x}, \vec{y}$ is identical, if they occur in the same **contexts**. A context is a pair $\langle \vec{w}, \vec{v} \rangle$, and $\langle \vec{w}, \vec{v} \rangle \in C(\vec{x})$ exactly if $\vec{w}\vec{x}\vec{v} \in L$ for a language L. (We ignore at this point the pervasive problems which result from a purely distributional analysis of natural language, and just focus on a very particular one).

Any interesting grammar is recursive at some point. On the other side, given our distributional definition, no finite dataset will ever be able to show this:

Proposition 1 *Assume that $\vec{x} = \vec{w}\vec{y}\vec{v}$, where at least one of $\vec{w}, \vec{v}$ is non-empty. Then there is no finite dataset with at least one occurrence of $\vec{x}$ or $\vec{y}$ where $C(\vec{x}) = C(\vec{y})$.*

We omit a proof, as this is quite obvious: there is always at least one distinguishing occurrence of $\vec{y}$; just take the longest string that contains $\vec{y}$, substitute $\vec{x}$ and you get a contradiction. This means that in finite datasets there is no distributional evidence for recursion. But this also holds for the

non-trivial case of the infinite dataset of observable utterances in the case of center embedding.

Now, recursion in the usual sense is a property of grammars, not of languages, and when we write a grammar for a language, we have to ask ourselves which rules are recursive and which not. If the language is infinite, we will of course have to use recursive rules at some point. However, if we first have to project a finite language into the infinite, then there is simply no argument which forces us to do so using recursive rules: it will always be a matter of choice.

Note that this is not an argument against analyses using recursive rules or even recursion in natural language; we just say that there is nothing which forces us to adopt this kind of analysis - it is a theory, rather than a datum. This holds for all the examples, whose failure consists in taking something for a datum which is only a theory.

2.7.4 Patterns and Dependencies

One of the central assumptions of classical linguistics can be dubbed the *phrase-structure hypothesis* (PSH). It is of fundamental importance for any theory which aims at covering both syntax and semantics of natural language. In my view, its importance lies exactly in the fact that it is mostly assumed to be without alternative, up to the point that it is not even worth mentioning it.

There are two types of evidence for syntactic structures: (i) patterns in strings we observe, and (ii) intuitions on which elements belong to each other in terms of meaning. It is important to note that both are distinct; (i) is properly syntactic in nature, whereas (ii) is rather semantic. Note also that this distinctness has often played a prominent role in linguistic theory, as for example in the discussion whether Dutch is context-free etc (see [27],[40]. Patterns are closely related to the concept of distribution and weak generative capacity. To adequately model them we simply need to generate the strings we observe in some way or other. Dependencies are a bit more complicated. Let f be a function of syntactic combination (which is usually concatenation, but let us try to remain more general); so it is a function from an ordered pair of strings to a single string. Now given a string $\vec{z} = f(\vec{x}, \vec{y})$, a dependency of $\vec{x}$ and $\vec{y}$ is simply an intuition that the meaning μ of $f(\vec{x}, \vec{y})$ is best expressed as a function on the meanings of $\vec{x}$ and $\vec{y}$. Note that this is not a rigid notion: what we describe is simply the intuition that the meaning is computed best in this way! As is well-known from type logical grammar, appropriate meanings can be computed in many ways and according to many syntactic compositions. Now the PSH can be roughly stated as follows:

(6) If there is a dependency between $\vec{x}$ and $\vec{y}$, then there is a rule $\alpha \to \beta \ \ \gamma$, where $\beta \to^* \vec{x}$, and $\gamma \to^* \vec{y}$.

That is to say, the hypothesis says that patterns and dependencies we observe are due to the same mechanism. This is of course closely related to the question of compositionality, a topic which is explored in much more detail and in a more abstract and precise fashion than I can undertake it in [40]. Therefore, I will only sketch the arguments in a very informal manner. It is quite easy to show that it is mathematically possible to maintain this view given any finite relation of form and meaning. On the other side, it remains an open problem whether

there exists is some relation for which there does not exist a semantics which does conforms with the hypothesis. So it is still unclear whether the PSH or compositionality is in fact an empirically testable issue given *infinite* languages. On the other side, it is clearly a testable issue if we assume that the language (one side of the relation) is generated by a restricted kind of grammar (see again [40] for examples and proofs).

So given the commitment to a restricted formalism, compositionality and PSH are empirically testable – if we are given an infinite relation, which for natural language is not the case! So the PSH and the related notion of compositionality are – for natural rather than formal languages – *not* empirical issues, but rather issues of projection in a syntactic and semantic sense, namely the construction of an infinite relation out of a finite one. I have to underline that Kracht is fully aware of the fact that even if compositionality was an empirical question for infinite relations, it surely is not for the finite fragments of "language" we observe (see [40], introduction). Yet, he seems to be an exception among semanticists. So again we have a topic which is usually considered to be empirical, but as a matter of fact it is one of linguistic metatheory. We simply *construct* languages in a way such that they conform to PSH, and if this is not possible, things get problematic (see the example War in Vietnam or no war...). The PSH (and compositionality) can only be thought of to be empirical, if we blur the distinction between linguistics and its metatheory. On the other side, to see the interaction of form and meaning to be a question of language construction rather than observation seems to me an extremely interesting enterprise.

2.7.5 Weak and Strong Generative Capacity

In formal language/grammar theory, one distinguishes between weak and strong generative capacity. The former designates the language as a set of strings, which a given grammar[6] generates, the latter designates the set of derivations (in Chomsky's terminology: structural descriptions) with the associated strings. According to Chomsky, weak generative capacity is of little if any interest to theoretical linguistics (see for example [5],p.16). This is somewhat surprising, given their evident epistemic priority: in fact, all we can observe is actually strings - no one has ever observed a derivation as a primary datum, nor will anyone ever do so, probably; all we see is utterances and speakers judgements on utterances. According to Chomsky, one reason for the neglect of weak generative capacity and associated problems in the generative theory[7] is the following: our language use is restricted in such a way that what we actually observe and need to handle is only a trivial fragment of the language we "know" (we use "know" parallel to "language", to say that this "knowledge" is not a datum, but a theoretical construct. Therefore, matters of complexity in the sense of formal languages, as decidability, parsing efficiency etc., are trivialized: for the fragment we use, these matters are of no importance anyway. What in turn *is* important is to get the derivations right, in order to achieve explanatory adequacy (and get a semantics).

There is a good point in this; but from our perspective of the priority of epistemology, we would see things exactly the other way round: if something

[6] We ignore for the moment other generating/recognizing devices.

[7] This only regards mainstream generativism; there is a tradition which is strongly concerned with both weak and strong generative capacity.

is empirically unaccessible (like the presumed complexity of language), there is no reason to assume its existence in the first place. The Chomskyan answer is of course: the reason for assuming its existence is precisely strong generative capacity and explanatory adequacy. Now from our point of view, accepting this argument, we can also go one step further: if we assume that questions of weak generative capacity are trivialized by language use, we might also ask: (i) is "language use" (in our terms: the observable language) restricted in a way such that it trivializes questions of generative capacity etc. within a precise bound? And if this is the case, then (ii) it could be the case that it trivializes questions of *strong* generative capacity! The first question is quite clear and has been addressed various times ([65], [20]). The second point needs some explanation. There are two main reasons derivations are interesting to linguistic theory. The first one is: we usually want some kind of compositional semantics, and semantic representations canonically depend on derivations. The second one is less innocent: we usually formulate our theoretical requirements for explanatory adequacy on derivations, not on associated strings. We have to explain what it means for a dataset/language to trivialize strong generative capacity. We start however by making the idea of trivialization precise for weak generative capacity. Let $\mathfrak{G}, \mathfrak{G}'$ be classes of grammars; we write $\mathfrak{G} \leq \mathfrak{G}'$ iff the class of languages generated by grammars in $\mathfrak{G}$ is a subclass of the class of languages generated by grammars in $\mathfrak{G}'$.

Definition 2 *Given two classes of grammars $\mathfrak{G}, \mathfrak{G}'$, $\mathfrak{G}' \leq \mathfrak{G}$, a class of languages $\mathcal{D}$, we say that $\mathcal{D}$ **weakly trivializes** $\mathfrak{G}$ with respect to $\mathfrak{G}'$ if for any $D \in \mathcal{D}$, if there is $G \in \mathfrak{G}$ with $L(G) \supseteq D$, then there is $G' \in \mathfrak{G}'$ with $L(G') \supseteq D$.*

Now this is not a very strong notion: if for any alphabet Σ, there is a $G \in \mathfrak{G}'$ such that $L(G) = \Sigma^*$, then $\mathfrak{G}'$ trivializes any class of grammars. What should give us a more adequate notion is a notion which considers both positive and negative data. As we will argue later on, the linguist is provided with both positive and negative data when performing his task. So assume we have a pair of sets $(\mathcal{D}^+, \mathcal{D}^-)$, such that $\mathcal{D}^+ \cap \mathcal{D}^- = \emptyset$ (here $\mathcal{D}^+$ is the positive, $\mathcal{D}^-$ the negative data). Now we say the following:

Definition 3 *Given two classes of grammars $\mathfrak{G}, \mathfrak{G}'$, $\mathfrak{G}' \leq \mathfrak{G}$, a class of pairs of finite, disjoint languages $\mathcal{D}$, we say that $\mathcal{D}$ **trivializes** $\mathfrak{G}$ with respect to $\mathfrak{G}'$, if for any $(D^+, D^-) \in \mathcal{D}$, if there is $G \in \mathfrak{G}$ with $L(G) \supseteq D^+$, $L(G) \cap D^- = \emptyset$, then there is $G' \in \mathfrak{G}'$ with $L(G') \supseteq D^+$, $L(G') \cap D^- = \emptyset$.*

This is a meaningful notion, which we have to explain briefly. By $\mathcal{D}$ we intend the class of observations we can make, and because observations are finite, we can assume that they always form a subset of the class of all disjoint pairs of finite languages. So we can take this class as an example, and it is in a sense the strongest case: because if $\mathcal{D}$ trivializes $\mathfrak{G}$ wrt. $\mathfrak{G}'$, and $\mathcal{D}' \subseteq \mathcal{D}$, then also $\mathcal{D}'$ trivializes $\mathfrak{G}$ wrt. $\mathfrak{G}'$. So assume $\mathcal{D}_{fin}$ is the class of all (L, L') such that L, L' are finite and $L \cap L' = \emptyset$. What trivialization results do we obtain? One obvious thing is the following: $\mathcal{D}$ trivializes any class of grammars wrt. the class of regular grammars, for obvious reasons. Even smaller classes of grammars do the same, take the star-free languages, and even the co-finite languages. So if we go for trivialization, we finally end up with quite trivial grammars. Note, by

the way, the connection with learning and Angluin's theorem (see [1]); we will elaborate on these notions in chapter 6.

Weak trivialization is what according to Chomsky makes weak generative capacity uninteresting for linguistics, and one might follow him in this point. But the same concept can be defined for strong generative capacity, though the definition requires a bit more work. When we start, we have to take some concept of structural description (SD) for granted (this notion is taken from LSLT; I will use it only here); and we denote by $SD(\vec{x})$ an SD which is associated to a string $\vec{x}$ (but note that there are possibly many SDs for a single string). We now assume that grammars, more than generating strings, generate strings associated with structural descriptions.

Definition 4 *Given classes of grammars $\mathfrak{G}, \mathfrak{G}'$, $\mathfrak{G}' \leq \mathfrak{G}$, and a class of disjoint pairs of languages $\mathcal{D}$, we say that $\mathcal{D}$ **strongly trivializes** $\mathfrak{G}$ wrt. $\mathfrak{G}'$, if for every $(D^+, D^-) \in \mathcal{D}$, if there is a $G \in \mathfrak{G}$ such that G assigns at least one SD to every $\vec{x} \in D^+$ and no SD to any $\vec{y} \in D^-$, then there is a grammar $G' \in \mathfrak{G}'$, such that (i) $\mathfrak{G}$ assigns an SD' to every $\vec{x} \in D^+$ and no SD to any $\vec{y} \in D^-$, and (ii) there is a bijective map $\phi : \mathcal{S}[D^+] \mapsto \mathcal{S}'[D^+]$, where $\mathcal{S}[D]$ ($\mathcal{S}'[D]$) is the set of structural descriptions which $\mathfrak{G}$ ($\mathfrak{G}'$) assigns to some $\vec{x} \in D$.*

Note that this definition is much more problematic than the first one, mainly because it is not always very clear what counts as a structural description. Take for example tree adjoining grammars (TAG). In most standard approaches to their semantics (see [28]), we do not interpret the derived tree, which usually counts as the structural description, but rather the *derivation tree*, which is a regular tree (contrary to the derived syntactic tree). Kobele ([35]) provided a semantics for minimalist grammars ([66], [50]) in a similar way. The reason this is possible is that the derivation tree *encodes* the syntactic tree. Therefore, ϕ just has to be an appropriate coding, which however might be difficult to find. Note, by the way, the fact that we only need no encode the final SD, not all of its derivation steps, and that the coding needs to work only for strings in D^+ – the structural descriptions on other strings do not matter at this point.

A by now classical example for what we here have defined as strong trivialization was provided by GPSG: though GPSG-grammars are context-free, they could handle phenomena of movement in much the same way as the much more powerful transformational grammars. They therefore trivialized them for what was considered to be "natural language" at that point. When this dataset was enlarged with data from Swiss German, GPSG-grammars turned out to be inadequate to describe some regularities (in the sense of structural descriptions). So the question of strong trivialization is actually a very interesting one. There has been some very interesting work on the topic of coding and codability of properties within grammars, as ([37],[57]) which for reasons of space we can only mention.

The intuition behind the definition of trivialization is that for the data we can observe, we trivialize a certain grammar if we can *simulate* the strong generative capacity of our desired grammar in terms of a weaker grammar. In this case, though we abandon the phrase structure hypothesis, all our intuitions can be captured and all our generalizations can be maintained (for the finite fragment we have observed!) within a weaker grammar. One might think of these arguments as essentially formal and non-linguistic, but if it is the case that our observable

fragment of language trivializes the grammars which we think to be descriptively adequate, this would be a great insight into the structure of language. It would mean that the patterns of our language are used precisely in a way such that we can simulate complex patterns with a simple grammar.

This brings us back to our projection problem. The notion of trivialization allows us for several choices: either we go the Chomskyan way, and say: we do not need to care for any complexity arguments, because our formalisms are trivialized by the data and much weaker formalisms. But this choice has a somewhat unscientific flavor, after all. So we rather might want to go the other way and say: why should we need stronger formalisms at all, given the data we have and formalisms which (strongly) trivialize them, which means that empirically speaking, they are as adequate? After all, scientific thought should lead us to choose the simplest hypothesis! The point is that the choice between the strong and the weak formalism is usually not based on the data we *observe*, but on the data we *construct*, which is, in the Chomskyan reasoning, exactly the more complex language! So this is another example where a question, which is of presumably linguistic nature, turns out to depend on projections and thus on linguistic metatheory. We will later on pursue the ideas we have sketched here in the setting of finitary linguistics.

2.7.6 Chunking

As psychological research has revealed, humans tend to construct chunks of subsequences when they perform complex action sequences (e.g. in dance: [62]). There is a lot of evidence that they do the same for complex mental operations (as for chess, see [13]). We might now ask: are there similar processes for natural language? Psychologically, the answer is probably yes. From the linguistic point of view, the question arises: are there any underlying regularities in the structure of natural language which are *not* visible on the level of exhaustive analysis? Most, if not all of generative grammar takes it simply for granted, that the most exhaustive analysis results in the most compact description. Though this is not implausible, from a mathematical point of view, there is no reason why this should necessarily be the case (see [45], for example the concept of higher block code). This question is thus not only motivated by cognitive considerations, but also by considerations from coding theory.

The reason why this is interesting in this context is the following: if we assume that our grammar specifies our language in terms of (distinct) rules for constructing chunks and rules for combining chunks, and not in terms of rules on categories and ultimately single words, then this might result in a completely different projection into the infinite. Take for example the following:

(7) The cat which the dog chased ran quickly.

The dependency structure for this sentence seems pretty clear. Using phrase structure rules (or any other kind of rules) to generate the desired dependencies, it is quite obvious that there is an NP within an NP, so the rules involved in generating this sentence are recursive, and therefore also generate sentences of the type:

(8) The cat which the dog which the mouse woke up chased ran away.

This is, however, not an inevitable consequence of the assumption of a certain structure: it only follows from the additional assumption that grammar rules always provide an exhaustive analysis. This in turn has the consequence that the embedded NP has the same status as the head NP. In an approach based on chunks, this need not necessarily be the case. Note that we have to take some care how we formally implement chunks. We can well do so using phrase structure grammars, but we need to introduce new categories, which allow to distinguish whether a category (say, NP) occurs within a chunk, or forms a chunk by itself. This is of course only one way to capture the fact that there are different levels of analysis, and it is the one we will adopt here.

So, there might be a chunk [NP RC], which is fully specified as such, and which does *not* allow for another category RC within the RC already specified. This is possible because the NP within the RC has a different status from the one outside, and rules for combining chunks are different from rules generating chunks.

Formally, we can easily translate chunks into phrase structure rules by assuming additional categories, which are assigned to elements depending on where they occur: as a chunk for themselves or within a chunk etc. So we can take RC-chunk as a shorthand for a (possibly even infinite) number of sentence generating rules. For example, RC could generate a set of non-terminals N_{RC}, which equals the set of standard non-terminals, but does not contain any rules introducing the category RC. This way, we can easily think of rules which make the following example a sentence of our "language", though not the previous:

(9) The cat ran away, which the dog chased, which the mouse woke up.

There are many more examples which could be handled like this. The concept of levels is also quite well-known; just consider the well-known root transformation from earlier generative theory (the concept of different levels of rule applicability is closely related to chunking).

As another phenomenon consider the case of parasitic gaps, which do not involve recursion at all (by definition!). Still they cause many formalisms to enhance their descriptive power - unnecessarily, due to the concept of transformation of the underlying phrase structure.

Note that we do not argue that the way of construing language by chunks is better than the usual one. We just want to point out that different analyses also result in different "languages". So chunking cannot be ruled out on the assumption that "language" has a certain shape in the infinite. We will further pursue this idea in the following sections, in particular in the context of what we will call the finitist meta-theory.

2.7.7 *pro*-drop, Syntactic Complexity and Trivialization

As we have seen, descriptions tightly interact with projections. We will now see a case where trivialization might play a crucial role not only for linguistic metatheory, but linguistic theorizing itself. We will have a look at a very well-known phenomenon. There are languages which allow for the so-called "little pro" or simply *pro* in subject position. *pro* is the category assigned to invisible arguments of verbs, as to phonetically empty subjects in languages as Italian or Spanish, and any arguments in languages as Japanese or Korean. A priori,

pro is conceived of as a normal pronoun with the property of being phonetically empty (we will restrict ourselves to Italian in what is to follow, but find a similar behavior in all of the above mentioned languages). However, *pro* behaves differently from "overt" arguments in an interesting way, as if it could take the properties of both a resumptive pronoun and a trace at a time. Consider the following well-known data:

(10) Who did you say t was there yesterday?

(11) *Who did you say that t was there yesterday?

(12) Chi hai detto ? c'era ieri?

(13) Chi hai detto che ? c'era ieri?

The first two examples exemplify a phenomenon which was attributed great importance in mainstream generative theory: empty categories are not allowed to be governed by a complementizer. These observations gave rise to the so-called empty category principle (ECP). However, in (at least some) languages which allow *pro* as subject, this does not seem to obtain, as can be seen from the Italian examples.

The solution to this puzzle proposed by Rizzi ([56],p.142) is that in Italian, INFL is a proper governor for empty categories in some cases. This accounts for pro-drop and for (13) at the same time: and ECP violations can be circumvented by first moving the subject into a postverbal position, and then extracting it. This accommodates the two observations, which are intuitively interrelated, within the framework and reduces them to a particular property. Nonetheless, it is somehow unsatisfactory as an explanation in an intuitive sense.

We have the intuition that the explanation is the fact that an Italian core clause is always well-formed without subject, and therefore, for such sentences, one can always put an NP somewhere into the periphery, which is then interpreted as the subject via some binding mechanism. In Chomskyan terms, we do not need to assume a t in the subject position of the extraction clause; there could also be a *pro*. But of course we cannot know what is the underlying structure of a sentence as (13).

More generally, we can ask for all *pro*-drop languages and "extractions" from a clause c, whether there is an invisible resumptive pronoun in the core clause c, thus making the extraction "improper". And whatever answer we chose, it will not be an empirical one: the answer will rather depend on theory internal considerations (i.e., which way do we get the best generalizations given our approach and additional assumptions). From this fact we might draw the consequence: maybe we need not even ask the question. Speakers of Italian, after all, will probably not feel the necessity to represent their language in a manner which is equally well-suited for English. Their theories of their language might be simply underspecified in this regard; if both concepts do not make different predictions, why should they care? And even if they would, how could they learn the correct structure? On the other side, if languages with *pro* behave differently than languages without *pro*, there is no point in reducing their behavior to the categories we use for languages without *pro*. At some point, we will have to express the difference in some way - as Rizzi did for Italian - but this seems rather of matter of mastering a particular theoretical machinery, not one of linguistic interest.

We conclude that considerations on resumption are not very meaningful if resumptive (and therefore unmarked) pronouns are invisible anyway. But actually, we can put the intuitive explanation into a form which is very meaningful for formal linguistics. In English, the clause from which the argument is extracted, is not a well-formed clause on its own. We therefore have a slight context-dependence, though it can be easily modelled by context-free grammars (as in GPSG). In Italian, things are more simple: the extraction-clause *is* well-formed, and therefore does not need a special licensing procedure. The extraction itself can be accounted for rather in terms of binding than in terms of movement and government, which is well-known to be much more liberal: and this for the reason that we do not *need* any additional syntactic rules to license the substructure E:

(14) Chi hai detto che [è venuto]$_E$?

To return to our above definition, we would say that the Italian standard grammar trivializes subject extraction (as a grammatical rule). Weak trivialization is quite obvious; do we also have strong trivialization? For this we have to encode the trace coindexation as a binding relation, which seems easy to do. This way, we see that issues of (strong) trivialization are relevant in order to capture linguistic generalizations. We see that extraction and empty categories directly interact with syntactic complexity. Of course our treatment was sketchy and intuitive. We think it would be worthwhile to make this intuitive sense precise and look whether it allows for a cross-linguistic generalization (after the generalization of Rizzi was shown to be typologically false by Newmeyer, [54]).

2.8 Ontologies of Linguistics and their Construction

2.8.1 On the Semantics of Linguistic Theories

To prevent possible misunderstandings: we are not concerned with semantics of natural language at this point, but with the meaning of linguistic theories. So what is the domain in which linguistic theories are interpreted?

In our perspective, "language", as the proper subject of linguistics, is the *semantics* of a linguistic theory. So given a linguistic theory, we usually have to also give a sort of interpretation, which specifies how we get from the theory to "language", that is, its model. This might be intended in analogy to the technical, logical sense; but we here intend a larger meaning, yet without a general technical formal specification. For example, if we approach language in a logical fashion (see [37],[58]), then "language" comes close to the model of the theory in the technical sense (though it is not the same, as in the model-theoretic approach, the language is the class of all models of a theory). If we take a theory to consist in a phrase structure grammar, then its model is the formal language it describes, which is obtained in the usual way as the language generated by the grammar.

In this general sense, doing linguistics is constructing theories for a model. It is the task of the metalinguist to construct a *model* for linguistic theories, which is satisfying in both the empirical, theoretical sense and in the sense of effectiveness of construction. In this section we will informally discuss what we think are

possible conceptions of "language", put differently, possible linguistic ontologies, which satisfy all our requirements, and what are possible relations of the ontology of linguistics and the ontology of metalinguistics. This discussion will remain quite informal, and is mainly understood to provide a good motivation of the mathematical notions and techniques developed in the next chapter.

As we have said, linguistic theories have to be intensional in a sense, as their range is the possible, not the actual. In a technical sense, however, theories are usually not intensional: they simply denote infinite sets, and creativity is simply interpreted as infinity. One of our main points will be, though, that "languages" do not need to be infinite sets; and that, consequently, linguistic theories might have a different semantics.

We will now describe 3 different approaches to linguistic metatheory, which we will for simplicity also call metatheories. The first one is the *classical* metatheory, which is a formalization of the canonical approach. The second one is the intensional metatheory, which is somewhat exotic in the sense that its conception (to my knowledge) does not occur in the literature. It is more complicated than the classical one, but helps to solve many intricate problems of modern linguistics. The last one is the finitary metatheory, which bears some resemblance to a position sometimes hold by linguists with a strong focus on empirical data; but also might comply theorists who reject many common idealizations of standard linguistics (see [67] for a recent and explicit discussion).

2.8.2 The Classical Ontology and Its Problems

As we said, in the classical (we could also say: standard) approach to linguistics, "language" is nothing but an infinite set. So the task of classical meta-linguistics is to get an infinite set out of a finite set. This is the classical "projection problem". The projection has to depart from a finite language. So we take an observed language, and map it onto an infinite language. The most appealing property of this approach is that it is so simple. In order to study its mathematics, the only thing we need to study is the properties of maps $f^* : \Sigma^* \to \Sigma^*$, such that for $I \subseteq \Sigma^*$, $I \subseteq f^*(I)$. For our purposes, we can restrict the domain of these maps in a way such that the domain is always finite; but we have to make sure, that at least for some subset of the domain, the range is infinite. So the advantages of the classical approach are obvious; less obvious are the drawbacks, which we will discuss now.

The first limitation to this approach is the following *fragmentation-problem*: as we can only project an *observed* language, rather than the observable language, this approach always remains preliminary, or rather, fragmentary, because o-language is, by assumption, infinite. Say that a "language" is **complete** (with respect to an o-language), if it contains this o-language; it is **fragmentary**, if it does not.

The problem is: when we project a finite dataset, we never know whether the result is complete or not, whether it is the one which gives us the entire and proper "language". Say that an observed language I is **relevant** wrt. an o-language O and projection f^*, if $f^*(I) \supseteq O$, that is, $f^*(I)$ is complete wrt. O. In principle, there is the possibility that we got a *relevant fragment*, that is, a fragment the projection of which yields a complete "language". As we do not have a finite description of o-language, however, there is no way to be *sure* about it.

This is obviously a problem for any meta-theory of language. But in the classical approach to language, it comes with another, more fundamental problem. Let I be an observed language; $f^*(I) = L$ be the projection of I. Now assume $w \notin I$; and we observe the utterance w after we have performed our projection. Now there are two cases: *case 1*: $w \notin L$. Obviously, we have to accommodate it in our "language" and our theory. But before, we have to accommodate it in our *pre-theory*, that is, we have to consider it for our projection. Now, the simplest approach would be to say that our language is simply $L \cup \{w\}$. However, this seems methodologically wrong: w should be part of the observed language, and therefore should be considered in the projection. We want a meta-theoretical justification also for the new language containing w, because we do not want the order of observation to influence our language as an artefact. Now, *a priori* there is nothing to make sure that $f^*(I \cup \{w\}) = L \cup \{w\}$, and not even that $f^*(I \cup \{w\}) \subseteq L \cup \{w\}$. In fact, $f^*(I \cup \{w\})$ might look very different from $f^*(I)$; we might even have $f^*(I \cup \{w\}) \subsetneq f^*(I)$. So the problem is that in the worst case, if we make a new observation, our projection will be a language very different from the old one, and it might happen that most of our *theory*, that is description of the language, turns out to be worthless, and we have to start all over again. But what is even more startling is *case 2*, where we have $w \in L$. In that case, we could say: well, all is fine, as we have predicted w to be in our language, and so our projection even got confirmed in a sense. But this comes with problems of its own: assume we would have observed w *before* we performed the projection. Maybe $f^*(I \cup \{w\})$ looks entirely different from $f^*(I)$, even though $w \in f^*(I)$. And this is really problematic, because the time and order in which we make our observations affects what "language" looks like! This problem strikes now with full generality: once we have projected our language, *any* new observation might bring us into this trouble. On the other side, we should expect to observe the strings we predict to be in "language"!

A similar problem might come up for the following reason: it is well-known that there is only a limited agreement on linguistic judgments, that is, whether certain utterances belong to a language or not; this holds especially in many interesting cases. The question is: given the many "borderline" cases of grammaticality, do these cases affect the "core" language[8], that is, the language which results from the projection of the language on which judgments generally agree? Whereas the problem before was one of changes which arise from adding new strings to an observed language, now the problem is one of taking away strings, or more generally, a problem of intersecting observations in order to provide agreement. Unfortunately, in general intersections might affect the projection in ways which cannot be predicted; we have no generally valid statements on this problem. So again, small changes in the data might cause unpredictable consequences.

Both problems will be addressed in the mathematical section; as we will see, we can devise meta-theories such that they partially solve the problems; on the negative side, we can show that there *cannot* be a satisfying solution for all of these problems.

Finally, there is another point which might be considered problematic. In the classical ontology, the properties of "language" depend on the properties of

[8]Note that we use the notion "core" here in a different sense than generative grammarians do!

projections in most relevant aspects. This is to say, after all we can say little about real language in the theoretical sense, because all of its major properties are given to it by pre-theoretical assumption. Moreover, once we have adopted a given projection, there is no way to redeem (or discharge) our assumptions: because from the point of view of formal language theory, there is little of interest we can say about a finite language; on the other side, everything we say about "language" depends on our projection. We call this problem the *circularity problem*, because statements on formal properties of infinite "languages" are circular, in the sense that they depend on our own assumptions. We will later on see that there is a way to get around this, by defining universal properties *modulo* a pre-theory.

2.8.3 The Intensional Ontology and its Motivation

To introduce the intensional ontology, we will first discuss why one would consider that an infinite set is not an adequate conception of "language", and not an adequate ontology for linguistics. Later on, we will discuss the formal problems following from alternative conceptions, and how one might solve them. The main linguistic problem with the classical ontology is the following: take the (fairly standard) assumption that "language" is a model of what speakers know (though not necessarily of their linguistic representation). If we then assume that "languages" are infinite sets, we make some very strong commitments:

1. For all possible utterances, a speaker knows at any given point whether the utterance is part of his language or not.

2. A speaker knows all utterances in his language in the same way.

Why would we possibly not accept the first commitment? We might want to state that for some problematic utterances, whether or not a speaker accepts them, might depend on the speaker making some reasoning (similar to the linguist). The result of this reasoning might strongly depend on the speaker himself, but also on the evidence he considers (or even the evidence at hand to him), and therefore is essentially non-deterministic: not even for a single speaker, we might be able to say in general whether he "knows" a certain utterance at a certain point. We might rather think, that him accepting or refusing will be the result of a process we cannot entirely predict. To illustrate this with some small example, consider the following critical case:

(15) people^4 see^4

Asking a speaker whether this is a sentence of English or not might bring us different answers: a speaker might answer "I do not understand this utterance (immediately), therefore it is wrong".[9] He could also say: "thinking about it, I understand it; yet, nobody would ever write or say it, so it is wrong". Or he could, as a linguist, say: "This sentence is okay, because it conforms to the rules of English grammar". One might object that this is the linguist talking, and therefore the statement is not "naive", and therefore invalid. But the intensional linguist can answer: how do you know? Anyone can reason like this; and on the

[9] Often, psycholinguists shorten the procedure methodologically by showing sentences only a short time. This amounts to allowing only this answer.

contrary, it might be naive to think of something like an absolutely naive speaker as a person who does not think about his language at all. In the sequel, we will argue that there is good evidence that even the most naive speakers reason about their language. There is even a forth possible answer of the speaker to the above question: "I do not know what the question means: do you mean the sentence is comprehensible or good according to rules?" So judgments essentially depend on reasoning and choices we make. A second example is the following. We consider the following famous "grammatical illusion":

(16) More people have been to Russia than I have.

Any linguist will agree that this sentence is wrong. As concerns the speakers, whereas the immediate, "intuitive" judgment is that the sentence is fine, a bit of thinking brings most speakers to the conclusion that it is wrong. So also in this case, the presumed "knowledge" changes with reasoning. If we think that judgments get more adequate, the more we reason about them, this can be fine; but this is exactly the contrary to what we usually think as linguists, being out for the immediate judgment. The usual way out of this dilemma is that we say: *as linguists*, we see the sentence as wrong, whereas speakers should judge it as fine because of whatever reason (which is irrelevant to proper knowledge of language). But, judging the sentence as wrong, how do we know it is the linguist in us talking, not the speaker thinking about it? After all, there is no principled difference between the two, and no formal criterion of superiority for the former. To put the point more generally: we as linguists do not know, in how far we are linguists and in how far we are just speakers thinking about language; and for the "normal" speakers, we do not know in how far they might act as linguists in their judgments, even though not being professionals.

This was the main critique for the first commitment. Why would we reject the second commitment, that is, that all utterances of a language are known in much the same way? Essentially, the reasons are similar. We can argue that our linguistic knowledge is *partial* in the following sense: there are some things we know for sure; our judgments are immediately available, so to speak. Other facts we only "know" mediately: we can arrive at a certain judgment in certain reasoning steps which derive it. But in these derivations, we make assumptions, similar to the linguist who makes assumptions on language when he projects it to the infinite. But there are two important things to consider: to have an immediate judgment on something is different from deriving a judgment under certain assumptions: because there are many different things we can assume, and the conclusions of our reasoning might be different every time we use different assumptions (compare the above discussion). Secondly, if we are not able to make a certain judgment of the form $w \in L$ at a certain point, this does not mean that we are unable in general: we might be able to derive it in a certain way we have not yet found. To but it simply, in the intensional view, the absence of positive knowledge is *not* the same as negative knowledge.

Now, from a conceptual point of view this might sound very appealing, but the question is: what should "language" then look like? We will approach this problem in a more detailed fashion later on, and provide here only a sketch of a possible solution.

In the intensional ontology, we do not interpret "creativity" as blunt infinity; we rather construct a model of language which is intensional in a more proper

sense. This is achieved in the following way: "language" is a structure over finite languages. We depart from a finite (observed) language, which we gradually extend; all extensions at a given point are finite languages; but there are infinitely many extensions. So the structure of language is essentially a tree, whose root consists in observations (*cum grano salis*, as we will see), edges are possible inferences on given assumptions, which connect languages with larger languages.

The linguistic ontology which underlies intensional "language" is the following: we assume a (finite) set of utterances which speakers know *immediately*, that is, without any reasoning. We can assume that we know them *verbatim*, because it is a finite set. We will call this set *i-language*, which is to say, the immediate language. From immediate knowledge, there is lots of knowledge we can derive, using certain analogies and linguistic inferences (so essentially the same mechanisms of the classical metatheory, only that they now form part of linguistics). A certain branch in the tree corresponds to a certain line of reasoning, connecting a set with a superset. It is here the infinity comes into play, and this way we redeem the creative commitment. But importantly, a intensional language is not an infinite set, but an infinite structure over finite languages.

Call a function f on a set of sets majorizing if $I \subseteq f(I)$ for any I of the domain. More formally, an intensional language is a structure $(I, \{f_i : i \in J\}, \{I_j : j \in J^*\})$, where I is is a language, the f_i are majorizing functions from (finite) languages to (finite) languages, and the set $\{I_j : j \in J^*\}$ is a set of languages, which is defined as follows: (1) $I \in \{I_j : j \in J^*\}$, where $I = I_\epsilon$; and (2) if $I_l \in \{I_j : j \in J^*\}$, $f_i \in \{f_j : j \in J\}$, then $I_{il} = f_i(I_l)$, and thus $I_{il} \in \{I_j : j \in J^*\}$. This is to say that each language carries as an index the "reasoning" by which it has been derived. We leave it open whether the set J is finite or infinite; for practical reasons it will remain finite in the sequel, but in principle, it needs to be only finitely specified. If J is non-empty, then J^* is infinite, but in order to provide us with infinitely many distinct I_j, the functions and/or I need to satisfy some additional requirements, which we will consider in the sequel.

So in this model, creativity is interpreted in an intensional manner, not in an extensional one, and results in a structure rather than a set. We might also say that creativity in this interpretation is *transcendent*, as speakers are only creative if they reason and thereby *transcend* their basic knowledge. Contrarily, the classical conception is *immanent*, in that we assume that all possible creativity is already present and fully specified in the knowledge of the speaker.

This approach solves the problems we have stated in the beginning: we clearly distinguish between more or less immediate knowledge. We also abandon the claim that speakers know everything at a given point, because our model of knowledge is intensional. We furthermore have a more reasonable position with respect to the last circularity problem, because we always remain aware of the assumptions we make in deriving a certain string. So this model of language has some quite appealing properties. But how does it relate to standard linguistic theory? Obviously, linguistics in the intensional sense would look entirely different from the linguistics we are used to. We will try to work this out later on; currently, we only make some clarifications.

A new notion we have introduced is the one of an i-language, the set of utterances we know *immediately*. Note that also under the intensional requirements, it is perfectly fine to treat this language as a set. It would be tempting to equate

i-language with an o-language or an observed language. But both seem wrong for several reasons. The first one is: i-language is supposed to be a cognitive notion, not a methodological one as o-language. Secondly, we cannot equate i-language with o-language, as the first one is necessarily *finite* – we cannot immediately know an infinite language – , the second one is necessarily infinite. Regarding the relation of i-language and observed language, there are problems in both directions. For the usual reasons, we cannot know whether we have observed all utterances of an i-language, therefore we cannot say that a certain observed language includes a certain i-language. But also in the other direction, we might observe utterances which are *not* in i-language, that is, some utterances which are only derived knowledge. After all, our intensional model does not prevent speakers from using language beyond i-language! So the only thing we can really say for sure is i-language must be contained in o-language, because all utterances which speakers know immediately are clearly observable in principle. On the other hand, o-language might contain some (infinite) branches of the tree, which coincide with the reasoning we can perform online.

Note that this means that our ontology is more complicated, but also somewhat richer; in particular, this complication gives us considerably more flexibility: whereas in the classical approach, we always have to base our projection on a set of observed data; and as the data changes, the projection changes. In the intensional ontology, "language" is based on an i-language, which need *not* coincide with the observed data. So assume we observe a string $w \notin I$, which is derivable by a certain line of reasoning, that is, is in some I_j. In the classical approach, we have to reconsider projection given this string. In the intensional approach, we do not: we simply assume it is not in *i-language*, but is simply some derived knowledge. As we will see later on, this can make a huge difference and simplify things a lot for us.

2.8.4 The Finitist Conception of "Language"

There is a third alternative conception of "language, which we call *finitist*. The finitist conception is based on the following assumption:

(17) "Language" *is* o-language.

This is a very strong assumption, which has some immediate consequences. The first and most important one is:

(18) Natural languages are contained in the regular languages.

The reason is that human language processors are finite; therefore, we cannot observe a language which is not recognized by some finite state automaton; therefore, natural languages are regular languages. This is a claim which runs against most linguists assumptions and intuitions, as in general natural languages are assumed to be not even context-free. So the assumption (19) is a high price to pay, but it comes at considerable gain, as we will see.

In order to proceed, we have to introduce a notion of **being falsifiable by finite languages**. This slightly corresponds with finite model property in logic, and we use this notion in a quite similar, though less technical sense. As we said, a linguistic theory is a description which denotes a certain model, which in turn might supposed to be a formal model of what we observe as language.

There is of course a more general notion of linguistic theory, namely in the sense of a class of possible theories, or, using the word *grammar* instead of linguistic theory, a linguistic theory in this more general sense is a specification of possible grammars.

Whereas the model of a grammar is supposed to be the formal analogue of a natural language, a linguistic theory in the more general sense specifies a class of possible grammars, and thereby possible models. So the class of models of a class of theories is the formal analogue of the class of languages which we consider to be *possible*. We can call a theory in this more general sense a **framework**. Theories/grammars are what we need in order to make statements on particular "languages"; a framework is what we need to make statements on *all* possible languages. Linguists are usually interested in both of them; and in the classical/intensional metatheories, both topics are addressed equally. In the finitary setting, as we will see, there is a slight focus on the second topic, namely making statements on the nature of all languages. For us the question is: can we do linguistics *without* projecting a language into the infinite? Is it possible to reasonably work with finite languages? We will here try to work out how a positive answer might be given.

First of all, we need to recall how the linguist works when he tries to make statements valid for *all possible* languages. Therefore, we must first distinguish *positive* statements of the form "all natural languages are context-free", and negative statements of the form "not all natural languages are context-free". A linguist can easily make statements of the negative form (though for a statement as the one above, he has to perform a projection!); but he will not be able to make positive statements of the positive form, except as a *working hypothesis*; because he does not know all languages; and even if he knew all actual natural languages, he would not know all possible natural languages (on this dilemma, see [54]). So all the linguist can do on this behalf is to *falsify* certain claims, and to *corroborate* certain conjectures by our inability to find falsifying evidence. So what the general linguist does is something like falsificationism. This falsificationism is the key to finitary linguistics.

For the finitist, the fact that the "languages" we work with are finite does *not* mean that also our *theories* have to be finitary in the sense that they denote finite "languages". On the contrary, we cannot assume this, because this would be clearly in conflict with the creative commitment. So what we do is: we consider finite languages, and write grammars for infinite languages which cover them.

But of course there is a problem to that: if we do not project languages to the infinite, adequacy becomes somewhat trivial, at least given the usual frameworks. For example, a claim like: "not all natural languages are context-free" cannot be made. The reason is that in formal language theory, regardless which descriptive devices we use, all normal frameworks (in the sense of classes of languages/grammars) contain the class finite languages. This means in particular, that given a finite language which is not projected, we cannot falsify a given framework, and therefore, we are unable to make any general statements on the nature of language via falsificationism. So without an effective projection, there is no empirical content to the construction of linguistic theories in the usual frameworks. like context-free or mildly context-sensitive grammars. We *can* falsify theories (grammars) by considering new data, but we cannot falsify the entire framework, and this is what is needed to make general statements on

language.

To make the process meaningful, we first introduce the notion of finite language property (FLP). Let FT be a framework; for simplicity, we assume that $FT = C$, where C is some class of languages. We say C has the finite language property, if there is a finite language L such that $L \notin C$.

The main proposal I want to make here is: we can do finitary linguistics, but only within frameworks which have something like finite language property. Translating this back into terms of linguistics: if we do finitary linguistics, we have to assume that the class of possible languages (formal counterparts of natural languages) does *not* contain the class of finite languages (FLP). As we will see later on, this is still slightly inadequate, and there are some additional subtleties to consider; but for this we need some more mathematical background. In this sense, we say that a class of languages has the FLP if it does not contain the class of finite languages.

A side comment is in order: not doing projection at all does not mean to do no idealizations in order to obtain abstract models of natural languages. In fact, there are plenty of idealizations which this meta-theory presupposes; but these are not the focus of this work.

Now things get interesting again. In the finitary setting, it is the linguists task to write his grammars according to a framework with FLP. Now, what the linguist is doing is essentially falsificationism: considering always new data, he is trying to falsify his framework, and thereby he is able to make some general statements on the nature of language, in much the same falsificationist fashion of the general linguist.

Note that the finitary linguists methodology is different from both in the classical and the intensional. In the latter paradigms, we had an explicit procedure for constructing infinite languages from finite ones (though not necessarily as infinite sets). We could therefore say that we had a given object which we had to describe adequately. In linguistic finitism, this is not the case: we do not perform a task of projection. This means that the "proper subject of linguistics" remains a finite object. At the first glimpse, this seems to conflict with our basic tenet of describing infinite objects, but this is not the case. The trick is that the finitist writes grammars for infinite languages, but being agnostic about their model, and then takes a falsificationist stance. So it is falsificationism which bridges the gap between finite and infinite, rather than projection. Both the classical and intensional linguist devise their theories on a given set of data they project; that means in particular, once pre-theory is done and theorizing starts, new data is no longer considered. And if the two have to consider new (relevant) data, then they have to start over, that is, once again they have to accommodate the data in the pre-theoretic procedure, before they can again start theorizing. So for these two schools, considering new data is *not* part of theorizing proper, but part of the larger process of pre-theory and theory.

For the finitist, things are different: for him, studying new data is an *essential part of theorizing*; without this, his theorizing would be vacuous. The finitist writes grammars/theories for infinite languages, but has only finite languages at hand. So by definition, there is a discrepancy between the the subject of his theorizing, and his theories. Put differently, he does not even construct "language". In particular, the adequacy of his theories/grammars is heavily underdetermined: everything which covers the finite fragment is fine in principle. This is why we have additional criteria: sub-regularity, FLP etc. But still, this

is insufficient. What we also need is that in theorizing, the finitist always has to check whether his theory is still adequate as new data comes in.

Note that this shows some similarity of the intensionalist and the finitist position: whereas the intensionalist considers a structure of *possible* utterances, which models the speakers growing knowledge according to new inferences, the finitist has to consider always new *actual* utterances, which comes to him not by construction or inference, but by experience. So, what we are the formal tools of finitary linguistics? We get the following easy observations:

Lemma 5 *There is no smallest class C of languages (contained in the regular languages) which contains all finite languages and some infinite language.*

Lemma 6 *There is no largest class C of languages (contained in the regular languages) which does not contain the finite languages.*

Both observations are quite obvious, so we omit the proof. What does that mean for us? There is no clear upper/lower bound for the class of languages we should consider. Whereas in principle, FLP does not provide us with an upper bound for the class of languages we are interested in, our finitistic philosophy does: as the regular languages are the largest class of languages which can be recognized by finitary means, there is no non-regular language which is observable, as this would necessarily involve projection anyway. But of course, the class of regular languages does not have FLP, so it is not a candidate; that even holds for smaller classes such as the star-free languages.

As in the finitary setting there is no proper construction of "language" and we always stay within the observations, there is a great gain we have: we simply avoid the entire problem of projection, circularity etc., sticking to a rigid falsificationism. We construct theories for infinite languages only considering observations. However, the price we have to pay will be considerable to most linguists.

Note that the idea to treat natural language in a subregular fashion is by no means new: see for example [55]. It has not found much support in the community, because from the point of view of formal linguistics it is not very appealing. Nonetheless, from the point of view of linguistic metatheory it is quite appealing. If we connect it with the notion of trivialization, introduced above, then it might become even a fruitful field of study in connection with the "classical mathematics of language", that is, the formal methods used in classical linguistics.

2.8.5 Finitism in a Broader Sense

One might argue that there is a broader sense in which we might accept the finitist commitment, which says that "language" is o-language. This is the following: we allow for projection, but when we choose a projection π, we have to make sure that for each finite language I, $\pi(I)$ will be a regular language; in particular, if we think of I as a set of observations being part of an o-language, then $\pi(I)$ has to be contained in this o-language – put differently, every string in the projection has to be observable in principle, or still differently: our projections have to preserve acceptability. We can call this approach **broad finitism**, whereas the former position can be called *narrow finitism*. Note that broad finitism is strongly related to the classical conception.

What we will show here is that the two positions, though theoretically distinct, in practice often coincide, and if not, narrow finitism seems to be preferable. The main reason is the following: assume they do not coincide. That means, we do not take a falsificationist stance based on something like FLP within broad finitism. But from the point of view of complexity, all we can really say about "language" in general is that it is regular - which it is by assumption. So this is quite a vacuous way to deal with natural language. If we want to make it more meaningful, we have to look for tighter upper bounds for "language". But again, this can be only achieved properties similar FLP; because if our framework contains the class of regular languages, there is no way to falsify it. So broad finitism as well has to recur to falsificationism if it is supposed to bring us interesting generalizations.

Conversely, assume $\pi(I)$ is a regular language. Then we can also in *narrow* finitism write a grammar for $\pi(I)$, and try to falsify it. This is, however, not exactly what happens in broad finitism: because making a new observation would lead us to reconsider entire the projection rather than falsify the grammar – this would be the same as in the classical metatheory. So the differences between broad and narrow finitism are mainly technical; and to put it bluntly, broad finitism seems to unify the drawbacks of narrow finitism - restriction to regular languages - with those the classical metatheory, which are all about the problems and difficulties of projection. After all, the main advantage of narrow finitism is that we avoid the critical step of projection altogether. Therefore, broad finitism is not of too much interest for us now.

There is another fundamental doubt I have about broad finitism: given a projection π, some dataset I, how can I ever know that $\pi(I)$ is contained in o-language? After all, o-language is an empirical notion, and even the fact that $\pi(I)$ is regular does not entail it is in o-language, as should be clear. Moreover, as $\pi(I)$ is supposed to be infinite, we can never test whether all its strings are acceptable/observable. So the position of broad finitism, which I have found sympathetic to many linguists, to me seems to be quite problematic and not worth elaborating at this point. I will however treat some mathematical questions which arise when we want $\pi(I)$ to be contained in o-language, with some surprising results (see subsection "On regular projection", chapter on classical metatheory).

As a final note, I am aware that there are still many different positions on the metatheory of language, and in fact much more than I can mention here. But I hope the ones I have outlined are the most important, reasonable and interesting ones.

Chapter 3

The Ontology of Metalinguistics

Summary of the Metalinguistic Ontology

The most fundamental datum of metalinguistics are judgments of the form $\vdash \vec{w} \in L$. But we here justify a more elaborate ontology; in particular, we introduce *negative data*. This negative data is however of a fundamentally different nature: it is not an empirical datum, but rather constructed by the argument: if we accept this, we have to accept anything. The reason we cannot use negative judgments as primitives is that this way, we would become way too much negative data: we usually want explicitly more than any speaker accepts. The positive and negative data give rise to we call *partial languages*: pairs of finite languages, their intersection being empty. The goal of metalinguistics is to complete them to full infinite languages; but for this purpose we can only use the positive data, which is given, not the negative data, which is a construction: apart from methodological reasons, we still need the negative data to check whether our pre-theories are adequate, that is, whether they agree with our intuitions, because if we do not have this method of control, we have no other and projections become quite arbitrary.

3.1 Preliminaries

As we have said, there are different possible conceptions of what "language" is. We will work out three fundamental positions. Each of them is distinguished by fundamental differences in the basic ontology of linguistics. In this introduction, we do not work on the particular ontologies of linguistics, but we first try to sketch an ontology for *metalinguistics*; that is, we want to give a formal inventory the metalinguist is given, and what he has to provide. This ontology is then basic and common for all three metatheories: so they are based on the same metalinguistic ontology, though they construct different linguistic ontologies. Whereas we already introduced the notions of "language", "o-language", observed language (data) etc., we will here look at more formal foundations for the mathematical procedures we use.

For now, the most important part of metatheory is the part which (meta-)linguists are given pre-theoretically. Contrarily to the linguistic ontology, this ontology must be *finite*, because we do not observe infinite languages or infinite objects in general. Given this finitary ontology, it is the task to develop formal procedures which develop an adequate ontology for linguistics. In this way, we get infinite "languages", out of our observations (or in the finitary setting, grammars describing infinite languages). Keep in mind that "language" is only a shorthand for "whatever we consider the proper subject of linguistics", it is thus a formally underspecified notion. This is the crucial step of linguistic metatheory: we formalize a procedure, which given some finite input, gives something infinite – namely "language" – as output. In the simplest, classical case, it is nothing but a function; in the intensional case, there are some additional non-deterministic choices to be made. So we have to determine the input of the function. But this is not all a metalinguistic ontology has to provide. We would like in addition to have some **criteria of adequacy** for metalinguistic procedures. That is to say, we want some non-trivial criteria to tell whether a procedure does a good job or not. There are two kinds of criteria: there are *a priori* criteria, which apply to the procedure regardless of any input, that is, they concern abstract intrinsic

properties of pre-theories. These should be in line with intuitions linguists have on "language", and its relation to the observed language. Secondly, there are empirical criteria, which help to decide whether a pre-theory does a good job with respect to a given input. That means, it should tell us whether the pre-theory agrees with the intuitions of a linguist observing the data, whether it is convincing to him. We will see that in order to allow for testing empirical adequacy, we will need to enrich our ontology beyond simple finite languages. Adequacy in this latter sense is not a general property of metalinguistic procedures. It rather depends on the particular linguistic object to which it is applied. We find exactly the same distinction in linguistics, where we can refuse a theory because, for example, in general it is undecidable, but also because it cannot adequately handle some linguistic phenomena.

We now want to construct a sufficiently rich ontology to be able to check procedures for empirical adequacy, though in the sequel, that is, when we actually develop the procedures, we will not be concerned with the empirical part itself, but rather with intrinsic formal properties.

3.2 Linguistic Judgments

A notion which is very fundamental for us is the one of **linguistic judgment**. Though we only use it technically, we should briefly explain its theoretical importance. Given an utterance w, we denote the linguistic judgment sustaining that w is in a language L by $\vdash w \in L$. So the difference between utterances and judgments is similar to the difference in logic between formulas and judgments, where the judgments sustains the truth/validity of the formula. We will be concerned with linguistic judgments mostly in a technical way, as we develop some different "proof theories" for these judgments, to keep up the analogy with logic. So philosophical concerns about what these judgments really are, what their true nature is, will not be a major topic of this work. Nonetheless, we should be clear about what these judgments mean. Importantly, they are not speech acts or anything similar, and in this sense, they are not "natural data" for linguistics, as the term is currently used by empirically motivated linguists. They are not the judgments which speakers make when they speak. They are the judgments linguists make, and the judgments speakers make when they are asked by linguists whether a sentence is correct/well-formed or not.

So a linguistic judgment is a judgment on an utterance, and it thus comes with a metalinguistic flavor. For us, linguistic judgments are the "**canonical datum**", that is, the most fundamental notion of (meta-)linguistics, and a notion which cannot be analyzed further. How does this contention go with the position, nowadays taken by many scholars, that the priority has to be given to something like "natural data"? Because whatever the latter means, linguistic judgments are not natural data, this is clear. My point is the following: if someone would claim the priority of "natural data", I would reply: so assume you want to describe German. You hear somebody speak, or take some written utterances. How do you know they are not French, or English? How do you know they are of any relevance to what you call German? After all, they might be French, we do not know! The reply to this might be: well, that is my knowledge which tells me the utterance is German; and my reply is: well, then you just transformed a piece of "natural data" into a linguistic judgment. By this argument, I think we can use

linguistic judgments as a basic and fundamental notion of linguistic metatheory. Metalinguistic procedures will be thus concerned with deriving judgments from judgments. What we need as premises, obviously, are sets of judgments, which correspond to languages.

3.3 Partial Languages

We have already mentioned the important notions of metalinguistics: a finite observed language, an infinite observable language (o-language), and "language". We have also mentioned that according to some views, it is not strictly necessary or desirable that "language" comprises o-language (see [22] on "grammatical illusions"); but by the same argument, one can argue that "language" does not comprise the observed language. This seems to us a complication we cannot take into account in metalinguistic procedures, as it would open the door to complete arbitrariness. Rather, if the linguist wants an observation not to figure in "language", he has to exclude it from the observed language by not making a linguistic judgment. So for us, things are as follows: observed language is a subset of observable language. We want to construct "language" from observed language in a way such that it comprises o-language. That is of course not an empirical criterion: we do not know o-language, because though observability is empirical, the whole language is infinite and therefore never actually observable in its entirety. So observable refers to its members, not the whole language. So this is rather just a theoretical requirement. Moreover, the reader should keep in mind that we usually *properly* include o-language: we usually have objects in "language" which are not observable.

So far there are three objects, only one of which is given. But as I already pointed out, a simple finite language does not provide us with a satisfyingly rich ontology, because it only tells us what we need to find within a "language", but give us no information on what we do *not* want to find within a "language". We therefore introduced a forth object, the so-called **negative language**, which is also finite. The status of the negative language is very problematic: from a metalinguistic point of view, all we know are positive judgments, and our commitments are – except for the finitary approach – such that we explicitly want to derive utterances which are not judged to be acceptable/observable in the first place. So we have to be careful: we cannot just consider negative evidence as it is given to us, we have to distinguish between "proper negative data" and improper one. So how can we justify this? Note that in the finitary setting, we also will need negative evidence for falsification, but it is much easier to obtain: if "language" is just o-language, we can anything which is not judged grammatical as negative evidence. For the other cases, we have to distinguish. A typical case would be the two examples

(1) John Mary love

and

(2) People people people see, see, see

The former should probably not be in "language", the latter should. But this is surely not an empirical distinction! I would make the following point: we

have to make a theoretical distinction between utterances of which we would say: they *might* be derivable, and those of which we say: they should not be derivable. But still the problem remains: how do we tell the difference? Here, we have to recall that we are doing metalinguistics rather than linguistics, so firstly we cannot *know* whether an unobservable utterance is derivable, because if at all, this holds by definition. Conversely, if we want to say that an utterance cannot be in "language", we would say that it cannot be derived by any amount of linguistic reasoning. Here we see the circularity: we do not yet have a formal notion of linguistic reasoning, but we are only on the way to formulate one!

The way out of this dilemma is the following: for a certain utterance w, it would be fine with us if it belongs to "language", because we are agnostic about it. For another utterance v, we might say: if this v belongs to "language", then really anything can belong to "language", and if this is the case, then there is no way to test empirical adequacy. So if there is any set which should – to our judgment – not be contained in "language", then it should comprise v. And this is our justification for the negative language: it is the only alternative to arbitrariness.

This gives, of course, a fundamentally weaker status for the negative language. Therefore, we agree that we will not use this language for metalinguistic procedures, but only for testing their adequacy. We could also do otherwise, but this would result in an entirely different paradigm, which would correspond to learning with both positive and negative data. This might of course be worthwhile looking at; but we will not do so for two main reasons: 1. on the one side I think the negative data is too weak for these purposes; and 2. on the other side, if we use the negative data for projection, we have used in up in the sense that we cannot use it anymore for checking adequacy of pre-theories. We thus lose the only instance of control, and any projection will be adequate. The latter is the most convincing reason for keeping negative data out of projection, because only this way, we have the possibility to check whether a projection agrees with our intuitions or not.

Note that of course, introducing some utterance into the negative language always requires an explicit linguistic judgment, and consequently, the negative language is always finite. Having now an observed language and a negative language, this introduce the concept of a **partial language**. Given a finite alphabet Σ^*, we define a partial language as follows:

Definition 7 *A partial language L_p over alphabet Σ is a pair of sets strings of strings (L_1, L_0), such that $L_1, L_0 \subseteq \Sigma^*$ are both finite, and $L_1 \cap L_0 = \emptyset$. Given a language $L \subseteq \Sigma^*$, we say $L_p = (L_1, L_0)$ is partial wrt. L, if*

1. *$w \in L_1 \Rightarrow w \in L$, and*
2. *$w \in L_0 \Rightarrow w \notin L$.*

If L_p is partial with respect to L and L is infinite, then we say L is a completion of L_p. By $comp(L_p)$ we denote the set of all completions of L_p. If L_p is partial wrt. L and L is finite, then we say L is a refinement of L_p; denote the set of all refinements of L_p by $ref(L_p)$.

This notion requires some explanation. Maybe the best way to think of a partial language is to think of it as a *partial characteristic function*. In this

way, the notion of completion becomes very clear. Partial languages are thus a model of *partial knowledge* of language. There are some (finitely many) strings of which we know they are in a language, some (finitely many) of which we know they are not in the language, and infinitely many of which we do not know. If we think of a completion L of a partial language L_p, we can think of it as a pair $(L, \overline{L})$, that is the language and its complement. Obviously, this presentation is redundant; it is however helpful to make clear that a completion is genuinely more informative than a partial language. The underlying idea is that for the creative commitment we have to assume that knowledge of language is partial as long as it comprises only finite sets; as soon as we redeem the requirement of infinity, there is no reason to assume this. Note the analogy with intuitionistic/constructive mathematics: there are true statements, wrong statements, and there are statements of which we do not know. The epistemic aspect of language is thus made explicit in the ontology. As soon as we have infinity, we can leave out our concerns regarding epistemology, and assume that our knowledge is complete.

The notion of refinements is of no importance for the classical metatheory, but crucial for the intensional metatheory, which is considerably more complicated, as we assume that knowledge remains partial. We use refinements in constructing non-classical ontologies for linguistics, that is, ontologies where languages are not just simply infinite sets.

This is the formal notion of what we are given as (meta-)linguists. One might also object that our basic assumptions are to *weak*, in the following sense: we know a lot more about language, such as degrees of (un-)grammaticality etc. The answer to this objection is simple: everything that can be obtained by our ontology can be *a fortiori* by obtained with a richer ontology, so this does not bother us.

So the procedures of metalinguistic theory are the following: 1. In the classical case, we construct partial languages from observations. At a certain point, we have to decide that our partial language is rich enough, and we *project*, according to a certain procedure, the positive language, which is an observed language, to a language which is hopefully infinite. Thereby, we construct "language". Then we check the resulting object for adequacy with respect to the partial language: it must not contain any of the objects in the negative language. 2. The intensional case is somewhat more complex: given a finite positive language, we have to decide which portion of it is contained in i-language, and which not. This, in my view, is nothing our formal apparatus should do, but the decision should rather be based on linguistic considerations; so we allow the (meta-)linguist some choices, which can be based on whatever information is available to him. In the finitary approach, things are properly different: as we renounce to projection, we always remain with the partial language. We then use it directly to falsify linguistic *theories* rather than metalinguistic procedures, to which we renounce.

Chapter 4

The Classical Metatheory of Language

Summary of the Classical Procedure

The classical procedure of constructing "language" works as follows: we devise a projection π (later on: pre-theory $(\mathfrak{f}, P)$); next, we gather some positive data I_1, and construct some negative data I_0. Then we use the projection and project our positive data; finally, we use the negative data to check whether the projection π is adequate. If it is, we have constructed "language". If not, either the projection was bad, and we have to find a new one; or the data was bad, and we have to gather more positive information and/or construct less negative data.

To illustrate the restrictions of information that we have, we can conceive of this procedure as a sort of "game" between *linguist* and *metalinguist*: the linguist gathers I_1 and constructs I_0. Then he hands only I_1 over to the metalinguist, who uses π to construct $\pi(I)$, which he gives back to the linguist. The linguist then controls for adequacy of π. In case π turns out to be inadequate, he can either change his (positive and negative) data and repeat the procedure, or look for another metalinguist.

Note that either way, it is not legitimate that we (or the linguist) disregard any positive data: whatever we collect has to be used for projection. As a consequence, it is not defined in this procedure what happens if the projection is adequate but afterwards the linguist makes a new observation: in principle, he has to repeat the entire procedure, thereby constructing a new, possibly different language, even if the observation was part of the constructed language.

4.1 The Classical Metatheory

As we said, in the classical conception, "language" is an infinite set. This means, the classical metatheory is concerned with mapping finite languages – we only project positive data – onto languages. This is the core of the classical metatheory, and the rest of this section will be concerned with maps of this kind. In this introduction, we will shortly sketch a bigger picture, which still is simple enough. The full metalinguistic procedure works as follows: we collect (positive) data, and construct negative data, thereby obtaining a partial language (I_1, I_0). Now comes an important point: at a certain point, we decide *explicitly*, that (I_1, I_0) is sufficiently rich for our purposes. This is very important, because it is the point which makes the classical metatheory different from classical formal learning in the sense of Gold: in this paradigm, utterances keep coming in, and at some point we converge towards the correct, underlying language (if we learn it). For the classical metatheory that does not make sense: for either we never know whether we have already converged or not, or we have to know the language in advance. So the classical procedure is not about learning in the limit and convergence: rather, we consciously choose a certain point (this is the decision of the linguist), and then *project* the language. So we really only need a function from finite languages to languages. When we have performed this projection, we have to check whether our projection was adequate: first of all, we check whether the result is really an infinite language; and secondly then we can use our negative language to test adequacy of this language.

So far the simple procedure. There are some things we have to add. The first thing is: we can interpret this procedure in two different ways. In the first way (which is stricter), if we make a new observation, we have to repeat the entire

procedure of projection, because the finite language we depart from changes. This even obtains if the observation is in the resulting "language". This would be the correct position, because the resulting "language" could have been quite different if we would have departed from the other observation. We could also take a more relaxed position and consider a new projection necessary only if we observe something which is not in our "language". This is however not very correct, because in the end it means that with the same observations I, we can have different "languages", depending on when we made the projection! This is clearly a bad thing, so we just mention this position. Either way, making new observations is never part of the metalinguistic procedure, and quite problematic. Whereas there is no really good solution on this conceptual level, we will try to implement a solution in the mathematical formalism, to make it invariant under new observations which have been predicted; this is possible, though it comes with quite some commitment, as we will see.

A second remark is the following: of course, we can interpret the resulting objects in different ways: we can say it is what speakers "know" ("cognize"), or it is an abstract object etc. So as far as its ontological nature is concerned, our metalinguistic procedure is quite agnostic. However we make our projection, the procedure has only solved one problem: "language" is now defined in a unique and mathematically precise way. Still, it is an object we define by stipulation of a projection, and all in all it is not an empirically testable object. So for example, if we assume "language" is what speakers know, we have to live with the fact that there are many ways to define it, none of which is preferable on empirical grounds. This is what gives the major point of criticism of the classical approach.

4.2 Introducing Pre-Theories

The classical metatheory is thus mainly a theory of mappings from finite languages to languages, which satisfy some properties. These mappings have to satisfy some additional properties. What we have mentioned in particular is: metalinguistic procedures must be finitary. This has an important consequence for their mathematical formalization: given a finite input language, projections must compute a *finite representation* of the output language in a *finite number of steps*. So they are not properly maps from (finite) languages to infinite languages, but rather maps from (finite) languages to finite representations of languages, which are furthermore computable in the standard sense. As we have to distinguish a language from a representation (for example, a grammar generating it, we introduce the operator L, which maps a representation of a language onto that language. That is quite vague at this point, but we cannot be more precise as long as we are lacking concrete projections. There are also some additional requirements, which are given in the following definition:

Definition 8 *A projection is a map π, such that for any finite alphabet Σ, $I \subseteq \Sigma^*$, we have*

1. *$\pi(I)$ is a finite representation of a language, such that that there is a computable map L, and $L(\pi(I))$ is the language $\pi(I)$ represents;*

2. *$I \subseteq L(\pi(I))$;*

3. *if I is infinite, then $L(\pi(I)) = I$;*

4. *π is computable, that is, if I is finite, then $\pi(I)$ can be computed in a finite number of steps;*

5. *there are some finite languages I such that $L(\pi(I))$ is infinite.*

So the images must be at least recursively enumerable and π must provide an enumeration procedure. Furthermore, we need some infinite languages as images. There are some more conditions which are very reasonable to require from a projection. Let Σ, T be arbitrary alphabets. A string isomorphism is a map $i : \Sigma^* \to T^*$, such that 1. $i(\epsilon) = \epsilon$, 2. for all $\sigma, \sigma' \in \Sigma$, if $i(\sigma) = i(\sigma')$, then $\sigma = \sigma'$, and 3. $i(\sigma\vec{w}) = i(\sigma)i(\vec{w})$.

Definition 9 *A projection π is **reasonable**, if for any Σ, $I \subseteq \Sigma^*$,*

1. *$L(\pi(I)) \subseteq \Sigma^*$,*

2. *if i is a string isomorphism, then $\pi(i(I)) = i(\pi(I))$, and*

3. *for any finite alphabet Σ with $|\Sigma| \geq 2$, there is an $I \subseteq \Sigma^*$ such that $\pi(I)$ is infinite.*

These conditions should also be clear: we do not allow π to "introduce" new letters, as these cannot be justified by patterns; also, π is closed under isomorphism, which means that it treats all letters equally. The third condition is a strengthening of the last of the first definition: it requires that the the cardinality of the alphabet does not play a role for projection, except for the unary case which is somewhat special. For example, we could have a projection which only has infinite images for input languages over an alphabet of cardinality 5. This however is quite odd. But again, this only holds for $|\Sigma| \geq 2$, because the case where $|\Sigma| = 1$ is very problematic: there are many patterns which simply cannot be observed in languages over one letter. Intuitively, one can see the correlation with the fact that we can encode any alphabet in a binary alphabet, but not in a unary alphabet.

This however still underspecifies the projections we are interested in; we are interested in projections to which we can give some linguistic meaning, which can be thought of as a formal analogue of what we have called linguistic reasoning. Our main tool to describe projections of finite languages are **pre-theories**. In the sciences, a pre-theory is a fixed way to interpret observations. This is necessary to bridge the gap between the observations, finitary in nature, and the theory itself, which is supposed to account for infinitely many phenomena.

As we will see, it is quite difficult to give a general definition of what counts as a pre-theory. Rather than starting with an incomprehensible definition, we will first present a simple example of the most important notions. The reason for this difficulty is mainly that what we consider to be linguistic objects strongly differs between different approaches to language. The most simple language-theoretic universe will only comprise strings; but more elaborate approaches will also use sets of strings as basic objects, then terms over functions from strings to strings, and finally we will use a type-theoretic encoding of strings as our universe.

We will from now on use the following convention: finite languages are usually denoted by letters I, J (possibly with subscripts), whereas L (possibly

with subscripts) is generally used for infinite languages. This convention will be redundant in the sense that we will be explicit about cardinalities of languages if it really matters, but I hope it will enhance readability. Furthermore, in many cases cardinality will play no technical role, but we will use this convention to indicate whether the definitions are mainly intended and relevant for finite or infinite languages.

Fix a finite alphabet Σ and assume $I \subseteq \Sigma^*$. A **simple string-based analogical map** P is a map $\wp(\Sigma^*) \to \wp(\Sigma^* \times \Sigma^*)$. If we have $(\vec{w}, \vec{v}) \in P(I)$, we also write $\vec{w} \approx_I^P \vec{v}$, such that $\approx_I^P \subseteq \Sigma^* \times \Sigma^*$ (we use the two statements equivalently). We will then write that the two words $\vec{w}, \vec{v}$ are P-similar for the pre-theory P. We will, unless stated otherwise, assume that the languages in the domain of pre-theories are finite. We say a simple string-based analogical map is **reasonable**, if and only if the following hold:

1. if I is infinite, then $P(I) = \emptyset$,
2. P is recursive (membership in the set $P(I)$ is decidable), and
3. for any string isomorphism i, $P(i(I)) = i(P(I))$.

The first conditions makes the case of infinite pre-images of pre-theories uninteresting. The intuition underlying analogical maps is that they tell us which substrings of a language are similar (not only in distribution). In the sequel, we will only work with reasonable analogical maps (as long as they are simple string-based).

For any pre-theory P and language I, we assume $\approx_I^P$ to be symmetric, as it is a similarity relation. We therefore also say that $\vec{w}$ and $\vec{v}$ are P-similar in I, if $\vec{w} \approx_I^P \vec{v}$. Actually, as similarity is clearly a reflexive concept, we could also require that P be reflexive. This would not change anything, but it would often amount to a trivial case which we have to mention explicitly, so we mostly require that P be irreflexive, just to make things simpler. There is a third property which is often attributed to similarity, namely transitivity. This is the most arguable of the three, because it is intuitively valid, but quickly leads to problems of the kind of the Sorites paradox: a number of small changes, each one preserving similarity, can lead to arbitrarily large changes. Therefore, transitivity for us is not an important criterion, but we will see that there are both transitive and intransitive similarity maps.

Next, we introduce the two notions of **analogy** and **inference**: An analogy consists in asymmetrically equating two objects, which are P-similar for a pre-theory P, where by "asymmetrically equating" we mean: ascribe all properties of one object to the other (it is clear that this is an asymmetric process for distinct objects). Importantly, this is only an example scheme: we might have more (or less) premises. What is general about this scheme that it infers an **analogy**, and uses P-similarity as a premise.

$$\frac{\vec{a} \approx_I^P \vec{b} \quad \vec{a} \neq \vec{b}}{\vec{a} \Leftarrow_I^P \vec{b}} \tag{4.1}$$

So this infers an analogy $\vec{a} \Leftarrow_I^P \vec{b}$, which in turn is crucial for inferring linguistic judgments. For this, we need other types of inferences. Let $f : \Sigma^* \to \Sigma^*$ be a function from strings to strings. A quite general example scheme for inferences

then is as follows; note that also here, we might have more premises; we will even encounter different conclusions.

$$\frac{\vdash f(\vec{a}) \in L \quad \vec{a} \Leftarrow_I^P \vec{b}}{\vdash f(\vec{b}) \in L} \tag{4.2}$$

This reads as follows: from an analogy $\vec{a} \Leftarrow \vec{b}$ and a judgment that $f(\vec{b}) \in L$, we can derive a new judgment, which consists in stating that $f(\vec{a}) \in L$. The interesting thing is of course the question what f does, which in a sense determines the property we talk about (so whereas f is a function, $f(\vec{x}) \in L$ is a property, so to speak parameterized by the function). The simplest and most obvious instantiation of f could be the identity, so that we just speak about being part of the language, as $f(\vec{a}) = \vec{a}$. An inference then looks as follows:

$$\frac{\vdash \vec{a} \in L \quad \vec{a} \Leftarrow_L^P \vec{b}}{\vdash \vec{b} \in L} \tag{4.3}$$

There are of course many other possibilities, and we will just look at the most natural and/or linguistically meaningful ones. Another reasonable choice would be the following: the property we talk about is that strings occur in the same contexts in the language; that is, for some context $(\vec{w}, \vec{v})$, $f_{(\vec{w},\vec{v})}(\vec{a}) = \vec{w}\vec{a}\vec{v}$. An inference then looks as follows:

$$\frac{\vdash \vec{w}\vec{a}\vec{v} \in L \quad \vec{a} \Leftarrow_L^P \vec{b}}{\vdash \vec{w}\vec{b}\vec{v} \in L} \tag{4.4}$$

Whereas the function f seems to be suitable to illustrate the concept of linguistic inferences from a didactic point of view, it would complicate things considerably if we would use it when we formally set up a calculus for inferences. We will therefore not use it later on. A pre-theory then simply specifies an analogical map P and a set of inference rules $\mathfrak{f}$; the two combined allow us to make analogies and inferences on finite languages. We now give the definition of pre-theories for the general case; for all particular instances of pre-theories we will present, we will be able to give more restrictive definitions, which we present at due time.

Definition 10 *A **pre-theory** is a pair $(\mathfrak{f}, P)$, such that*

1. *P is an analogical map, such that for any alphabet Σ, there are some sets M, N, such that $P : \wp(\Sigma^*) \to \wp(M \times N)$ maps languages onto relations over M, N,*

2. *$\mathfrak{f}$ is a class of inference rules of the form $(\overline{M}, \phi)$, such that*

   (a) *$\overline{M}$ is a finite sequence of statements, ϕ is a single statement;*

   (b) *for any alphabet Σ, $\vec{w} \in \Sigma^*$, $I \subseteq \Sigma^*$, there is a rule $(\langle \vdash \vec{w} \in I \rangle, \vdash \vec{w} \in \mathfrak{f}_P(I)) \in \mathfrak{f}$,*

   (c) *for $P : \wp(\Sigma^*) \to \wp(M \times N)$, $(x, y) \in M \times N$, there are rules $(\overline{M}, \phi) \in \mathfrak{f}$ such that $(x, y) \in P(I)$ is among the statements in $\overline{M}$.*

This definition is extremely general: it only requires that we can extend linguistic judgments from $\vec{w} \in I$ to $\vec{w} \in \mathfrak{f}_P(I)$, and use some analogies as premises. Due to the lack of restrictions, our inference rules are proper classes, we cannot even construe them as sets. An important notion is the following: we call the **language** of the pre-theory $(\mathfrak{f}, P)$ the set or class of statements ψ which figure in some inference rule $(\overline{M}, \phi) \in \mathfrak{f}$, which do *not* have the form $(i) \vdash \vec{w} \in I$, (ii) $\vdash \vec{w} \in \mathfrak{f}_P(I)$, (iii) $(x, y) \in P(I)$ or (iv) $x \Leftarrow_I^P y$; we refer to this language with $L_\mathfrak{f}$.

We just give this definition for the sake of generality, and work with more restricted versions. For the beginning, we will consider pre-theories based on strings only; these can be given a quite restrictive definition. The notion of a structure is well-known from model-theory (for a classical presentation, consider [3]; for a presentation independent from logic, consider [18]). We now introduce language-theoretic structures as follows:

Definition 11 *Let Σ be an alphabet. A **language-theoretic structure** $\mathcal{M}$ is a tuple $\langle \Sigma^*, \cdot, \rangle$, where $\cdot : (\Sigma^*)^2 \to \Sigma^*$ is the concatenation operation.*

We thus have a structure over domain Σ^* with the distinguished function of concatenation. We denote the class of all these structures by **LTS**. Next, we introduce a signature for classical first-order logic, denoted by FOL. We take the usual inductive definitions of terms and well-formed formulae over $x : x \in X, \wedge, \neg, \exists x : x \in X$, where X is a countably infinite set of variables. We also grant ourselves the equality with its usual syntax, and a binary function symbol $\star$, such that we work with the logic $\mathsf{FOL}(=, \star)$, that is, first-order logic with equality and a binary function symbol $\star$. We interpret equality always as extensional equality, and $\star$ as concatenation. Let $\mathcal{M} \in \mathbf{LTS}, \phi \in \mathsf{FOL}(=, \star)$; we write $\mathcal{M} \models \phi$ if ϕ is true in $\mathcal{M}$ under the usual definitions, interpreting $\star$ as $\cdot$. By $\phi(x_1, ..., x_i)$ we denote a formula with exactly i free variables. We say $\mathcal{M} \models \phi(x_1, ..., x_i)[\vec{w}_1, ..., \vec{w}_i]$, if $\phi(x_1, ..., x_i)$ is true in $\mathcal{M}$, if each x_j is interpreted as $\vec{w}_j$.

We now define the **relational language** of a simple string based pre-theory, which we call $L_\mathfrak{f}$. Such a language is characterized by a pair of a finite sequences $(\langle P_1, ..., P_i, \rangle, \langle \phi_1, ..., \phi_i \rangle)$, where for $1 \leq j \leq i$, P_j is a relation symbol of arity n_j. We call these symbols the **relational signature** of $\mathfrak{f}$. Moreover, for $1 \leq j \leq i$, ϕ_j is a $\mathsf{FOL}(=, \star)$-formula. We say that ϕ_j *corresponds to* P_j, and if P_j has arity n_j, then ϕ_j has exactly n_j free variables $x_1, ...x_{n_j}$.

We now define the relational language $L_\mathfrak{f}$ as follows: ψ is in $L_\mathfrak{f}$, characterized by $(\langle P_1, ..., P_i, \rangle, \langle \phi_1, ..., \phi_i \rangle)$, if and only if it has the form $P_j(\vec{w}_1, .., \vec{w}_n)$, where $1 \leq j \leq i$, the arity of P_j is n, and $\vec{w}_1, .., \vec{w}_n \in \Sigma^*$ for some alphabet Σ. We say that $P_j(\vec{w}_1, .., \vec{w}_i)$ is **true**, if for $\phi_j(x_1, ..., x_n)$ the formula corresponding to P_j, there is an $\mathcal{M} \in \mathbf{LTS}$ such that $\mathcal{M} \models \phi(x_1, ..., x_i)[\vec{w}_1, .., \vec{w}_i]$.

We usually leave the definitions of such languages implicit, and use relation symbols such as $\sqsubseteq$ for substring, corresponding to the formula $\exists yz.\star(\star(y, x_1), z) = x_2$, and we write $\vec{w} \sqsubseteq \vec{v}$ instead of $\sqsubseteq (\vec{w}, \vec{v})$. We reserve $' ='$ for the (extensional) equality, and $' \neq'$ for the inequality. The reason for this slightly complicated definition is that on the one hand we do not want to commit ourselves *a priori* to an alphabet; but on the other hand we neither want to figure any meta-language in our inference rules, but rather proper strings! We therefore define the relational language over arbitrary alphabets, so that the relational language

itself is not a language in the proper sense of a set. Still, apart from that, the language is very restricted: syntactically, we only allow simple statements of the form $P(\vec{w}_1, ..., \vec{w}_i)$, and semantically, the truth of such a statement is restricted via definability of relations in $\mathsf{FOL}(=, \star)$ in some language-theoretic structure (see e.g. [17] on definability within these structures). The simple reason for these restrictions is: it works just fine for our simple string-based pre-theories. As we will introduce more complex pre-theories, we have to adapt these notions, for examples to signatures containing more functions than just $\star/\cdot$, and even to the universe of λ-terms with $=_{\alpha\beta}$ congruence instead of strings etc. We will however not spell these definitions out as we do here, as the underlying concepts should be clear and it is nothing but an exercise in formalizing mathematics to formal logic.

Definition 12 *A **simple string-based pre-theory** is a pre-theory $(\mathfrak{f}, P)$, such that*

1. *P is an analogical map, such that for any language, if $I \subseteq \Sigma^*$, then $P(I) \subseteq \Sigma^* \times \Sigma^*$,*
2. *$\mathfrak{f}$ is a set of inference rules of the form $(\overline{M}, \phi)$, where ϕ is either*
   (a) *a linguistic judgment of the form $\vdash \vec{v} \in \mathfrak{f}_P(I)$,*
   (b) *a statement $\vec{w} \approx^I_P \vec{v}$ (equivalently, $\vec{w}, \vec{v} \in P(I)$) or $\vec{w} \Leftarrow^I_P \vec{v}$ for some pre-theory P,*
   (c) *or a statement $(\vec{w}_1, ..., \vec{w}_{n_i}) \in R_i$, which is part of a fixed relational language $L_\mathfrak{f}$,*

   and $\overline{M}$ is a finite sequence of any of these kind of statements.

We will henceforth use $(\mathfrak{f}, P)$ as a meta-variable for pre-theories, whereas particular pre-theories will have different names. We have still defined $\mathfrak{f}$ in a very general manner, thereby allowing many non-sensical pre-theories. But it is important that there is a fixed (relational) language which labels our trees, and for which there is some notion of truth; we will refer to this language as $L_\mathfrak{f}$. So whereas in the general definition of pre-theories, we define the language $L_\mathfrak{f}$ in terms of the pre-theory, in the case of more restricted pre-theories we first fix a (relational) language, which then restricts the pre-theory. The reason for this becomes obvious in the definition of a $(\mathfrak{f}, P)$-derivation, where we need a notion of truth of a statement. We now give the definition of a derivation for pre-theories in full generality; it basically is the usual definition of transitivity of deductive inference, with some additional features regarding the leaves of a proof-tree.

Definition 13 *The set of $(\mathfrak{f}, P)$ derivations is the smallest set such that*

1. *If $\vec{w} \in I$, then $\overline{\vdash \vec{w} \in I}$ is an $(\mathfrak{f}, P)$ derivation of $\vdash \vec{w} \in I$.*
2. *If ϕ is a true statement of the (relational) language $L_\mathfrak{f}$ of $\mathfrak{f}$, then $\overline{\phi}$ is an $(\mathfrak{f}, P)$ derivation of ϕ.*
3. *If $(x, y) \in P(I)$, then $\overline{(x, y) \in P(I)}$ is a $(\mathfrak{f}, P)$ derivation of $(x, y) \in P(I)$.*

4. *If* $(\langle \phi_1, ..., \phi_n \rangle, \psi) \in \mathfrak{f}$ *is an inference rule,* T_1 *is a* $(\mathfrak{f}, P)$ *derivation of* ϕ_1*,....,*T_n *a* $(\mathfrak{f}, P)$ *derivation of* ϕ_n*, then*

$$\frac{T_1 \quad ... \quad T_n}{\psi}$$

is a $(\mathfrak{f}, P)$ *derivation of* ψ*.*

Some notes are in order. Firstly, we now see why the definition of the relational language and its semantics have been slightly complicated: we want to have a notion of truth, but a language over arbitrary alphabets. We cannot – of course – give a general definition of truth for arbitrary statements, and therefore it is up to any particular pre-theory to define its notion; and therefore we have given the definition only for the simple string-based case. Secondly, in virtue of definition 10, if $\vec{w} \in I$, then

$$\frac{\overline{\vdash \vec{w} \in I}}{\vdash \vec{w} \in \mathfrak{f}_P(I)}$$

is an $(\mathfrak{f}, P)$ derivation of $\vdash \vec{w} \in \mathfrak{f}_P(I)$. Moreover, in virtue of this definition, we can derive any true statement of the relational language. We could also have implemented this directly in the inference rules, but to me it seems preferable to have a notion of *truth* figure in a derivation rather than the inference rules. Of course, it is only 4. which gives the calculus its proper strength by transitivity of inferences: we can derive any tree, as long as the local subtrees are well-formed according to the inference rules in $\mathfrak{f}$. 1., 2. and 3. serve to get the correct premises for inferences. So the set of $(\mathfrak{f}, P)$ derivations are defined almost as the usual derivations in proof-theory, with one important difference: the statements (labels) of the leaves of a derivation tree are not written out explicitly in the inference rules $\mathfrak{f}$, but are defined independently of the calculus.

As a consequence of this definition, we can use $(\mathfrak{f}, P)$-derivations in order to define a map $\mathfrak{f}_P$, which for any alphabet Σ is a map $\wp(\Sigma^*) \to \wp(\Sigma^*)$, which is defined by

$$\mathfrak{f}_P(I) := \{\vec{w} : \text{ there is an } (\mathfrak{f}, P)\text{-derivation of } \vdash \vec{w} \in \mathfrak{f}_P(I)\}. \tag{4.5}$$

Mathematically, this is a deductive closure. This definition also allows us to conceive of $\mathfrak{f}_P$ as a function $\mathfrak{f}_P : \wp(\Sigma^*) \to \wp(\Sigma^*)$, which for any I yields $\mathfrak{f}_P(I)$. This is the way we will think of $\mathfrak{f}_P$ in the sequel. It cannot be checked in general that $\mathfrak{f}_P$ defines a reasonable projection in the above sense: for example, the condition that there is a finite language with an infinite image under $\mathfrak{f}_P$ depends very much on the details of the pre-theory. Note that we assume that for $\vec{w} \in I$, the linguistic judgment $\vdash \vec{w} \in I$ "comes for free".

So the finite languages we project determine the derived languages in two main ways: firstly, by the set of analogies they allow us to make, and secondly, by providing us with a set of premises for our inferences. The two factors are quite different in nature. On the one side, they are too strongly entangled to allow us to consider them separately. On the other side, they are to loosely related to allow us to make strong inferences from one to the other in most cases. This was the cause of some headache during writing what is to follow; yet I

do not see any other reasonable alternative to this "double usage" of observed languages both for linguistic judgments and analogies. This also motivates some additional notations. Let $A \subseteq \Sigma^* \times \Sigma^*$ be a relation (ontologically equivalent to P-similarity). Then we put

$$\mathfrak{f}_A(I) := \{\vec{w} : \text{ we can derive } \vdash \vec{w} \in \mathfrak{f}_P(I), \text{ by allowing } \vec{u} \approx_I^P \vec{v} \text{ as a label iff } (\vec{u}, \vec{v}) \in A\} \tag{4.6}$$

That way, we can choose analogies and premises in I independently. By definition we usually have $\mathfrak{f}_P(I) = \mathfrak{f}_{P(I)}(I)$. This will turn out to be useful in some proofs later on.

To sum up, there is considerable freedom in the instantiation of properties, as there is in the choice of linguistic objects. But there is one crucial constraint: $(\mathfrak{f}, P)$ is a proper pre-theory, if and only if $\mathfrak{f}_P$ is a map from finite languages to (some infinite) languages, where we conceive of it as a function as explained above. So pre-theories uniquely define maps, which should be projections. And if they do, they can be tested for adequacy.

We will now consider some interesting or illustrative pre-theories, which we divide into some general classes, according to the properties we use. Unfortunately, at this point I do not see how these pre-theories can be ordered within a clear taxonomy; but there are several properties or parameters by which we can distinguish them. The first one is the nature of the language theoretic objects we talk about. We can have simply strings, bracketed strings which encode some structure, but we can also use sets of strings or even sets of terms which encode string-like entities. (This will be the main principle for the order of presentation here). The second main parameter are the properties we talk about, in the sense of $\vdash f(w) \in L$, as explained above. In the end, of course, we want to derive linguistic judgments from linguistic judgments. But on the way we can use many different tools, from simple substitution to recursive functions on strings. The third main parameter consists in the criteria for establishing the relation $\approx_L^P$ between two objects in a given language L. This is the most fine-grained one, but as we will see, in some sense it cuts across the boundaries of the language theoretic objects, in that we can have very similar or equivalent criteria on different classes of objects. To give an overview, we present a table which considers only the first two parameters; note that the classes of languages written do not mean that we induce all of their members, but only provide an upper bound to the class of induced languages.

	Strings	Sets of Strings	Sets of n-Tuples	Terms	Sets of Terms
Substitution	undecidable	uninteresting	uninteresting	uninteresting	uninteresting
Structured Substitution	CFL	CFL	n-MCFL	does not work	open
Functions	PTIME?	not covered	not covered	not covered	not covered
Membership	gives completeness	uninteresting	uninteresting	uninteresting	uninteresting

A word of explanation; I start by describing the labels. The first row describes the objects we talk about, our "ontology", so to say: we can just talk about strings (first column), certain sets of strings (second column), sets of tuples of strings etc. So our judgments have the form $\vdash \vec{w} \in I$, $\vdash M \subseteq I$ etc. The first column describes the property we preserve over analogy; so, for the first row, if we have an analogy $\vec{w} \Leftarrow \vec{v}$, then we infer $\vec{x}\vec{v}\vec{y} \in I$ from $\vec{x}\vec{w}\vec{y} \in I$, that is, we substitute. In the second row, we rather infer $\vec{x}(\vec{v})\vec{y} \in I$ from $\vec{x}\vec{w}\vec{y} \in I$; in the last row, we just infer $\vec{v} \in I$ from $\vec{w} \in I$. The label "gives completeness" in the

last row means: "allows us to simulate any projection", a result to which we refer as completeness, because it shows that pre-theories come with no loss of generality wrt. projections.

Having said this, I have to add: the above table is only an attempt to provide an intuitive taxonomy of the pre-theories we consider. This is to say: in their technical definition, the changes work differently and not according to this scheme. For example, in using sets of strings rather than strings, we have to provide additional inference schemes in order to derive judgments of the form $\vdash M \subseteq I$ from sets of judgments of the form $\vdash \vec{w} \in I$, and vice versa. So the table does not exactly reflect our formal treatment, but rather underlying intuitions. Secondly, also the classes of languages figuring as an upper bound for the languages which are induced have to be taken with care: they hold for the pre-theories *we have considered* under this intuitive rubric. It does not mean that for all pre-theories which fall in this rubric the same will hold. Also, we have to explain what we mean by uninteresting, as this means different things in different rows. In the first row, I explain this judgment as follows: a simple substitutional pre-theory on strings turns out to induce languages which are not decidable. This is for me an argument to not investigate substitutional pre-theories on more complex objects. It does not follow that these will also be undecidable, but still the treatment of the string-based substitutional pre-theories makes sufficiently clear that these will not be very interesting. In the last row, we explain this as follows: we show our completeness result – for every projection there is an (extensionally) equivalent pre-theory – by means of the string based pre-theory over simple membership. This is the only reason why I consider the inference over membership interesting: it is hard to give it any linguistic meaning or motivation. As this gives us the desired result, there is no reason to consider the other ontologies, so I label them uninteresting. The reason I have not covered the pre-theories with inferences on functions beyond strings is that they do not seem to work in the desired way; so this does not seem to be promising to me, but of course I cannot tell whether they are really uninteresting.

So as we can already deduce from the table, we will look at the most important properties of pre-theories using only the simple string based pre-theories as examples, and establish thereby some exemplary results. For pre-theories which have a more complex ontology we will not be equally specific, in order to keep this dissertation in a manageable size, and in particular because the results, which are partly already complicated to obtain in the simple case become much more complicated in more general cases and pre-theories. What we will do, however, is to indicate how certain reductions to the string based case can be put to work – or can no longer be put to work – if we lose for example freeness of the underlying monoid. So we go very much into depth for the string based pre-theories, and then present extensions together with results on (im)possibility of reductions.

4.3 Substitutional Pre-Theories

The substitutional pre-theories comprise probably most of the pre-theories which would be considered linguistically interesting, though not necessarily all of them, in particular not from a transformational perspective. The reason is the particular role substitution plays in linguistic theory: natural languages are

generally assumed to have underlying structures, and these structures often seem to be a grammar-theoretic *pendant* of the language-theoretic notion of substitution (but as we will see and most people know, this is true only under some additional assumptions). We will stick with this, as it is our goal to find a formal foundation for linguistics that linguists would judge to be adequate. But it is important to keep in mind that many things which at the level of linguistics seem to have the status of observations, at the level of metalinguistic have the status of mere assumptions, and so does the presumably structural nature of natural languages. So from a metalinguistic point of view, this choice is by no means without alternatives, and later on we will in fact we will in fact consider very interesting alternatives (or rather: extensions).

In the treatment of pre-theories, whether based on strings or not, it is much easier for presentation to depart from a given alphabet; therefore, we will adopt this convention. Note however that pre-theories are defined independently of alphabets. So take a (finite) alphabet Σ and a language $L \subseteq \Sigma^*$. We now introduce two relations over $\Sigma^* \times \Sigma^*$, which will be fundamental for what is to follow.

1. Write $\vec{v} \sqsubseteq \vec{w}$ if and only if for some $\vec{x}, \vec{y} \in \Sigma^*$, $\vec{x}\vec{v}\vec{y} = \vec{w}$;

2. $\vec{w} \leq'_L \vec{v}$ if and only if $\vec{x}\vec{v}\vec{y} \in L \Rightarrow \vec{x}\vec{w}\vec{y} \in L$.

The first one is anti-symmetric, reflexive and transitive, as can be easily checked. It is quite self-explaining, and will be referred to as (contingent) substring relation. To illustrate the second one, for $I := \{ab, a\}$, we have $a \leq_I ab$, but not $ab \leq_I a$. The second relation is a pre-order, that is, it is reflexive and transitive, but not anti-symmetric. Reflexive is obvious, transitivity follows from the transitivity of the logical implication by which it is defined; it is also antisymmetric *modulo* the distributional equivalence $\sim_L$. Given a language L, we call a string $\vec{w}$ *trivial in* L, if there is no $\vec{v} \in L$, such that $\vec{w} \sqsubseteq \vec{v}$; triviality means the string has no occurrence in any word in L. Denote the set of substrings of $\vec{w}$ by $fact(\vec{w})$; we extend this notion to sets on the natural way and write $fact[L]$. $\vec{w}$ is then trivial in I if and only if $\vec{w} \notin fact[I]$.

As $\leq'_L$ is defined by an implication, we allow any trivial string on its right hand side. To avoid complication arising from this, we will assume that $\leq_L$ is the restriction of $\leq'_L$ to strings which are non-trivial in L, that is, $\leq_L = \leq'_L \cap (fact(L) \times fact(L))$. Now to continue the example: if we have nontrivial strings $\vec{w}, \vec{v}$, $\vec{w} \neq \vec{v}$, $\vec{w} \sqsubseteq \vec{v}$ and $\vec{v} \leq_L \vec{w}$, then we can deduce that L is necessarily infinite, as can be easily shown by iterated substitution. Showing that a language is necessarily infinite is a type of argument which we will encounter quite some times in what is to follow. To make these arguments neat, we need always the restriction to non-trivial strings, so this is another good reason for excluding trivial strings.

Now, given a finite language I, non-empty strings $\vec{w}, \vec{v}$, we put $\vec{w} \approx_L^{P1'} \vec{v}$ iff

1. $\vec{w} \sqsubseteq \vec{v}$, and

2. $\vec{w} \leq_I \vec{v}$;

$\approx_I^{P1}$ is the reflexive closure of $\approx_I^{P1'}$.

This defines our first, simple analogical map $P1$, by putting $P1(I) = \{(\vec{w}, \vec{v}) : \vec{w} \approx_I^{P1} \vec{v}\}$, provided I is finite. We define ʃ1 as a set of inference rules over the

relational language defined by the structure $\langle \Sigma^*, \neq \rangle$, so what all statements of the relational language have the form $\vec{x} \neq \vec{y}$. There are two rule schemata, which, to enhance readability, we immediately write in the form of trees:

$$\frac{\vdash \vec{w}\vec{y}\vec{v} \in \mathfrak{f}1_P \quad \vec{y} \Leftarrow_I^P \vec{x}}{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}1_P(I)}, \tag{4.7}$$

$$\frac{\vec{x} \approx_I^P \vec{y} \quad \vec{x} \neq \vec{y}}{\vec{x} \Leftarrow_I^P \vec{y}} \tag{4.8}$$

Note that the P on the trees is a variable for analogical maps: they can be used with arbitrary maps (as with arbitrary languages); the rules just make sure the identities are preserved. Same holds for I and string symbols; so actually these two schemes serve as a shorthand for infinitely many rule instances. We will have a short look at an example to see this pre-theory at work:

Example 14 *Take $I := \{ab, a\}$. Clearly, we have $P1(I) = \{(a, ab), (ab, a)\}$. Therefore, we have $\mathfrak{f}1_{P1}(I) = a(b^*)$. To show how our calculus works, we will show this result in a some detail. For example, consider the following derivation:*

$$\dfrac{\dfrac{\dfrac{\vdash ab \in I}{\vdash ab \in \mathfrak{f}1_{P1}(I)} \quad \dfrac{a \approx_I^{P1} ab \quad a \neq ab}{a \Leftarrow_I^{P1} ab}}{\vdash abb \in \mathfrak{f}1_{P1}(I)} \quad \dfrac{a \approx_I^{P1} ab \quad a \neq ab}{a \Leftarrow_I^{P1} ab}}{\vdash abbb \in \mathfrak{f}1_{P1}(I)} \tag{4.9}$$

By this example it is easy to see that $a(b)^* \subseteq \mathfrak{f}1_{P1}(I)$; to see the inverse inclusion, consider that we cannot derive anything else by means of the analogy $a \Leftarrow_I^{P1} ab$; and moreover, we cannot derive anything else by the inverse $ab \Leftarrow_I^{P1} b$ either: all derivable strings are already in the language. This can be easily seen in considering the following example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash ab \in I}{\vdash ab \in \mathfrak{f}1_{P1}(I)} \quad \dfrac{a \approx_I^{P1} ab \quad a \neq ab}{a \Leftarrow_I^{P1} ab}}{\vdash abb \in \mathfrak{f}1_{P1}(I)} \quad \dfrac{a \approx_I^{P1} ab \quad a \neq ab}{a \Leftarrow_I^{P1} ab}}{\vdash abbb \in \mathfrak{f}1_{P1}(I)} \quad \dfrac{ab \approx_I^{P1} a \quad ab \neq a}{ab \Leftarrow_I^{P1} a}}{\vdash abb \in \mathfrak{f}1_{P1}(I)} \tag{4.10}$$

This simple pre-theory, though very elementary, has a considerable complexity, which we will show for pedagogical reasons, so to speak.

Say a pre-theory $(\mathfrak{f}, P)$ is **undecidable**, if for some word $\vec{w}$ and some finite language I, the problem $\vec{w} \in \mathfrak{f}_P(I)$ is undecidable. If a pre-theory is undecidable, then its adequacy will be undecidable in general, so we would like our pre-theories to be decidable in any case. The following theorem might be surprising at the first glimpse, but is not surprising any more if we consider that in $\mathfrak{f}1$, our string substitutions are completely unrestricted:

Theorem 15 *The simple pre-theory $(\mathfrak{f}1, P1)$ is undecidable.*

We show this by reduction to the word problem for semi-groups, which is well-known to be undecidable (see for example Kleene's classical [34]). We would get an almost immediate proof by Thue systems, if we would not have the additional requirement of the substring relation $\sqsubseteq$ in order to allow for substitution, so there is some work to do.

A *semigroup* is a structure $(M, \cdot)$, where M is a set, $\cdot$ is an associative, binary operation on elements of M, and M is closed under $\cdot$. As $\cdot$ is associative, we omit brackets, and we write $m_1 m_2 ... m_i$ as shorthand for $m_1 \cdot m_2 \cdot ... \cdot m_i$. A semigroup is *free*, if for every $m \in M$ there is a unique term denoting it; that is, all equalities are trivial. Every free semigroup has a unique smallest set of generators; if M has a neutral element 1, if is denoted by $(X-1)^* - ((X-1)^*)^2$, otherwise it is just $X^* - (X^*)^2$. We denote generator set of M by $gen(M)$, so that we have $(gen(M))^+ = (M, \cdot)$, where M is the closure of the generators under the operation.

An unfree semigroup $(M, \cdot)$ has a presentation $((\Sigma^+, \cdot), E_S)$ by the free semigroup $(\Sigma^+, \cdot)$ over Σ and a set of equations E_S of the form $\vec{w} = \vec{v}$, for $\vec{w}, \vec{v} \in \Sigma^+$. We obtain $=_S$, the set of equalities holding on terms over M in (Σ^+, E_S), as the *smallest congruence* over Σ^+ containing all equations in E_S. So (Σ^+, E_S) presents $(M, \cdot)$, if $(M, \cdot) \cong [\Sigma^+]_{=_S}$, the free semigroup modulo the congruence. A semigroup is finitely presented, if both Σ and E_S are finite. Now the word problem for (finitely presented) semigroups is as follows:

Given a (finite) presentation (Σ^+, E_S), $\vec{w}, \vec{v} \in \Sigma^+$, does $\vec{w} =_S \vec{v}$ hold?

As we said, this problem is undecidable in general, and is also undecidable for finitely presented semigroups. We will now reduce the decision problem for $(\mathfrak{f}, P1)$ to this problem. So assume we have a finitely generated semigroup (Σ^+, E_S), and we want to decide whether the equation $\vec{w} = \vec{v}$ is valid in (Σ^+, E_S). We show that for every finite presentation (Σ^+, E_S), every $\vec{w}, \vec{v} \in \Sigma^+$, we can construct a finite language I such that $\vec{v} \in \mathfrak{f}_{P1}(I)$ if and only if $\vec{w} =_S \vec{v}$. The core of the proof consists in the construction of an appropriate language $I(\vec{w}, \vec{v})$.

So take an equation $\vec{w} =_S \vec{v}$, the validity of which we want to decide. Recall that $\vec{w}, \vec{v}$ in the sequel will always be used in this given sense. Before we construct the language $I(\vec{w}, \vec{v})$, we have to construct its alphabet. Assume $\vec{x} = \vec{y} \in E_S$, and $\vec{x} \not\sqsubseteq \vec{y}$, $\vec{y} \not\sqsubseteq \vec{x}$. We then take three letters a^y_x, b^y_x, c^y_x, which are unique for any $\vec{x}, \vec{y}$, and which are not in Σ. Now assume $\vec{x} = \vec{y} \in E_S$, and neither $\vec{x}$ nor $\vec{y}$ are substrings of $\vec{w}$. Then in addition take a letter d^x_y, which is also unique (regardless of whether $\vec{x} \sqsubseteq \vec{y}$ or not). Furthermore, for any string in $fact[\{\vec{x} : (\vec{x} = \vec{y}) \in E_S \text{ or } (\vec{y} = \vec{x}) \in E_S \text{ or } (\vec{x}) = \vec{w}\}]$, we take a unique letter e^x_x. Now we define $I(\vec{w}, \vec{v})$ as the smallest language, such that:

1. $\vec{w} \in I(\vec{w}, \vec{v})$;
2. if $\vec{x} \in fact[\{\vec{x} : \vec{x} = \vec{y} \in E_S \text{ or } \vec{y} = \vec{x} \in E_S \text{ or } \vec{x} = \vec{w}\}]$, then $e^x_x \vec{x} e^x_x \in I(\vec{w}, \vec{v})$.
3. if $\vec{w} = \vec{w}_1 \vec{x} \vec{w}_2$, $\vec{x} = \vec{y} \in E_S$ (or $\vec{y} = \vec{x} \in E_S$), $\vec{x} \sqsubseteq \vec{y}$ (or $\vec{y} \sqsubseteq \vec{x}$), then $\vec{w}_1 \vec{y} \vec{w}_2 \in I(\vec{w}, \vec{v})$;
4. if $\vec{w} = \vec{w}_1 \vec{x} \vec{w}_2$, $\vec{x} = \vec{y} \in E_S$ (or $\vec{y} = \vec{x} \in E_S$), $\vec{x} \not\sqsubseteq \vec{y}$ and $\vec{y} \not\sqsubseteq \vec{x}$, then $\vec{w}_1 a^y_x \vec{x} b^y_x \vec{y} c^y_x \vec{w}_2 \in I(\vec{w}, \vec{v})$, and $\vec{w}_1 \vec{y} \vec{w}_2 \in I(\vec{w}, \vec{v})$.
5. if $\vec{x} \notin fact(\vec{w})$, $\vec{x} = \vec{y} \in E_S$ (or $\vec{y} = \vec{x} \in E_S$), $\vec{x} \sqsubseteq \vec{y}$ (or $\vec{y} \sqsubseteq \vec{x}$), then $d^x_y \vec{x} d^x_y, d^x_y \vec{y} d^x_y \in I(\vec{w}, \vec{v})$;

6. if $\vec{x} \notin fact(\vec{w})$, $\vec{x} = \vec{y} \in E_S$ (or $\vec{y} = \vec{x} \in E_S$), and $\vec{x} \not\sqsubseteq \vec{y}$ and $\vec{y} \not\sqsubseteq \vec{x}$, then $d_y^x \vec{x} d_y^x, d_y^x \vec{y} d_y^x, d_y^x a_x^y \vec{x} b_x^y \vec{y} c_x^y d_y^x \in I(\vec{w}, \vec{v})$;

This defines the language $I(\vec{w}, \vec{v})$. We can easily check that $I(\vec{w}, \vec{v})$ is finite, because each condition only adds finitely many strings; in particular: each condition has the form of an implication, where the premise does not get changed by the other conditions being satisfied or not.

What is the first important point is $P1(I(\vec{w}, \vec{v}))$.

Lemma 16 *We have $(\vec{s}, \vec{t}) \in P1(I(\vec{w}, \vec{v}))$, if and only if either 1. $\vec{s} = \vec{t} \in E_S$ (or $\vec{t} = \vec{s} \in E_S$), and $\vec{s} \sqsubseteq \vec{t}$ (or vice versa), or 2. $\vec{t} = a_x^y \vec{s} b_x^y \vec{u} c_x^y$ (or inversely), where $\vec{s} = \vec{u} \in E_S$, or $\vec{t} = a_x^y \vec{u} b_x^y \vec{s} c_x^y$ where $\vec{s} = \vec{u} \in E_S$, and $\vec{s} \not\sqsubseteq \vec{u}, \vec{u} \not\sqsubseteq \vec{s}$.*

Proof. The *if* direction is clear by definition of $I(\vec{w}, \vec{v})$. We show the *only if*-direction. We have some $(\vec{s}, \vec{t}) \in P1(I(\vec{w}, \vec{v}))$ not satisfying the above conditions. By assumption, we must either have $\vec{s} \sqsubseteq \vec{t}$ or $\vec{t} \sqsubseteq \vec{s}$; assume wlog that $\vec{s} \sqsubseteq \vec{t}$. But by assumption, $\vec{t}$ does not have the form in 2., and by assumption, $\vec{s} = \vec{t}, \vec{t} = \vec{s} \notin E_S$. So there are strings $e_s^s \vec{s} e_s^s$, $e_t^t \vec{t} e_t^t$, in which each of the two have unique, distinct contexts, so we have $\vec{s} \not\leq_{I(\vec{w},\vec{v})} \vec{t}, \vec{t} \not\leq_{I(\vec{w},\vec{v})} \vec{s}$; contradiction. □

From this it easily follows that:

Lemma 17 *For $\vec{w}, \vec{v} \in M^*$, if $\vec{w} =_S \vec{v}$, then $\vec{v} \in \mathfrak{f}_{P1}(I(\vec{w}, \vec{v}))$*

Proof. Obvious, because each substitution corresponding to one equation in E_S can be simulated by at most two analogies. □

We now have to show the the other direction: for $\vec{v} \in \Sigma^*$, if $\vec{v} \in \mathfrak{f}_{P1}(I(\vec{w}, \vec{v}))$, then $\vec{w} =_S \vec{v}$. To see this, we first make sure: if $\vec{v} \in \mathfrak{f}_{P1}(I(\vec{w}, \vec{v}))$, then $\vec{v} \in \mathfrak{f}_{P1(I(\vec{w},\vec{v}))}(\vec{w})$, because all other strings in $I(\vec{w}, \vec{v})$ are either derivable as well from $\vec{w}$ by means of the analogies, or they do not allow to derive $\vec{v}$, because they contain the letters d_x^y, e_x^x, for which there is no way to get rid of by means of any analogy, and which by assumption do not occur in $\vec{v}$. So we can see that the statement we have to prove can without loss of generality be weakened to the statement: if $\vec{v} \in \mathfrak{f}_{P1(I)}(\vec{w})$, then $\vec{w} =_S \vec{v}$. So we prove:

Lemma 18 *For any $\vec{v} \in \Sigma^*$, if $\vec{v} \in \mathfrak{f}_{P1(I(\vec{w},\vec{v}))}(\vec{w})$, then $\vec{v} =_s \vec{w}$.*

Proof. This is clear for all analogies $(\vec{s}, \vec{t})$, where $\vec{s} = \vec{t} \in E_S$, and moreover $\vec{s} \sqsubseteq \vec{t}$ (or vice versa). For all analogies of the other kind, which introduce symbols not in Σ, the argument is the following: each of these analogies introduces a substring $a_x^y \vec{x} b_x^y \vec{y} c_x^y$. As a_x^y, b_x^y, c_x^y are unique, we can only get rid of them by the two analogies which introduce them. This means in particular, we can either substitute it by $\vec{x}$ or by $\vec{y}$, where $\vec{x} = \vec{y} \in E_S$; there is no other to get it out from a string.

So we have to use two analogies, one to introduce it, one to get rid of it, and they exactly correspond to one equation in E_S. □

This completes the proof of the above theorem: we have $\vec{w} =_S \vec{v}$ iff and only if, for $\vec{v} \in \Sigma^*$, $\vec{v} \in \mathfrak{f}1_{P1}(I(\vec{w}, \vec{v}))$. So if the latter were decidable, so would be the former – contradiction.

So we see that already for this very simple pre-theory, which uses nothing but substitution in contexts, we get an undecidability result. This means that the adequacy of $(\mathfrak{f}, P1)$ is undecidable. The main reason for this negative result is

obvious: there is no notion of structure in the pre-theory, substrings which result from substitutions are neither marked nor recognized as such. This is not only the main reason for undecidability, but also goes strongly against our intuitions on the nature of language. The other main reason for undecidability is that analogies do not make strings only longer, but also shorter, as the symmetry of $P1$-similarity is transferred to analogies. We will therefore now introduce new inference rules doing away with both problems, and to which we will refer as **structural inference**.

4.4 Structural Inference

We now introduce a new set of inference rules $\mathfrak{g}$, which we will refer to as "structural inference". It uses special markers $(,)$, which are supposed not to be in the alphabet of any of the (finite) languages we consider as the range of pre-theories. These mark the beginning and the end of any of the substrings which has been altered by an inference, such that our inference rules – already written in tree-form – now look as follows:

$$\frac{\vdash \vec{w}\vec{b}\vec{v} \in I \quad \vec{b} \Leftarrow_L^P \vec{a}}{\vdash \vec{w}(\vec{a})\vec{v} \in I}, \qquad \frac{\vec{x} \approx_I^P \vec{y} \quad \vec{x} \neq \vec{y} \quad \vec{x} \sqsubseteq \vec{y}}{\vec{x} \Leftarrow_I^P \vec{y}} \tag{4.11}$$

Henceforth, we use $\vec{x} \sqsubset \vec{y}$ as an abbreviation for $\vec{x} \sqsubseteq \vec{y}$ and $\vec{x} \neq \vec{y}$. Again, these schemes represent "metarules", where we use strings as variables for arbitrary strings and P as a variable for an arbitrary analogical map etc. Note that the first rule properly introduces the bracket: it is not given or specified among the premises! For $I \subseteq \Sigma^*$, we now have $\vec{w}, \vec{v} \in \Sigma^* \cup \{(,)\}$. The important thing is: we do not allow that (or choose the two such that) the two distinguished symbols $(,)$ figure in the language we observe; and consequently, they will not figure in any of the analogies we get, if we define analogies in the usual and natural way. We thus restrict the inferences to substrings which are either already present in the original premise, or which have been introduced entirely in a single inference.

The introduction of structure into inferences also leads to a distinction of the strong language, which contains the distinguished symbols $(,) \notin \Sigma$, and is used to make inferences; and the weak language, which is the result of inferences and is obtained from the weak language through a homomorphism h mapping $(,)$ onto ϵ and computing the identity for anything else.

This is a significant modification of our ontology. Till now, there were only strings and nothing else. We now assume that the languages our pre-theories provide have *structure*. It now seems to be a matter of taste whether we want to think of "language" as the set of derivable strings without any structure or as structured entities; formally, whether "language" is supposed to be $h \circ \mathfrak{g}_P(I)$ or $\mathfrak{g}_P(I)$ for some P and I. The latter is more natural and immediate, but we might also want to think of "language" as a set of structured objects, for example a set of trees, as many people do. The latter perspective is usually strongly put forward by generative linguists, who believe in the cognitive reality of the structures they posit. We obviously disagree with this position for epistemic reasons. But apart from this, in our case, things are slightly different, because we only posit the structure we introduce by inference, which concerns strings we do not observe in the first place, whereas (not only) generative linguists assume that

languages are structured "all the way down", that is, we have a tree structure where we allow only visible words in the sense of atoms as leaves, not entire strings. So linguists would probably not be happy with our results anyway.

The most simple and convincing argument for saying that structures are not part of language is that we do not observe them, but only posit them for the sake of our theories. A variant of this position is put forward for example in [16]. But note that this very convincing argument does not obtain in our case: our pre-theoretic inferences are about deriving strings we do not observe *anyway*; so there is nothing really empirical we can say about these objects and their nature, in principle it could be trees or anything we consider an adequate model of natural language utterances. So again we must not confuse what we construct with what we observe: there is a neat distinction between the level of linguistics and metalinguistics. It might sound like a convincing argument that we observe strings, and so we should also infer strings; conversely, one might reply that the inferred objects have a different status, so why should they not be somewhat different in nature? Given this, it seems basically a matter of taste and choice what to assume as "language".[1]

Also another remark is in order. If we use structured inference, there might be inferences, which do not allow to derive additional strings, but allow to derive *additional structures*. In particular, provided we have a simple string $\vec{w}$ in our positive language from which we depart, we might be able to derive it via inferences in a certain *structured fashion*. This is not very important for our principal motivation here, but might be considered of some relevance if we think that our utterances should be structured "all the way down".

An easy example is the following: recall the simple analogical map $P1$, but now with structured inference, yielding the pre-theory $(\mathfrak{g}, P1)$. Take a language $I := \{ab, aabb, aaabbb\}$. Now we have $ab \approx^I_{P1} aabb$. So we can derive the strings $a(ab)b$, $a(a(ab)b)b$ etc. These inferences are obviously uninteresting for linguistic metatheory in its primary sense, but they might be, up to a certain point, a formal counterpart for what the linguist is doing when inferring the structure of utterances he empirically knows to be part of the language in question. Note however that I do not claim that any of the pre-theories I look at can actually provide a complete model of this procedure. Also, none of them provides a guarantee that all strings of a finite language can be structured "all the way down".

Our first analogical map was $P1$; we will now define a second one, which is slightly more rigid in its premises. These two maps are very fundamental for this work, because most analogical maps we consider are basically the one or the other in some disguise or variation. The map we introduce now is the map Pr of simple *pseudo-recursion*. Its crucial concept is the one of pseudo-recursion, which is defined as follows:

Definition 19 *In a (finite) language $I \subseteq \Sigma^*$, $\vec{w}, \vec{v} \in \Sigma^*$, $\vec{w} \neq \vec{v}$, $\vec{w}$ and $\vec{v}$ are* ***pseudo-recursive****, if for some $\vec{x}, \vec{y}$,*

1. *$\vec{x}\vec{v}\vec{y} = \vec{w}$ (in other words: $\vec{v} \sqsubseteq \vec{w}$);*

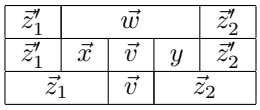
[1] As a short remark, I might add at this point that I find the idea that language is structured, but not all "the way down", very appealing. This idea is also not totally out of linguistic mainstream, as it seems to guide some work on chunking and construction grammar, as well as work on data-oriented parsing.

2. $\vec{v} \leq_I \vec{w}$, *and*

3. *if* $(\vec{z}_1, \vec{z}_2)$ *does not have the form* $(\vec{z}_1'\vec{x}, \vec{y}\vec{z}_2')$, *then* $\vec{z}_1\vec{v}\vec{z}_2 \in I \Rightarrow \vec{z}_1\vec{w}\vec{z}_2 \in I$ *(or vice versa for all three).*

The notion of pseudo-recursion needs some explanation, in particular, the meaning of the third condition. It is clear that in any finite language I with substrings $\vec{w}, \vec{v}$, $\vec{w} \neq \vec{v}$ and $\vec{w} \sqsubseteq \vec{v}$, $\vec{w}$ and $\vec{v}$ cannot have the same distribution, that is, we cannot have $\vec{w} \sim_I \vec{v}$ (we showed this in the previous section). The reason is that either we will find a distinguishing occurrence, or the language is necessarily infinite. The essence of pseudo-recursion of two strings $\vec{w}, \vec{v}$, $\vec{v} \sqsubseteq \vec{w}$, is that they are as similar in distribution as they *can* be in a finite language.[2]

To illustrate the concept of pseudorecursion, consider the following illustration:



You see that every occurrence of $\vec{w}$ introduces an occurrence of the $\vec{v}$, with a distinguishing context. We call this context a *recursive context*. So $(\vec{x}, \vec{y})$ is a recursive context for $\vec{w}, \vec{v}$, if $\vec{w} = x\vec{v}y$. Note that the third condition might be thought to be unnecessarily complicated. This is not true, as it also covers the case $I = \{b, bb\}$. In that case, we have two occurrences of b in bb, and two contexts (ϵ, b) and (b, ϵ). Still, have the pseudo-recursion $(b, bb) \in Pr(I)$! So there are critical cases where the recursive context in definition 18 is not unique, and here we implicitly quantify over all (two) possible contexts. So pseudo-recursion might be thought of as saying: except for the recursive contexts which necessarily distinguish them, there are no other distinguishing contexts for $\vec{w}, \vec{v}$.

We give a small example. Say a language is pseudo-recursive if some non-trivial substrings of it are. Then we have the following example languages:

1. $I_1 := \{a, ab\}$ is pseudo-recursive;

2. $I_2 := \{a, ab, ac\}$ is not;

3. $I_3 := \{a, ab, ac, abc\}$ is pseudo-recursive.

In I_1, a and ab are distinguished only by the context (ϵ, b), which is a recursive context for the two. In I_2, we have the additional context (ϵ, c), which is not recursive for a, ab, but distinguishes the two substrings. In I_3, in turn, (ϵ, c) is no longer a distinguishing context for a, ab, and so the two are again pseudo-recursive, because the only context distinguishing them, which is (ϵ, b), is recursive for a, ab.

We get the analogical map Pr by putting $\vec{w} \approx_L^{Pr} \vec{v}$ if and only if $\vec{w}, \vec{v}$ are pseudo-recursive in L. We now use the rules in $\mathfrak{g}$, and so analogies will be

[2] Note that there is an interesting alternative definition of the third condition:

3. if $(\vec{z}_1, \vec{z}_2)$ does not have the form $(\vec{z}_1'\vec{x}', \vec{y}'\vec{z}_2')$, such that $\vec{x}'\vec{v}\vec{y}' = \vec{x}\vec{v}\vec{y}$, then $\vec{z}_1\vec{v}\vec{z}_2 \in I \Rightarrow \vec{z}_1\vec{w}\vec{z}_2 \in I$ (or vice versa for all three).

This definition gives rise to different analogies and is quite interesting; we will however not investigate it at this point, and only mention it.

asymmetric, as we made it a precondition of analogies that we have a proper substring relation. So finish with the scheme:

$$\frac{\vec{w} \approx_I^{Pr} v \quad \vec{w} \sqsubseteq v \quad \vec{w} \neq \vec{v}}{\vec{w} \Leftarrow_I^{Pr} \vec{v}} . \tag{4.12}$$

$h \circ \mathfrak{g}_{Pr}$ is a reasonable projection. One might now think that for $\vec{w} \approx_I^{Pr} \vec{v}$, we have $\vec{w} \sim_{h(\mathfrak{g}_{Pr}(I))} \vec{v}$. This however is not true: we do have $\vec{w} \sim_{\mathfrak{f}_{Pr}(I)} \vec{v}$, but in the structural inference, we do *not* give $\vec{w}, \vec{v}$ an equal distribution, exactly because of the brackets we introduce! So if we have $\vec{w} \approx_I^{Pr} \vec{v}$, it is not clear at all whether $\vec{w} \sim_{h(\mathfrak{g}_{Pr}(I))} \vec{v}$, though it might be the case. However, it is obvious that structural inference makes it much easier to obtain some positive results on complexity and expressive power:

Theorem 20 *For any finite language I, $h(\mathfrak{g}_{Pr}(I))$ and $h(\mathfrak{g}_{P1}(I))$ is a context-free language.*

Proof. We give a construction equally valid for both cases, but we only illustrate it for $P1$. Take a fixed finite language I; the terminals of our grammar are the alphabet of I. For any $\vec{w} \in \mathit{fact}[I]$ (!), we introduce a non-terminal $N_{\vec{w}}$. Now, for every $\vec{v} \in I$, we introduce a rule $S \to N_{\vec{w}}$; for all $\vec{u}, \vec{v}, \vec{w} \in \mathit{fact}[I]$, such that $\vec{u}\vec{v} = \vec{w}$, introduce a rule $N_{\vec{w}} \to N_{\vec{u}} N_{\vec{v}}$, and finally, for all $\vec{w} \in \mathit{fact}[I]$, we add a rule $N_{\vec{w}} \to \vec{w}$. The resulting grammar G' is finite because I is finite, and it generates exactly I, though in all "possible ways", that is, with all possible, binary trees.

Now, for every analogy $(\vec{x}, \vec{y}) \in P1(I)$, we simply add a rule $N_{\vec{x}} \to N_{\vec{y}}$. This does the job as required, because we can decompose every $N_{\vec{y}}$ into its "substring nonterminals"; but we cannot go the other way round, in the same way as we can introduce brackets, but not get rid of them in derivations. □

4.5 Properties of Pre-Theories I

4.5.1 Problems for Infinite Languages

We show here that for infinite languages, things get undecidable very quickly. This section is not meant to present any important positive results, but rather supposed to show negative results, and in particular, motivate our mistrust in infinite languages. Mistrust is to be understood in the sense: we do not want to work with infinite languages. This has, of course, a linguistic/philosophical motivation, which we discussed at length in the first section. But this is not sufficient to justify our mistrust: we could, for example, investigate **fixed points**: We could map a finite language I onto an infinite language $\mathfrak{f}_P(I)$. We could then argue: as linguists we are realists, so for us, $\mathfrak{f}_P(I)$ is in some sense real. Now it might be that some patterns, which we have not observed in I, become visible only in $\mathfrak{f}_P(I)$, and these are nonetheless relevant to "language". This is a valid argument, as we might argue that I is insufficient in a strong sense, it does not even show all the patterns we need; these patterns might become visible only in $\mathfrak{f}_P(I)$. So rather than define "language" as $\mathfrak{f}_P(I)$, we could define it as $(\mathfrak{f}_P)^*(I)$, the least fixed point of the iterated mapping.

This position, though linguistically/philosophically reasonable, from my position is unsustainable on purely mathematical reasons, because the mappings

and their values very quickly become undecidable. This is what we will argue for in this intermezzo. As regards the conclusion of this section, we should have put it earlier in the order of contents. However as regards the methods applied, we think at this point it will be much easier to understand by the reader. Our first result is the following:

Theorem 21 *Let G be a CFG, such that $L(G) = L \subseteq \Sigma^*$. Then the relation $\leq_L \subseteq \Sigma^* \times \Sigma^*$ is in general undecidable.*

Proof. As is well-known, the universality problem for CFLs is undecidable, that is, there is no algorithm which for any CFG G, tells us whether $L(G) = \Sigma^*$ or not. As is also well-known, the emptiness problem for CFGs *is* decidable, that is, there is an algorithm which tells us whether $L(G) = \emptyset$ for any CFG G.

Now assume $\leq_{L(G)}$ is decidable. We show that under this assumption the universality problem is decidable, yielding a contradiction. To check universality, we first check whether $L(G) = \emptyset$, which is decidable. If it is, we can answer the universality question negatively. So assume $L(G) \neq \emptyset$. Then there is a word $\vec{w} \in L(G)$. We check whether $\epsilon \in L(G)$. Next we check whether $\epsilon \leq_{L(G)} a$ for all $a \in \Sigma$. If both are answered positively, then we have $L(G) = \Sigma^*$; if one is negative, then $L(G) \neq \Sigma^*$. This way, we can effectively decide whether $L(G) = \Sigma^*$ for any CFG G. This is a contradiction, as this is an undecidable problem. So $\leq_{L(G)}$ is undecidable. □

A similar result is the following:

Theorem 22 *Let G be a CFG, $L(G) \subseteq \Sigma^*$. Furthermore, let $W \subseteq \Sigma^*$ be a set of strings. Then it is undecidable whether $W \in [\Sigma^*]_{\sim_{L(G)}}$, that is, whether W is an equivalence class of strings. Furthermore, it is in general undecidable whether $\vec{w} \sim_{L(G)} \vec{v}$ for some $\vec{w}, \vec{v} \in \Sigma^*$.*

Proof. The proof is very similar: assume that $\sim_{L(G)}$ is decidable. Then from the decidability of the emptiness problem we can easily deduce that universality is decidable, yielding a contradiction, because $|[\Sigma^*]_{\sim_{L(G)}}| = 1$ if and only if $L(G) = \Sigma^*$ or $L(G) = \emptyset$, and the latter is decidable. □

So there is little we can say about interesting classes of infinite languages, and we have to decide on things in the finite. This last result also shows why equivalence classes are really of little use for us: for the infinite, they are undecidable, and in the finite, they do not contain interesting patters, like strings in a substring relation (this will change however in the sequel).

There are some further things to consider: for example, the relations $\leq_L$ and consequently $\sim_L$ are *compact* in the following sense:

Lemma 23 *Given an infinite language $L \subseteq \Sigma^*$, $\vec{w}, \vec{v} \in \Sigma^*$, we have $\vec{w} \leq_L \vec{v}$ ($\vec{w} \sim_L \vec{v}$) if and only if for every finite fragment $I \subseteq L$ there is a finite $J : I \subseteq J \subseteq L$ such that $\vec{w} \leq_J \vec{v}$ ($\vec{w} \sim_J \vec{v}$).*

We have shown an equivalent result above. So the properties in the infinite are determined by the finite fragments, even though, of course, there are infinitely many. However, there are other properties of $\leq_L$ which are peculiar to the infinite. For example, take the property of **well-foundedness**. Well-foundedness means: for $\leq \subseteq M \times M$, for each $m \in M$, the set $\{n : n \leq m\}$ is finite. Now obviously, for any finite language L, $\leq_L$ is well-founded. This does however not hold for infinite languages:

Lemma 24 *There exists context-free languages L, such that $\leq_L$ is not well-founded.*

Proof. Take $L := \{a^m b^n : m \geq n\}$. In this case, we have $a^m \leq_L a^n$ iff $m \geq n$. This is obviously not well-founded. □

4.5.2 On Regular Projection

There is a good argument in favor of weaker forms of projection. For example, one might be one of the advocates of the regularity of natural language. A more reasonable position is the following: we want to allow inferences only if they "preserve acceptability"; that is: we want to make sure, that if we infer a new judgment, then it should be as acceptable as its premise. Whereas this sounds simple from a linguistic point of view, from a formal point of view it is obviously problematic: in the relevant case, we can derive infinitely many new judgments – we cannot simply check whether they preserve acceptability. So in translating this requirement into a formal theory, one has to do some work, and we will investigate two approaches.

The first approach is the following: one might say: the set of observable (acceptable) utterances is regular, so we have to take care that projected languages are regular; everything else will be fine by that point. Therefore, we allow pre-theories of the above type, but we restrict analogies to the scheme:

$$\vec{x} \Leftarrow \vec{x}\vec{x}_1 \tag{4.13}$$

that is, we make analogies roughly correspond to regular rules. Obviously, this is a particular instance of our above scheme.

We can accordingly define the analogical maps $RPr, RP1$ by $(\vec{x}, \vec{y}) \in RPr(I)$ (and $\vec{x}, \vec{y}) \in RP1$) if and only if

1. $\vec{x} \approx_I^{Pr} \vec{y}$ (and $\vec{x} \approx_I^{Pr} \vec{y}$), and
2. $\vec{x} = \vec{y}\vec{y}_1$ or $\vec{y} = \vec{x}\vec{x}_1$.

So what we do is: in addition to the $P1$- or Pr-requirements on contexts etc., we restrict the form analogies can have. What are formal properties that come with this scheme? One might conjecture that, given that the scheme does not allow for "center embedding", for any finite language I, $\mathfrak{g}_{RP1}(I)$ is a regular language. However, this is not true: just assume we have an analogy of the form $a \Leftarrow aab$, and a premise ab. Then we can make derivations of the form

$$\dfrac{\dfrac{\vdash ab \in \mathfrak{f}_P(I) \quad a \Leftarrow aab}{\vdash (aab)b \in \mathfrak{f}_P(I)} \qquad a \Leftarrow aab}{\vdash (a(aab)b))b \in \mathfrak{g}_P(I)} \tag{4.14}$$

Of course, we can also derive strings which are not of this form; but for $I = \{ab\}$, $P(I) = (a, aab)$, we obtain $\mathfrak{g}_P(I) \cap a^*b^* = \{a^n b^n : n \in \mathbb{N}\}$. From this we can conclude that we derive a non-regular language. So the restricted analogy scheme alone does *not* prevent us from deriving non-regular languages! But that does not show whether $(\mathfrak{g}, RP1), (\mathfrak{g}, RPr)$ actually do derive non-regular languages.

Lemma 25 *There are finite languages I, such that $\mathfrak{g}_{RP1}(I)$ is not regular.*

Proof. Just take the language $I = \{ab, aabb\}$. That gives exactly the above example. □

For RPr, things are more complicated: the example $I = \{ab, aabb\}$ does not work, as $RPr(I) = \emptyset$. In general, in RPr we cannot have any analogies of the form $(\vec{x}, \vec{x}\vec{y}\vec{x}\vec{z})$: because in this case, assume we have $\vec{w}\vec{x}\vec{v} \in I$. Then by the Pr conditions, we need $h(\vec{w}(\vec{x}\vec{y}\vec{x}\vec{z})\vec{v}) \in I$ (we write the brackets to make our reasoning more comprehensible; h takes them out again so that we get a simple string). But then we also need $h(\vec{w}\vec{x}\vec{y}(\vec{x}\vec{y}\vec{x}\vec{z})\vec{z}\vec{v}) \in I$ etc., such that I needs to be infinite. Actually, this argument gives us a stronger result, namely that if $(\vec{x}, \vec{x}\vec{y}) \in RPr(I)$, then there can be no analogy $(\vec{z}, \vec{z}\vec{u}) \in RPr(I)$ such that $\vec{z} \sqsubseteq \vec{x}\vec{y}$ or $\vec{x} \sqsubseteq \vec{z}\vec{u}$, unless $\vec{x} = \vec{z}$. Given this, we can conclude:

Lemma 26 *For any finite language I, $\mathfrak{g}_{RPr}(I)$ is regular.*

Proof. We show this by constructing a (non-deterministic) finite state automaton: take a language I, and construct an FSA as follows: 1. for any distinct input, it goes into a distinct state; the state is accepting, if the path is labelled by $\vec{w} \in I$; and all transitions not being reachable by a prefix of a word in I are undefined. This automaton recognizes I, and each state is uniquely characterized by a single input word; we therefore write $q_{\vec{w}}$ with the obvious meaning. 2. Now for every $(\vec{x}, \vec{x}\vec{y}) \in RPr(I)$, we add a (distinct, non-deterministic transition from all $q_{\vec{u}\vec{x}}$ to itself, which is labelled by $\vec{y}$.

The important thing is that we cannot read a new $\vec{x}'$ being on the left-hand side of an analogy, before we go back into state we have been after reading $\vec{u}\vec{x}$. Therefore, this automaton recognizes $\mathfrak{g}_{RPr}(I)$, and obviously, it is finite. □

So we see that in this case, the induced languages are regular. That is what we desired; however, it does not directly follow from the restriction of the substring condition, but from properties of Pr. It is doubtful whether this argument is really related to the fact that inferences preserve acceptability. Of course, this has some advantages: as the projected languages are regular, they have much easier decision problems; in particular, the above undecidability results do not hold. This is however only a minor comfort, if we consider that it runs counter to many intuitive arguments, we would *a priori* restrict our view to regular languages – and we do not even know whether this conforms at all with our intuitions on understanding!

We now come to the second approach to "regularity". This approach is more clever in the following sense: whereas the first approach assumes *a priori* that "language" is regular, just by restricting possible substitutions, the second approach just assumes stricter criteria for similarity: these do not *necessarily* result in regular languages, but are chosen in a way that under the presumed shape of datasets we observe, we only obtain regular languages. This presumed shape of the restrictions is the following: we have observed that certain patterns are observable only until a certain bounded depth, whereas for others we do not have this restriction. We again restrict ourselves to substitution of substrings. We write $\vec{x} \approx_I^{Pr-k} \vec{x}_1\vec{x}\vec{x}_2$, where $k \in \mathbb{N}$, if the following hold:

1. $\vec{x} \approx_I^{Pr} \vec{x}_1\vec{x}\vec{x}_2$
2. if $(\vec{w}, \vec{v})$ is not recursive for $(\vec{x}, \vec{x}_1\vec{x}\vec{x}_2)$, $\vec{w}\vec{x}\vec{v} \in I$, then for every $i \leq k$, $\vec{w}(\vec{x}_1)^k\vec{x}(\vec{x}_2)^k\vec{v} \in I$.

The pre-theory Pr-k thus requires that the substitution already has k instances in I. The underlying reasoning is the following: if we can do it k-times (rather than once), we can do it arbitrarily often. For $P1$, there does not seem to be a reasonable analogue. It is intuitively clear that in natural languages, by choosing k large enough, we can exclude any analogy $(\vec{x}, \vec{x}_1\vec{x}\vec{x}_2)$ where both $\vec{x}_1 \neq \epsilon \neq \vec{x}_2$. Note that this argument, though obvious from a "performance oriented" view, is very subtle from a metalinguistic point of view: even if there were no experimentally measurable limits on center embedding etc., we could still choose a k which would exclude these analogies for any finite dataset I. We have no formal constraint to first fix the pre-theory, and then begin to gather the data. However, an explicit part of our procedure is that after fixing the pre-theory, we are still allowed to gather as much data as we want, before we do the projection (though we are not allowed to discard data!). So we can choose a reasonably small k, and performance constraints will make sure this actually does the job we want it to do.

In principle, there is nothing we can say against this approach to "language"; it can also easily be adapted to other pre-theories we have presented so far and which we will present in the sequel. Note that the difference of Pr and Pr-k is not so much of mathematical relevance, as of linguistic relevance: it is the particular nature of the language we observe which makes a huge difference between the two. As we focus on mathematical properties, we will not ponder very much about this restriction. We mention however that Pr-k gives us a very good example of what a property of "language" modulo a pre-theory means, a notion we will scrutinize later on: the fact that with sufficiently large k, "natural languages" are regular under $(\mathfrak{g}, Pr\text{-}k)$ is (might be) an empirical property of natural language. Consequently, adopting $(\mathfrak{g}, Pr\text{-}k)$, natural languages being regular is still not an empirical property, but neither is it a truism on methodological grounds (we will call this a *methodological universal*), as it could be otherwise! So it is something in between, and we thus say that "languages" are regular modulo $(\mathfrak{g}, Pr\text{-}k)$; whereas they might not be modulo $(\mathfrak{g}, Pr)$! So we can speak of properties modulo pre-theories.

There is however also a mathematical difference between Pr and Pr-k. Put, for example, $k = 4$ and $I = \{ab, aabb, aaa, caaac\}$. We get $\mathfrak{g}_{Pr}(I) = \{a^nb^n : n \in \mathbb{N}\} \cup \{c^naaac^n : n \in \mathbb{N}\}$. For Pr-4 we get $\mathfrak{g}_{Pr-4}(I) = I$. This is clear; what is less clear that there is not even an extension J, such that $J \supseteq I$, and $\mathfrak{g}_{Pr}(I) = \mathfrak{g}_{Pr-4}(J)$! To get $\{a^nb^n : n \in \mathbb{N}\}$, we need $\{ab, aabb, aaabbb, aaaabbbb, aaaaabbbbb\}$. But if we have these strings in a language, there is no way of getting an analogy $(aaa, caaac)$ anymore! (Note how these examples relate to upward normality, a notion we consider later on). So Pr-k is not only smaller in the sense of the results of projections of a given finite language, but also in the sense that the class of languages it can induce from any finite language seems to be quite restricted, and it seems to me that it is not restricted in a very favorable way (we will prove this claim later on).

4.5.3 On Similarity

We will have a short look on some properties of the similarity relations we have introduced so far. $\approx_I^{P1}, \approx_I^{Pr}$ are symmetric by definition. As such, they are both intransitive. To see this for $P1$, just but $I = \{a, ab, b\}$, where $P1(I) = \{(a, ab), (ab, a), (b, ab), (ab, b)\}$. This is intransitive, because $(a, b) \notin P1(I)$. The

same holds for Pr, and can be seen with the same example. Is this a bad thing? There seems to be dispute on whether similarity as such should be transitive or not; there are good arguments in favor and disfavor. There seems to be a strict conceptions of similarity, which is intransitive, and a broad conception, which allows for transitivity and thereby is more liberal. We do not want to go into this, but just point out the following: in its asymmetric reading, $P1$ is in fact transitive, that is:

Lemma 27 *If* $\vec{w} \approx_I^{P1} \vec{v} \approx_I^{P1} \vec{u}$ *and* $\vec{w} \sqsubseteq \vec{v} \sqsubseteq \vec{u}$, *then* $\vec{w} \approx^{P1} \vec{u}$.

The proof is simple: as both $\leq_I, \sqsubseteq$ are transitive, we know that $\vec{w} \leq_I \vec{u}, \vec{w} \sqsubseteq \vec{u}$. So for $P1$, if we skip the symmetry, we get transitivity. For Pr, this does not obtain, and in fact, the question for transitivity is meaningless:

Lemma 28 *There is no finite language* I, *with distinct* $\vec{w}, \vec{v}, \vec{u} \in fact(I)$, *such that* $\vec{w} \approx_I^{Pr} \vec{v} \approx_I^{Pr} \vec{u}$ *and* $\vec{w} \sqsubseteq \vec{v} \sqsubseteq \vec{u}$.

Proof. Assume we have $\vec{w} = \vec{x}, \vec{v} = \vec{y}_1\vec{x}\vec{y}_2, \vec{u} = \vec{z}_1\vec{y}_1\vec{x}\vec{y}_2\vec{z}_2$. Assume we have $\vec{a}\vec{x}\vec{b} \in I$, where $(\vec{a}, \vec{b})$ is the shortest (in terms of string-length) context of $\vec{x}$ in I. This entails that it is non-recursive for the analogies in question. If $\vec{w} \approx_I^{P1} \vec{v}$, then $\vec{a}\vec{y}_1\vec{x}\vec{y}_2\vec{b} \in I$. As $\vec{v} \approx_I^{P1} \vec{u}$, we also have $\vec{a}\vec{z}_1\vec{y}_1\vec{x}\vec{y}_2\vec{z}_2\vec{b} \in I$. But then, we also need $\vec{a}\vec{z}_1\vec{x}\vec{z}_2\vec{b} \in I$ (downward Pr); and so we need $\vec{a}\vec{z}_1\vec{z}_1\vec{y}_1\vec{x}\vec{y}_2\vec{z}_2\vec{z}_2\vec{b} \in I$ (upward Pr); then we need $\vec{a}\vec{z}_1\vec{z}_1\vec{x}\vec{z}_2\vec{z}_2\vec{b} \in I$ etc., such that I is necessarily infinite. □

So we might think of $P1$ as representing the liberal, transitive notion of similarity, Pr representing the restrictive, intransitive one.

4.6 Properties of Pre-Theories II

4.6.1 Characteristic and Downward Normal Pre-Theories

Now we will scrutinize properties of string-based substitutional pre-theories with structural inference. Our main question is now: what makes us prefer some pre-theory over another? It is important to keep in mind that this is only to a very small extent an empirical question. Adequacy with respect to partial languages, as we have sketched it before, is for us not a very effective criterion in any direction in the first place, because we are not going into "empirical details" here (we do not actually construct realistic partial languages, as this is the linguists task). So our only "hard" criterion of adequacy is the infinity of some images. So there is some empirical aspect to evaluation, the partial languages we observe/construct can show some pre-theories inadequate, but firstly this will not be very rich, and secondly, this is a linguistic issue (as it depends on actual linguistic observations) rather than a metalinguistic one. From a metalinguistic point of view, there is at this point nothing we can say about the "quality" of pre-theories. We will now consider general properties of pre-theories, what we can call *a priori* properties. As we will see, these will lead to *a priori* properties of the "languages" we can obtain, and properties of functions from finite to infinite languages. The first important property we will consider is **characteristicity**.

In this first discussion, we will introduce a theme which is of some importance for this general work. Many problems of linguistics present themselves in a

similar form in metalinguistics. However, whereas in the linguistic view, there is little we can do about it, in metalinguistics we can make it much more amenable by choosing, in the classical paradigm, an appropriate pre-theory. We will first discuss this with a variant of the problem of "private languages".

A problematic fact for linguistic theory is the following: there is a considerable incongruence in linguistic judgments, not only between different speakers, but also regarding the same speaker and different contexts/times. Whereas it might be plausible to attribute the former effect to the fact that different speakers know different languages, this does not sound plausible regarding the same speaker over different times, which in the case of some priming effects might even be very short.

This question of "private languages" indeed poses fundamental problems to linguistics (see [42]). Whereas we cannot deal with these here, there is certainly a similar problem for the metalinguist: different linguists will surely not observe exactly the same fragment of a language; even worse, one linguist will find the same utterance one time to be acceptable, whereas another time it will not be judged unacceptable. This is not as bad as one would think: if it is not judged to be acceptable, it does not follow that it is in the negative language. Still, this is problematic, and it should somehow be possible to yield an agreement on the projected language despite differences in the observations. Note that maybe a linguist, as concerned with the cognitive reality of speakers, might think differently. But keep in mind that we are doing *metalinguistics* here, and we have the goal of constructing the proper subject of linguistics, on which linguists should agree! In a word, we would have a better situation if we can have agreement on "language" despite some disagreement on the observed language. So it would be very favorable to have a property of pre-theories which up to a certain point can ensure this. This is done by the concept of characteristic pre-theories:

Definition 29 *We say a pre-theory* $(\mathfrak{f}, P)$ *is* ***characteristic****, if for every language* L *and for all languages* $I_1, ..., I_n$ *such that* $\mathfrak{f}_P(I_i) = L : 1 \leq i \leq n$*, there is a unique smallest language* J *such that* $f(J) = L$ *and* $J \subseteq \bigcap_{1 \leq i \leq n} I_i$*.*

Note that this covers the special cases where there is no finite language I such that $\mathfrak{f}^*_P(I) = L$; because then the only language inducing L is L itself. We can say that this unique smallest language is characteristic of L, and if a pre-theory is characteristic, then for every language L it induces there is a smallest characteristic language. As we said, this is motivated by the question of critical data: with characteristic pre-theories we know which part of an observed language I is essential for the language it generates under P, and which not.

A pre-theory $(\mathfrak{f}, P)$ is **injective**, if for any I, J, if $I \neq J$, then $\mathfrak{f}_P(I) \neq \mathfrak{f}_P(J)$. Obviously, any pre-theory $(\mathfrak{f}, P)$ such that $\mathfrak{f}_P$ is *injective* is trivially characteristic, and in this case, the concept of characteristicity is entirely meaningless. We will however not consider such pre-theories, because they would violate a fundamental principle of linguistics: as linguists, and equally as speakers, we are exposed to very different data (observed languages), yet still we agree broadly on what "language" is. Obviously, we do not need to, but if we *cannot* agree, then there is really no hope. We will later consider some other important properties of pre-theories, which exclude the injective pre-theories categorically; such that this simplistic solution is completely out of question.

For the pre-theories we have seen so far, we have seen in some examples that different finite languages give rise to the same infinite language under some pre-theory, and this is the case where things get interesting for us.

We have said that the disagreement on whether certain strings belong to a language or do not is a big problem for us. In how far do characteristic pre-theories address this? What we want is a kind of downward monotonicity: we want to be sure that by taking certain strings away from our database, we still obtain the same language. It is however unclear how we can make this desideratum precise: a general version of downward monotonicity of the form: if $\mathfrak{f}_P(I) = L$, $J \subseteq I$, then $\mathfrak{f}_P(J) = L$ is out of the question, because not only it is much too strong to have any linguistic plausibility, but it would also have a devastating effect from a purely mathematical point of view: it would trivialize the entire procedure of projection (contrary to a similar, upward version of monotonicity, which we will discuss later on). Being characteristic is in some sense a weak version of downward monotonicity, but much weaker than the general downward monotonicity, and it seems reasonable to require it. Characteristicity requires that if we have $\mathfrak{f}_P(I_1) = \mathfrak{f}_P(I_2)$, then there is a $J \subseteq I_1 \cap I_2$ such that $\mathfrak{f}_P(I_1) = \mathfrak{f}_P(J)$. This says, among other, that if two linguists agree on the "language" L, yet they disagree on their observations I_1 on the one hand and I_2 on the other, they will find an I_3, on which they both agree and which still yields the language L.

Contrary to what one might think, characteristicity is a highly non-trivial requirement. And in fact, one might argue that it is way to strong: because it makes the presupposition is that I_1, I_2 induce the same language. This however might already by undecidable for many pre-theories we consider! That in turn would make the strong requirement practically useless. We therefore formulate a more careful version of this property, which addresses the same problem in a fashion which is more satisfying:

Definition 30 *A pre-theory* $(\mathfrak{f}, P)$ *is* ***downward normal****, if for any* I_1, I_2 *such that* $\mathfrak{f}_P(I_1) \supseteq I_2$, $\mathfrak{f}_P(I_2) \supseteq I_1$, *there exists a* $J \subseteq I_1 \cap I_2$, *such that* $\mathfrak{f}_P(J) \supseteq I_1 \cup I_2$.

Note that downward normality and characteristicity do not imply each other in any direction. It is however clear that downward normality solves the above problem in a very satisfying manner: assume we disagree over the symmetric difference $I_1 \Delta I_2$ (we put $M \Delta N := (M \cup N) - (M \cap N)$). Then there is a solution with data we both agree on, such that still any string in $I_1 \Delta I_2$ is contained in the "language" resulting from projection. So we can reject arguable data for projection, but we always find a way to make sure that the strings are still part of "language". Because this is fully effective – provided the pre-theory is decidable – we will prefer this property over characteristicity.

So the question is: what are the requirements for analogies and inferences to make sure that pre-theories are characteristic or downward normal? This is a difficult question, and all our criteria so far are insufficient.

Consider the pre-theory $(\mathfrak{g}, P1)$, where we allow for an analogy only if $\vec{w} \leq_I \vec{v}$ (that is, the set of contexts in which the two occur are in inclusion relation) and $\vec{w} \sqsubseteq \vec{v}$ (provided $\vec{w} \neq \vec{v}$. These conditions are not sufficient: take $I_1 := \{axb, cxd, aixjb\}$, $I_2 := \{axb, cxd, cixjd\}$. We can make the analogy $x \Leftarrow ixj$ in both languages; in $I_1 \cap I_2 = \{axb, cxd\}$ we cannot, as we have no occurrence of the substring ixj, nor can we in any still smaller language. So $P1$

is neither characteristic nor upward normal, and we need stronger requirements. The following is less obvious.

Lemma 31 $(\mathfrak{g}, Pr)$ *is not characteristic and not downward normal.*

Proof. For illustration purposes, we show this with two typical examples.

Counterexample 1 to characteristicity. Consider the following two languages: $I_1 := \{ab, aabb, aaabbb\} \cup \{cb, ccbb\} \cup \{bbb, dbbbd\}$; $I_2 := \{ab, aabb\} \cup \{cb, ccbb, cccbbb\} \cup \{bbb, dbbbd\}$. We have $I_1 \cap I_2 = \{ab, aabb, cb, ccbb, bbb, dbbbd\}$. In this case, we have $bbb \approx^{Pr}_{I_1 \cap I_2} dbbbd$, but we do not get this similarity in I_1 or I_2. Yet, there is no $J \subseteq I_1 \cap I_2$, such that $bbb, dbbbd \in J$ but $bbb \not\approx^{Pr}_{J} dbbbd$, as can be easily seen (or checked by hand).

Counterexample 2 to characteristicity and downward normality. Put $I_1 = \{ab, aabb, aaabbb, xaaaxbbb\}$, $I_2 = \{ab, aabb, xaaaybbb, xxaaayybbb\}$. We have $\mathfrak{g}_{Pr}(I_1) = \mathfrak{g}_{Pr}(I_2)$, as can be easily checked, but for $I_1 \cap I_2 = \{ab, aabb, xaaaxbbb\}$, we have $\mathfrak{g}_{Pr}(I_1 \cap I_2) \subsetneq \mathfrak{g}_{Pr}(I_1)$. □

So how can we ensure the two properties? The way to characteristicity is long and complicated: basically, we have to ensure that an infinite language uniquely encodes the smallest finite language that induces it. This is feasible with more or less reasonable methods; yet, it is somehow counterintuitive from my point of view: linguistic metatheory is all about having finite objects and constructing infinite objects; whereas characteristicity is more about having infinite objects and constructing (showing the existence) of finite objects. Even worse, we cannot even claim to *have* the infinite objects: maybe its relevant properties cannot be simply read off from our finite characterization. So it is the concern about our commitment to finitary procedures which speaks most strongly in favor of downward normality as opposed to characteristicity. We will therefore only describe an approach to obtain downward normality.

What is most problematic about downward normality is that analogies are permitted or prohibited by global properties of the language: we always have to consider all strings of a language, unless of course we have some additional information about them, such as that they do not contain a certain substring. We will now present a pre-theory, where analogies can be determined *locally*, that is, we can allow for a certain analogy in a certain context, though not in some other context, and we can compute them only by looking at a certain subset of the language.

We say a string $\vec{w}$ is **elementary**, if it does not contain any substring of the form $\vec{x}_1\vec{x}_1\vec{x}\vec{x}_2\vec{x}_2$, where $\vec{x} \neq \epsilon \neq \vec{x}_1\vec{x}_2$. We define the analogical map $P2$ as follows:

$(\vec{w}\vec{x}\vec{v}, \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v}) \in P2(I)$, if

1. $\vec{w}\vec{x}\vec{v}, \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v} \in I$,
2. $\vec{w}\vec{x}\vec{v}$ is elementary, and
3. (there is no $\vec{z} \in I$ such that $\vec{w}\vec{x}\vec{v} \sqsubset \vec{z} \sqsubset \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v}$.)

Next, we define the inference (meta-)rules $\mathfrak{g}2$; note that here and in the sequel, we have the convention that $\vec{w}$ represents a possibly bracketed string; to refer to its unbracketed version, we write $h(\vec{w})$:

$$\frac{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{g}2_{P2}(I) \quad h(\vec{w}\vec{x}\vec{v}) \Leftarrow_I^{P2} h(\vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v}) \quad \vec{x} \in \Sigma^*}{\vdash w(\vec{x}_1(\vec{x})\vec{x}_2)\vec{v} \in \mathfrak{g}2_{P2}}, \tag{4.15}$$

where $(,) \notin \Sigma$, and h is the usual homomorphism mapping $(,) \mapsto \epsilon$. As we can see, our relational language thus comprises statements of the form $\vec{x} \in \Sigma$; we thus need a unary relation $R = \Sigma^*$ in the language-theoretic structure. This scheme is complicated because we need the distinction between the bracketed string on the one side (for judgments) and the unbracketed string for analogies on the other on other; moreover, we want to make sure that the string $\vec{x}$ does not contain any brackets (the third premise). Say a pair of brackets $(,)$ in a string $\vec{w}(\vec{x})\vec{v}$ is simple, if $\vec{x}$ does not contain any brackets. This first rule is necessary to introduce simple brackets, so to speak, because the next rule presupposes them:

$$\frac{\vec{w}(\vec{x}_1(\vec{x})\vec{x}_2)\vec{v} \in \mathfrak{g}2_{P2} \quad \vec{w}'\vec{x}\vec{v}' \Leftarrow_I^{P2} \vec{w}'\vec{x}_1\vec{x}\vec{x}_2\vec{v}'}{\vec{w}(\vec{x}_1(\vec{x}_1(\vec{x})\vec{x}_2)\vec{x}_2)\vec{v} \in \mathfrak{g}2_{P2}}, \tag{4.16}$$

where $\vec{w}', \vec{v}'$ are arbitrary and unrelated to $\vec{w}, \vec{v}$. So once we have introduced a bracketing of the form $(\vec{x}_1(\vec{x}_1(\vec{x})\vec{x}_2)$, we can expand it without contextual restrictions. This scheme thus only requires that there exists *some* context in which the analogy is legitimate. Note that the second premise makes sure that $\vec{x}, \vec{x}_1, \vec{x}_2$ to not contain any brackets, together with the third and last inference rule of $\mathfrak{g}2$:

$$\frac{\vec{w}\vec{x}\vec{v} \approx_I^{P2} \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v}}{\vec{w}\vec{x}\vec{v} \Leftarrow_I^{P2} \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v}.} \tag{4.17}$$

This is sufficient. Note that already $\approx^{P2}$ is asymmetric; we could make it symmetric and only allow asymmetric analogies, but we skip this for reasons of simplicity. We first make the following observation:

Lemma 32 *Let I, J be two finite languages. If $I \subseteq J$, then $\mathfrak{g}2_{P2}(I) \subseteq \mathfrak{g}2_{P2}(J)$.*

We will call this property **monotonicity**, and later on devote a proper subsection to it.

Proof. It is clear that $P2(I) \subseteq P2(J)$, because if $\vec{w}, \vec{v} \in I$, then $\vec{w}, \vec{v} \in J$. Also, the set of premises for $\mathfrak{g}2_{P2}(I)$ is a subset of $\mathfrak{g}2_{P2}(J)$. □

Lemma 33 *Assume that $h(\vec{w}) \notin I$, $\vec{w} \in \mathfrak{g}2_{P2}(I)$. Then $h(\vec{w})$ is not elementary.*

Proof. Assume that $h(\vec{w})$ does not contain a substring of the form $\vec{x}_1\vec{x}_1\vec{x}\vec{x}_2\vec{x}_2$, with $\vec{x} \neq \epsilon \neq \vec{x}_1\vec{x}_2$. If $\vec{w} = h(\vec{w})$, then $\vec{w} \in I$ and the claim follows. Assume that $\vec{w}$ contains substrings of the form $(\vec{x}_1(\vec{x})\vec{x}_2)$. Then each of these substrings has been introduced by an inference which, by the last definitions, must have the form

$$\frac{\vdash \vec{w}_1\vec{x}\vec{w}_2 \in \mathfrak{g}2_{P2}(I) \quad h(\vec{w}_1\vec{x}\vec{w}_2) \Leftarrow_I^{P2} h(\vec{w}_1\vec{x}_1\vec{x}\vec{x}_2\vec{w}_2)}{\vdash \vec{w}_1(\vec{x}_1(\vec{x})\vec{x}_2)\vec{w}_2 \in \mathfrak{g}2_{P2}(I)},$$

which presupposes that $h(\vec{w}_1\vec{x}_1\vec{x}\vec{x}_2\vec{w}_2) \in I$ (check the definition of $P2$). So if $h(\vec{w})$ does not contain a substring of the form $\vec{x}_1\vec{x}_1\vec{x})\vec{x}_2\vec{x}_2$, then $\vec{w} \in I$, and by contraposition the claim follows. □

We can now show the following:

Theorem 34 $(\mathfrak{g}2, P2)$ *is downward normal.*

Proof. Assume we have I_1, I_2 with $h \circ \mathfrak{g}2_{P2}(I_1) \supseteq I_2, h \circ \mathfrak{g}2_{P2}(I_2) \supseteq I_1$. We show that every string $\vec{w} \in I_1 \cup I_2$ can be derived by a subset of $I_1 \cap I_2$. By the above monotonicity result, the claim then follows.

We prove only one part, namely that $I_1 \subseteq \mathfrak{g}2_{P2}(I_1 \cap I_2)$; the proof for $I_2 \subseteq \mathfrak{g}2_{P2}(I_1 \cap I_2)$ is identical. We do this by an induction on the strings of I_1, using the partial order $\sqsubseteq_\omega$. By $\sqsubseteq_\omega$ we denote the *scattered substring* relation, that is, we have $\vec{w} \sqsubseteq_\omega \vec{v}$, iff $\vec{w} = \vec{w}_1 ... \vec{w}_i$ and $\vec{v} = \vec{v}_1 \vec{w}_1 \vec{v}_2 ... \vec{v}_i \vec{w}_i \vec{v}_{i+1}$. Importantly, we make an induction on the strong language containing brackets, where the crucial step is the following: we show that for every $\vec{w} \in I_1$, $\vec{v} \in \mathfrak{g}2_{P2}(I_2)$ such that $h(\vec{v}) = \vec{w}$, we have $\vec{v} \in \mathfrak{g}2_{P2}(I_1 \cap I_2)$.

The induction base is clear: every $\sqsubseteq_\omega$ minimal string of I_1 is in $I_1 \cap I_2$, because inferences strictly increase string length.

Induction hypothesis: take a $\vec{v} \in I_1$, $\vec{w} \in \mathfrak{g}2_{P2}(I_2)$ such that $h(\vec{w}) = \vec{v}$, and assume the claim holds for all $\vec{w}' \in \mathfrak{g}2_{P2}(I_2)$ such that $\vec{w}' \sqsubset_\omega \vec{w}$ and $h(\vec{w}') \in I_1$.

Case 1: $\vec{w}$ contains no brackets – in this case, we have $\vec{w} \in I_2$, and as $\vec{w} \in I_1$, we have $\vec{w} \in I_1 \cap I_2$, and thus $\vec{w} \in \mathfrak{g}2_{P2}(I_1 \cap I_2)$.

Case 2: assume $\vec{w}$ can be derived in a derivation whose last step is

$$\frac{\vdash \vec{w}_1 \vec{x} \vec{w}_2 \in \mathfrak{g}2_{P2}(I_1) \quad h(\vec{w}_1 \vec{x} \vec{w}_2) \Leftarrow^{P2}_{I_2} h(\vec{w}_1 \vec{x}_1 \vec{x} \vec{x}_2 \vec{w}_2) \quad \vec{x} \in \Sigma^*}{\vdash \vec{w}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(I_1)},$$

and thus $\vec{w} = \vec{w}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{w}_2$ In this case, we know that $h(\vec{w}_1 \vec{x} \vec{w}_2), h(\vec{w}_1 \vec{x}_1 \vec{x} \vec{x}_2 \vec{w}_2) \in I_2$, because otherwise there could not be an analogy as the one above. Moreover, by assumption, we know that $h(\vec{w}_1 \vec{x}_1 \vec{x} \vec{x}_2 \vec{w}_2) \in I_2 \cap I_1$. But we also know that $h(\vec{w}_1 \vec{x} \vec{w}_2) \in I_1$: as it is in I_2, it must be by assumption in $\mathfrak{g}2_{P2}(I_1)$, and as it is the left-hand side of an analogy, it has to be elementary. So if $h(\vec{w}_1 \vec{x} \vec{w}_2) \notin I_1$, we contradict the last lemma. Furthermore, by induction hypothesis, we have $\vec{w}_1 \vec{x} \vec{w}_2 \in \mathfrak{g}2_{P2}(I_1 \cap I_2)$. So the same analogy works with $I_1 \cap I_2$. So we also have $\vec{w}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(I_1 \cap I_2)$.

Case 3: there is a derivation of $\vec{w}$, the last step of which is

$$\frac{\vec{w}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(I_2) \quad \mathbf{a}}{\vdash \vec{w}_1 (\vec{x}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(I_2),}$$

where $\mathbf{a}$ is an appropriate analogy. We know that $\vec{w}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(V)$ for some $V \subseteq I_1 \cap I_2$ by induction hypothesis; we also know that we have an appropriate analogy over V at hand, because otherwise we could not have introduced the brackets. Consequently, we have $\vec{w}_1 (\vec{x}_1 (\vec{x}_1 (\vec{x}) \vec{x}_2) \vec{x}_2) \vec{w}_2 \in \mathfrak{g}2_{P2}(V)$.

Note, by the way, that the three cases are not mutually exclusive, but they cover all possibilities. This proves the induction step, and shows that any string $\vec{w} \in \mathfrak{g}2_{P2}(I_2)$ such that $h(\vec{w}) \in I_1$ is actually also in $\mathfrak{g}2_{P2}(I_1 \cap I_2)$. □

This proof even gives us a stronger corollary:

Corollary 35 *Assume we have I_1, I_2 with $h \circ \mathfrak{g}2_{P2}(I_1) \supseteq I_2, h \circ \mathfrak{g}2_{P2}(I_2) \supseteq I_1$. Then for every bracketed string $\vec{w}$ such that $h(\vec{w}) \in I_1 \cap I_2$, we have $\vec{w} \in \mathfrak{g}2_{P2}(I_1)$ iff $\vec{w} \in \mathfrak{g}2_{P2}(I_2)$.*

This shows us how a downward normal pre-theory has to look like. The crucial properties we need to obtain this result are firstly monotonicity, secondly the fact that analogies are restricted to contexts, and thirdly another peculiarity about analogies: the analogies introducing certain brackets actually presuppose that exactly the strings, which are being inferred, are in the language in question. I do not think that $(\mathfrak{g}2, P2)$ is uninteresting as such; however, I think it is preferable to not have these properties, that is, it is more interesting to compute analogies globally regardless of the context in which they occur. But of course it is also much more challenging, as we cannot easily make statements about $\mathfrak{g}_{Pr}(I)$ without observing I as a whole. This is the main reason that we do focus here on $(\mathfrak{g}, Pr), (\mathfrak{g}, P1)$ in the sequel.

4.6.2 Upward Normality

Another important property is what we will call **upward normality**

Definition 36 *A pre-theory $(\mathfrak{f}, P)$ is **weakly upward normal**, if for every infinite L such that there is a finite I with $L = \mathfrak{f}_P(I)$, there are infinitely many distinct (finite) $I', I'', ...$ such that $\mathfrak{f}_P(I') = \mathfrak{f}_P(I'') = ... = L$.*

Note that the finiteness of I', I'' actually follows under our assumption that pre-theories map infinite languages onto their identity.

The concept of upward normality is important for the following reason: If we observe a finite language and construct an infinite language out of it, then it is very plausible that there are many fragments of L which can induce the same language L; in fact, there should be infinitely many. For otherwise, our "language" L is plausible under a pre-theory $(\mathfrak{f}, P)$ only if we have a bounded number of observations; but this makes it implausible *a priori*: because we are interested in pre-theories and languages, such that the languages are induced by datasets, to which there is *no* upper bound. In fact, this is one of the major premises of linguistic metatheory. It would be ridiculous to probably any linguist to say: we think that "language" looks like this, but if we make more than 1000 observations, we can no longer think so - regardless of what we observe! To prevent this situation, we have weak upward normality: for every infinite language induced by $(\mathfrak{f}, P)$, there are infinitely many finite languages inducing it under P.

This requirement seems trivial from a linguistic point of view, but actually it is from a mathematical viewpoint:

Lemma 37 *$(\mathfrak{g}, Pr)$ is not weakly upward normal.*

Proof. Take $I := \{xyz, xxyyz, xyyzz, xxyyyzz, yyy\}$. We have
$Pr(I) = \{(xy, xxyy), (yz, yyzz), (yyy, xxyyyzz)\}$.

Now assume (*case 1*) we add a string in $(xx)^j yyy(zz)^j$, where $j \geq 2$. Then we lose the analogies $(xy, xxyy)$ and $(yz, yyzz)$. We can of course add strings to allow new analogies deriving the same strings; these analogies must have the form $(x^k y^k, x^{k+i} y^{k+i})$, where $k \geq j$. But then in turn we lose the analogy generating the sublanguage $\{(xx)^n yyy(zz)^n : n \in \mathbb{N}\} \subseteq \mathfrak{g}_{Pr}(I)$, and we can only restore it by adding$((xx)^{k'} yyy(zz)^{k'}, (xx)^{k'+i'} yyy(zz)^{k'+i'}$, where $k' > k$, thereby again preventing the analogies generating $\{x^n y^n : n \in \mathbb{N}\}$, and so on. This argument can be iterated arbitrarily. The argument also clear why *case*

2 and *case 3*, where we add a string derivable by the analogy $(xy, xxyy)$ or $(yz, yyzz)$ are completely parallel.

Consequently, there is no finite language $J \supsetneq I$ such that $\mathfrak{f}_{Pr}(I) = \mathfrak{f}_{Pr}(J)$. □

So for "restrictive" pre-theories as Pr, weak upward normality is in fact a problem. We will now introduce an even stronger requirement, the one of **strong upward normality** or simply upward normality.

Definition 38 *A pre-theory P is **upward normal** (in the strong sense), if for every infinite language L such that $\mathfrak{f}_P(I) = L$ for some finite I, the following holds: for every finite $J \subseteq L$, there is a finite $J' \supseteq J$ such that $\mathfrak{f}_P(J') = L$.*

Obviously, strong upward normality entails weak upward normality, but is much stronger: whereas in weak upward normality, there only needs to be some arbitrarily large extension inducing the same language, for strong upward normality we must be able to extend in "every direction", so to speak. Upward normal in this strong sense means: we cannot have too much data regarding a language; whatever fragment of L we observe, there is a larger fragment which convinces us that we are observing fragments of L. No finite fragment of L is convincing evidence *against* L in the sense that it excludes it as a candidate "language". This is obviously motivated by the following fact: given a presumable "language" L, we reasonably assume *a priori* that we can observe arbitrary and arbitrarily large fragments thereof.[3] Now the fact that we can observe arbitrary fragments of "language" should make us exclude languages which are excluded by fragments we can observe, because this is a contradiction, and it seems to fundamentally contradict our sense of scientific positivism.

There are some points to note. Firstly, strong upward normality does not say anything about convergence or about the fact that the larger the fragment of L we observe, the more "plausible" L becomes. On the contrary, it might happen that a fragment leads us onto the "wrong track" in the following sense: Given a sequence of finite fragments I_i, such that for all $i \in \mathbb{N}$, $I_i \subseteq L$, $i \leq j \Rightarrow I_i \subseteq I_j$, $|I_i| = i$, L a language induced by $(\mathfrak{f}, P)$, it might be that for each I_i, the smallest $I_j \supseteq I_i$ such that $\mathfrak{f}_P(I_j) = L$ has size at least k^i. That is, we need exponentially many strings to lead us back onto the right track. But this is not the kind of question or problem we are interested in at this point.

Corollary 39 $(\mathfrak{g}, Pr)$ *is not upward normal (in the strong sense).*

This is because upward normality obviously entails weak upward normality, but not vice versa, as we have seen. Consequently, upward normality is also highly non-trivial, and we find plenty of counterexamples. Upward normality even fails to hold for much less restrictive pre-theories such as $P1$:

Lemma 40 $(\mathfrak{g}, P1)$ *is not (strongly) upward normal.*

Proof. Take the finite language $I := \{ab, xxaxxb, ayybyy, xy, xxyy\}$. We have

$P1(I) = \{(a, xxaxx), (b, yybyy), (xy, xxyy)\}$.

If we however put $I' = I \cup \{xxaxxyybyy\} \subseteq \mathfrak{g}_{P1}(I) =: L$, then we have $xy \not\approx^{I'}_{P1} xxyy$, because $xy \not\leq_{I'} xxyy$. We cannot restore this analogy by adding

[3] Though this is not necessarily the case: compare our discussion on o-language and "language".

the string $xxaxybyy$, as it is not in L. On the other side, we can easily enrich the language as to allow for additional analogies which work to the same effect, allowing to derive $\{x^n y^n : n \in \mathbb{N}\}$. For example, we can define $I'' = I' \cup \{xxyyy\}$. Now we have $(xxyy, xxxyyy) \in P1(I'')$, and this is fine; but the problem is now the following: we can apply this analogy also to the string $xxaxxyybyy$, thereby deriving $\{xxax^n xyy^n byy : n \in \mathbb{N}\}$, which is not in L. So the problem is that our resulting language is too big rather than to small.

Is there a way out of this problem? The answer can be shown to be negative. Assume we have a finite language $J : I' \subseteq J \subseteq \mathfrak{g}_{P1}(I)$. In order to make sure that $\{x^n y^n : n \in \mathbb{N}\} \subseteq \mathfrak{g}_{P1}(J)$, we need some strings of the form $x^n y^n \in J$; and assume that $x^k y^k \in J$ is the longest of these strings. Then there are two possibilities:

Case 1: we do not have $\vec{x}a\vec{x}\vec{y}b\vec{y} \in J$, such that $x^k y^k \sqsubseteq \vec{x}\vec{y}$. In this case, $(xy, x^k y^k) \in P1(J)$, and so, we can derive strings from $xxaxxyybyy$ which are not in L, as above.

Case 2: we do have $\vec{x}a\vec{x}\vec{y}b\vec{y} \in J$, such that $x^k y^k \sqsubseteq \vec{x}\vec{y}$. In this case, whatever analogies we have that allow us to derive the strings $\{x^n y^n : n \in \mathbb{N}\}$, they are also applicable to $\vec{x}a\vec{x}\vec{y}b\vec{y}$, thereby again deriving strings not in L. □

So upward normality is quite problematic for pre-theories, though it is a most natural requirement. A reason for that is that so far we have used the relation $\leq_L$. But this relation, as we have seen, is not preserved under projection, and in the worst case, it might even become undecidable under projection. This means it also becomes undecidable for upward normality: assume that $L = \mathfrak{f}_P(I)$ for some P based on $\leq_I$. The question whether for every finite $J \subseteq L$, there is a finite $J' \supseteq J$ such that $\vec{w} \leq_{J'} \vec{v}$ might turn out to be undecidable, as well as it might be undecidable for L itself.

Lemma 41 *Given a CFL L, the question whether for every finite fragment $I \subseteq L$, there is a (finite) $J : L \supseteq J \supseteq I$ such that $\vec{w} \leq_J \vec{v}$ is undecidable.*

Proof. If for some finite $I \subseteq L$ there is no finite $J \supseteq I$ such that $\vec{w} \leq_J \vec{v}$, then $\vec{w} \not\leq_L \vec{v}$, for the following reason: assume $\vec{w} \leq_L \vec{v}$. Then for every $\vec{x}\vec{w}\vec{y} \in I$, there is $\vec{x}\vec{v}\vec{y} \in L$. Take the set of these strings, which is finite. This contradicts the assumption.

Conversely, assume we have for every finite I a finite $J \subseteq L$ such that $\vec{w} \leq_J \vec{v}$. Then we have $\vec{w} \leq_L \vec{v}$. For assume we do not have $\vec{w} \leq_L \vec{v}$. Then there is $\vec{x}\vec{v}\vec{y} \in L$, $\vec{x}\vec{w}\vec{y} \notin L$. So for the finite set $I = \{\vec{x}\vec{v}\vec{y}\}$, there is no finite $J \subseteq L$ such that $\vec{w} \leq_J \vec{v}$, otherwise we would have $\vec{x}\vec{w}\vec{y} \in J$ and so $J \not\subseteq L$. Contradiction.

Thereby, we see that we have $\vec{w} \leq_L \vec{v}$, for a CFL L, exactly if for every finite set $I \subseteq L$, there is a finite $J : I \subseteq J \subseteq L$, such that $\vec{w} \leq_J \vec{v}$. This in turn means the decision problems are equivalent, and as the problem "for L a CFL, is $\vec{w} \leq_L \vec{v}$?" is undecidable, so is our current problem. □

On the other side, there are few meaningful alternatives to $\leq_L$ in substitutional pre-theories. So if we want upward normality, there seems to be only one reasonable way to go: when considering a finite language J, we first *reduce* it to some relevant fragment thereof, and only *afterwards* project it.

As some simplistic solutions for characteristicity (as injective pre-theories) have shown us the right way to go, we might want to look simplistic solutions for upward normality. A first upward normal pre-theory might be constructed on the basis of the following: Take a pre-theory $(\mathfrak{f}, P)$ and fix a constant $k \geq 0$. Define

the map p_k, mapping stringsets on stringsets by $p_k(I) = \{\vec{w} \in I : |w| \leq k\}$. We can now define $P^k := P \circ p_k$. This does not ensure upward normality: for $(\mathfrak{g}, Pr^k)$, take the language $I := \{aaa, aaaa, ab, aabb\}$. If we add $aaabbb, aaaabbbb$ to the language, even if they do not have relevance for the analogies, they allow to apply the analogy to new strings like $a^n b^m$ for $n > m$, where we use the additional strings as premises for inferences, not analogies.

So in order to get upward normality, we must extend this map to the premises, for example by defining $(\mathfrak{f}^k, P^k)$, such that $\mathfrak{f}^k_{P_k} := \mathfrak{f}_P \circ p_k$. But in this case we obviously fail a fundamental requirement, namely the requirements that our maps be *increasing*, that is, that we have $I \subseteq \mathfrak{f}_P(I)$. So this is highly unsatisfying, as it does not even define a projection

How can we proceed? There is another, more elegant approach. We mostly look at pre-theories based on $\leq_L$. Take an integer $k \geq 0$, and define the relation $\leq^k_L$ as follows:

Definition 42 *For $L \subseteq \Sigma^*$, $\vec{w} \leq^k_L \vec{v}$ if and only if for all $(\vec{x}, \vec{y}) \in \Sigma^* \times \Sigma^*$, if $\vec{x}\vec{w}\vec{y} \in L$ and $|\vec{x}\vec{w}\vec{y}| \leq k$, then $\vec{x}\vec{v}\vec{y} \in L$.*

So $\leq^k_L$ is some kind of restriction of $\leq_L$, but not in a set-theoretic sense: in general, we can both have $\leq_L \not\subseteq \leq^k_L$ and $\leq^k_L \not\subseteq \leq_L$ for a given k and a given language L. So from a set-theoretic point of view, the relations are incomparable. Now take a pre-theory $P_{\leq_L}$, which uses $\leq_L$, and change all occurrences of $\leq_L$ in the conditions to $\leq^k_L$, thereby obtaining $P_{\leq^k_L}$. This alone does not give us upward normality for the same reason as above: using inference f, either we can derive additional strings with the new premises (though they do not play any role for the analogies), or there are strings we cannot recover.

We can solve this problem as follows: assume we have a pre-theory $(\mathfrak{f}, P)$, and, for a given alphabet Σ, a finite set of (infinite) languages $\{L_t : t \in T\}$, with $|T| \leq k$, which satisfies the following criteria:

1. $\bigcup_{t \in T} L_t = \Sigma^*$;
2. for each $t \in T$, there is a finite $I \subseteq \Sigma^*$ such that $\mathfrak{f}_P(I) = L_t$;
3. for each $t, t' \in T$, $L_t \cap L_{t'}$ is finite.

The motivation behind this definition is as follows: the first condition is to say: no observation is impossible; the second one says: every one of these languages is finitely induced, and the third condition is to make sure that each of the languages can be uniquely characterized by a finite set. We can also avoid making the alphabet explicit at this point, by defining a function $\mathcal{L}$ from any alphabet to a finite set of infinite languages over this alphabet such that for any Σ, $\mathcal{L}(\Sigma)$ is a set of languages satisfying the above constraint. We now devise a projection as follows:

$$\mathfrak{f}^{\mathcal{L}}_P(I) = \begin{cases} L_t, \text{ if } t' \neq t \Rightarrow I \not\subseteq L_{t'}, \\ I \text{ otherwise.} \end{cases} \tag{4.18}$$

By this procedure, we can make any pre-theory upward normal. Note that we did not actually define the pre-theory here, only the projection to which it gives rise. The reason we can grant this is the following: we will show later on, that *every* projection can be formalized as a pre-theory. So writing a projection

as a pre-theory is in this sense rather an exercise, which can be tedious and not be very instructive, and therefore we skip it at this point. That does of course not mean that it is useless to look at pre-theories at all: mostly, we define pre-theories for their plausibility, and then look to which projections they give rise.

Now, the price we pay for upward normality of $(\mathfrak{f}^{\mathcal{L}}, P)$ is obvious: we get a finite number of possible languages. Note that this is maybe not as bad as it seems: many linguistic schools of thought have tried to restrain the space of possible languages, and the mainstream generative school has tried to cut them even down to a finite number (modulo some considerable abstraction).

From our point of view, this assumption has some interesting consequences:

Lemma 43 *Let $\mathfrak{f}_P$ be a projection, such that there are only finitely many infinite languages induced by $\mathfrak{f}_P$ over a given alphabet Σ. Then there is a finite set $J \subseteq \Sigma^*$ such that for all $I : J \subseteq I$, we have either $\mathfrak{f}_P(I)$ is finite, or we have $\mathfrak{f}_P(I) = \Sigma^*$.*

Proof. Let Σ^*_{fin} denote all finite subsets of Σ^*; it is thus a subset of $\wp(\Sigma^*)$.

Case 1: $\bigcup_{I \in \Sigma^*_{fin}} \mathfrak{f}_P(I) \subsetneq \Sigma^*$. Then there is a finite set $J \subseteq \Sigma^*$ which is contained in no infinite language induced by $\mathfrak{f}_P$. Consequently, if $J \subseteq I$, I finite, then $\mathfrak{f}_P(I)$ is finite. Note that in this case, J can be chosen to be a singleton $\{\vec{w}\}$, thereby strengthening the result.

Case 2: $\bigcup_{I \in \Sigma^*_{fin}} \mathfrak{f}_P(I) = \Sigma^*$. Let $L_t : t \in T$ be the languages induced by $\mathfrak{f}_P$ over Σ^*. For all $L_t : t \in T$, if $L_t \neq \Sigma^*$, choose a $\vec{w} \notin L_t$. This yields a finite set J. Assume (i) there is an $L_t = \Sigma^*$. In this case, we must have for $I \supseteq J$, $\mathfrak{f}_P(I) = \Sigma^*$ or $\mathfrak{f}_P(I)$ finite, because there is no other possible image. Otherwise (ii) if there is no $L_t = \Sigma^*$, for any $I \supseteq J$, $\mathfrak{f}_P(I)$ must be finite. □

The main point in the proof is that there are infinitely many finite languages over Σ. This result seems to be of some relevance to a view of language as the one put forward by the principles and parameters program, which is somewhat in line with the approach of $(\mathfrak{f}^{\mathcal{L}}, P)$ laid out above: it says that the number of possible languages – modulo lexicon – are finite. This entails thus that we find sets of strings of the above kind, which is very implausible in the first place. But of course, in applying this purely language-theoretic result to a linguistic theory one has to be very careful, and we will not work this out at this point. Anyway, this result shows that pre-theories which induce only finitely many infinite languages have some properties we would judge as disfavorable. So this road is not very appealing.

And so the question remains open: how can we devise an interesting pre-theory which is strongly upward normal?

4.6.3 Normalizing Maps

We see that for upward normality, the challenge is to obtain an upward normal pre-theory which induces infinitely many infinite languages. On this occasion, we can also note the following: we have said that every injective pre-theory is characteristic, but as is easy to check, an injective pre-theory *cannot* be upward normal, because for every infinite L there is at most one finite I generating it.

We will now use a **normalizing map**, which first reduces a language, to get a subset whose projection is larger. We first define the linear radix order *rad* on Σ^* by first using length and then a lexicographic ordering; that is, we

presuppose a linear order $\prec_\Sigma$ on Σ, and define $\vec{w}\ rad\ \vec{v}$ iff $|\vec{w}| < |\vec{v}|$ or $|\vec{w}| = |\vec{v}|$, $\vec{w} = \vec{x}a\vec{y}, \vec{v} = \vec{x}b\vec{z}$, and $a \prec_\Sigma b$. This order is somewhat arbitrary, so there is no particular importance of this choice; the only important thing is that rad is well-founded, that is, for every element we want the set of its predecessors to be finite (the lexicographic ordering for example is not well-founded).

We extend rad onto subsets of Σ^* by defining $rad^* \subseteq (\wp(\Sigma^*)^2$ as follows: $M rad^* N$, iff the rad-largest element of $M \Delta N$ (symmetric difference) is in N. Note that this trivially subsumes the case $M \subseteq N$. Given a set $M \subseteq \Sigma^*$, we also denote its rad-largest element by $max_{rad}(M)$; for sets of sets, we adopt the same convention for rad^*.

Now we define the map $p : \Sigma^*_{fin} \to \Sigma^*_{fin}$ mapping finite languages onto finite languages. Put $per_{(\mathfrak{f},P)}(I) := \{M \subseteq I : (I) \subseteq \mathfrak{f}_P(I - M)\}$, the set of subsets of strings whose subtraction does not prevent the projected language from including the original language. We now define $p_{(\mathfrak{f},P)}(I) := I - max_{rad^*}(per_{(\mathfrak{f},P)}(I))$. That is, we take the rad-maximal set of peripheral strings and subtract it from I. Note that the map $p_{(\mathfrak{f},P)}$, contrary to appearances, is *not* a fixed point map, because $p(I)$ is the rad-smallest set inducing a language larger than I; but $p(p(I))$ is the smallest inducing a language larger than $p(I)$, not I! Consider the following example:

Example 44 *Put* $I := \{ab, aabb, aaabbb, aaaxbbbx, aaaxxbbbxx\}$. *Then we have*
$p_{(\mathfrak{g},Pr)}(I) = \{ab, aabb, aaabbb, aaaxbbbx\}$, *and*
$p_{(\mathfrak{g},Pr)}(p_{(\mathfrak{g},Pr)}(I)) = \{ab, aabb, aaaxbbbx\}$.
It is easy to see that $\mathfrak{g}_{Pr}(p_{(\mathfrak{g},Pr)}(p_{(\mathfrak{g},Pr)}(I))) \not\supseteq I$.

The most important property of p is the following:

Lemma 45 *If* $I \subseteq J \subseteq \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I)$, *then* $p_{(\mathfrak{f},P)}(I) = p_{(\mathfrak{f},P)}(J)$.

Proof. Recall that rad^* is a linear order on Σ^*. Moreover, recall that $p_{(\mathfrak{f},P)}(I)$ is the rad^*-smallest subset of of I inducing a language larger than I (if we subtract the rad^*-largest, we get the rad^*-smallest).

Now assume the claim does not hold, and $p_{(\mathfrak{f},P)}(I) \neq p_{(\mathfrak{f},P)}(J)$. Then there are two cases:

Case 1: Assume that $p_{(\mathfrak{f},P)}(I) rad^* p_{(\mathfrak{f},P)}(J)$. But then, as $\mathfrak{f}_P(p_{(\mathfrak{f},P)}(I)) \supseteq J$, it follows that $p_{(\mathfrak{f},P)}(J)$ is not the rad^* minimal language inducing a larger language than J; contradiction.

Case 2: Assume that $p_{(\mathfrak{f},P)}(J) rad^* p_{(\mathfrak{f},P)}(I)$. But as $I \subseteq J$, we have $\mathfrak{f}_P(p_{(\mathfrak{f},P)}(J)) \supseteq J \supseteq I$, and thus $p_{(\mathfrak{f},P)}(I)$ is not the rad^* minimal language inducing a larger language than I; contradiction

□

This is the crucial lemma; note however that the following, stronger claim is wrong:

Lemma 46 *Assume* $I \subseteq J \subseteq \mathfrak{f}_P(I)$. *Then we do not necessarily have* $p_{(\mathfrak{f},P)}(I) = p_{(\mathfrak{f},P)}(J)$.

Proof. Just take the language $I = \{ab, aabb, aaabbb, aaaxbbbx\}$, $J = I \cup \{aaaxxbbbxx\}$. Then we have $p_{(\mathfrak{g},Pr)}(I) = \{ab, aabb, aaaxbbbx\}$, $p_{(\mathfrak{g},Pr)}(J) = \{ab, aabb, aaabbb, aaaxbbbx\}$. □

We now have to integrate this into a pre-theory. We can use any pre-theory we like; we just have to make sure, that the inferences of the form

$$\frac{\vdash \vec{w} \in I}{\vdash \vec{w} \in \mathfrak{f}_P(I)} \tag{4.19}$$

are changed to the form

$$\frac{\vdash \vec{w} \in p_{(\mathfrak{f},P)}(I)}{\vdash \vec{w} \in \mathfrak{f}_P(I)} \ . \tag{4.20}$$

This allows us to define the pre-theory $(p\mathfrak{f}, P \circ p_{(\mathfrak{f},P)})$ for any pre-theory $(\mathfrak{f}, P)$. The first result we obtain is the following:

Theorem 47 *For any pre-theory $(\mathfrak{f}, P)$, the pre-theory $(p\mathfrak{f}, P \circ p_{(\mathfrak{f},P)})$ is upward normal.*

Proof. It can be easily checked that for any I,

$$p\mathfrak{f}_{P \circ p_{(\mathfrak{f},P)}}(I) = \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I).$$

Given this result, the claim follows from the preceding lemma, because if $I \subseteq J \subseteq \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I)$, then we have $\mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I) = \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(J)$. □

We will in the sequel simply write equations of the above form. We will call this an **implicit definition** of a pre-theory, as opposed to the explicit definitions, which accord to the standard scheme. The important thing is that every implicit definition can be transformed into an explicit one, as we will prove later on.

The simple proof of theorem 46 gives us two further results: the first is that there is no problem regarding the question whether the resulting projections are increasing: an easy argument yields that $\mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(P)(I) \supseteq I$, otherwise we run into a contradiction. The other result is the following, which might be discouraging:

Corollary 48 *If $I \subseteq J \subseteq p\mathfrak{f}_{P \circ p_{(\mathfrak{f},P)}}(I)$, then $p\mathfrak{f}_{P \circ p_{(\mathfrak{f},P)}}(I) = p\mathfrak{f}_{P \circ p_{(\mathfrak{f},P)}}(J)$*

This is immediate, and it shows that p is very expectable. So in a sense, we might consider the normalizing map p to be too rigid. There is another bad thing about the map p: the map itself is not a fixed point map, that is, there are finite languages I such that $p(p(I)) \subsetneq p(I)$. For example, put $I := \{ab, aabb, aaabbb, aaaxbbbx, aaaxxbbbxx\}$. We then get $p(I) = \{ab, aabb, aaabbb, aaaxbbbx\}$, and $p(p(I)) = \{ab, aabb, aaaxbbbx\}$. So the finite language gets smaller, and from this example we can also learn that in general, $\mathfrak{f}_{Pr} \circ p \circ p(I) \not\supseteq I$. This poses no problem in general, but still it seems to be somewhat awkward if we think of the "relevant fragment" of a language as something being closed. There is an alternative, which we will call q, which actually is a fixed point and only slightly differs in definition:

Let $(\mathfrak{f}, P)$ be a pre-theory. Define the q-periphery $qper_{(\mathfrak{f},P)}(I) := \{M \subseteq I : \mathfrak{f}_P(I - M) \supseteq \mathfrak{f}_P(I)\}$, and $q_{(\mathfrak{f},P)}(I) = I - max_{rad^*}(qper_{(\mathfrak{f},P)}(I))$.

Despite the apparent similarity, the map q properly differs from p. For example, we have

$$q_{(\mathfrak{g},Pr)}(\{ab, aaabbb, aabb, aaaxbbbx\}) = \{ab, aabb, aaabbb, aaaxbbbx\},$$

despite the fact that $aaabbb$ is derivable from $\{ab, aabb, aaaxbbbx\}$, because otherwise we would lose an analogy and diminish the derived language; but of course, we have $p_{(\mathfrak{g},Pr)}(\{ab, aaabbb, aabb, aaaxbbbx\}) = \{ab, aabb, aaaxbbbx\}$.

It is easy to see that q is a fixed point: for assume that $q_{(\mathfrak{f},P)}(q_{(\mathfrak{f},P)}(I)) \subsetneq q(I)$. Then there exists a $J \subsetneq q(I)$ such that $\mathfrak{f}_P(J) \supseteq \mathfrak{f}_P(q(I)$, such that $Jrad^*q(I)$, contradicting the condition.

Does this give us an upward normal pre-theory? Things are now much less simple. Upward normality for p could be obtained in full generality for any pre-theory $(\mathfrak{f}, P)$. For q this will not work. The main reason seems to be that there is no analogue to lemma 45 and 46 for q; that is, if $I \subseteq J \subseteq \mathfrak{f}_P(I)$, we cannot say anything about the inclusion relation of $q_{(\mathfrak{f},P)}(I)$ and $q_{(\mathfrak{f},P)}(J)$. So we will not go for the general case, but rather for the particular case of $(\mathfrak{g}, Pr)$. We now define the pre-theory $(q_{(\mathfrak{g},Pr)}\mathfrak{g}, Pr \circ q_{(\mathfrak{g},Pr)})$ implicitly by the equation

$$q_{(\mathfrak{g},Pr)}\mathfrak{g}_{Pr\circ q_{(\mathfrak{g},Pr)}}(I) = \mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}(I) \tag{4.21}$$

What is crucial for obtaining normality for $(q_{(\mathfrak{g},Pr)}g, Pr\circ q_{(\mathfrak{g},Pr)})$ is a property of Pr, for which unfortunately I have not yet a proof:

Conjecture 49 *Assume $\mathfrak{g}_{Pr}(I) = L$. Then for every J such that $I \subseteq J \subseteq L$ there is a J' such that $J \subseteq J'$ and $\mathfrak{g}_{Pr}(J') \subseteq L$.*

Note that in case $\mathfrak{g}_{Pr}(I) = I$, we have $I = J = J'$. As I said, I do not yet have a proof; yet the odds seem to be not bad: we can always think of new strings we add in order to "spoil" analogies. These will possibly allow to make larger analogies, which then derive a smaller language. The problematic point is that these larger strings possibly function as additional premises for analogies. Now we get the following result:

Theorem 50 *If conjecture 48 is true, $(q_{(\mathfrak{g},Pr)}\mathfrak{g}_{Pr\circ q_{(\mathfrak{g},Pr)}})$ is upward normal.*

Proof. Assume $I \subseteq J \subseteq \mathfrak{g}_{Pr}(I)$. We have to consider two cases:

Case 1: $\mathfrak{g}_{Pr}(q(J)) \subseteq \mathfrak{g}_{Pr}(q(I))$. Then we have $q(J) \subseteq q(I)$. Put $\overline{L} := \mathfrak{g}_{Pr}(q(I)) - \mathfrak{g}_{Pr}(q(J))$.

If $\overline{L}$ is finite, we put $J^{\#} = J \cup \overline{L}$. Then we define $q(J')$ as a smallest language $J' \supseteq J^{\#}$ such that $\mathfrak{g}_{Pr}(J') \subseteq \mathfrak{g}_{Pr}(I)$. By the conjecture, this language exists, and thus upward normality obtains for this case.

Assume $\overline{L}$ is infinite. Then we put $J^{\#} = J \cup \{\vec{w}\}$, for some $\vec{w} \in \overline{L}$, and define J' as the rad^*-smallest $J' \supseteq J^{\#}$ such that $\mathfrak{g}_{Pr}(J') \subseteq \mathfrak{g}_{Pr}(I)$, which by the conjecture exists.

Now we have $q_{(\mathfrak{g},Pr)}(J') \subseteq q_{(\mathfrak{g},Pr)}(I)$. Put $\overline{L}' := \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(I)) - \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(J'))$; and construct $J^{\#'} := J' \cup \{\vec{w}'\}$ for some $\vec{w}' \in \overline{L}'$, and J'' as the rad^*-smallest $J'' \supseteq J^{\#'}$ such that $\mathfrak{g}_{Pr}(J'') \subseteq \mathfrak{g}_{Pr}(I)$. By this construction, we have $q(I) - q(J') \supsetneq q(I) - q((J'')$, that is, the difference between the two must properly diminish. So, by a finite iteration of this construction, we get $(..((J)')..)' \supseteq J$ such that $q_{(\mathfrak{g},Pr)}((..((J)')..)') = q_{(\mathfrak{g},Pr)}(I)$, as the difference was finite from the beginning. So for this case, upward normality follows.

Case 2: $\mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(I)) \subseteq \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(J))$.

Assume that $\mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(I)) \subseteq \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(J))$. We can prove the lemma by choosing a $J' \supseteq J$ such that $\mathfrak{g}_{Pr}(J') \subseteq \mathfrak{g}_{Pr}(I)$. which by our conjecture exists. Then we have $q_{(\mathfrak{g},Pr)}(J') \subseteq q_{(\mathfrak{g},Pr)}(I)$, and, by definition of q, we also

have $\mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(J')) \subseteq \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(I))$. This reduces the second case to the first case. □

Conversely, assume that conjecture is wrong. We show that under this assumption, $q\mathfrak{g}_{Pr}$ is not upward normal.

Lemma 51 *If conjecture 49 is wrong, then* $(q_{(\mathfrak{g},Pr)}g, Pr \circ q_{(\mathfrak{g},Pr)})$ *is not upward normal.*

Proof. By assumption, we have finite languages I, J such that $I \subseteq J \subseteq \mathfrak{g}_{Pr}(I)$, and there is no $J' \supseteq J$ such that $\mathfrak{g}_{Pr}(J') \subseteq \mathfrak{g}_{Pr}(I)$. That means that for all $J' \supseteq J$, we have $\mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(J')) \not\subseteq \mathfrak{g}_{Pr}(q_{(\mathfrak{g},Pr)}(I))$, and hence we have $\mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}(J') \not\subseteq \mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}(I)$. Therefore, upward normality fails. □

Of course, these last results do not refer to any particular properties of $(\mathfrak{g}, Pr)$, so they can be easily seen to obtain in much more generality.

4.6.4 Normality and a Normal Pre-Theory

We now introduce the notion of general normality:

Definition 52 *A pre-theory is* ***normal*** *if it is both downward and upward normal (in the strong sense).*

Theorem 53 $(\mathfrak{g}2, P2)$ *is normal.*

Proof. We have already shown that $(\mathfrak{g}2, P2)$ is downward normal. We now show it is upward normal.

Assume we have $I \subseteq J \subseteq \mathfrak{g}2_{P2}(I)$, for finite I, J. If $\mathfrak{g}2_{P2}(I)$ is finite, then $\mathfrak{g}2_{P2}(I) = I$, and the claim follows. So assume that $\mathfrak{g}2_{P2}(I)$ is infinite, and $J \neq I$. Then we know by monotonicity that $\mathfrak{g}2_{P2}(I) \subseteq \mathfrak{g}2_{P2}(J)$. Now assume that $\vec{w} \in J - I$. Then $\vec{w} \in \mathfrak{g}2_{P2}(I) - I$, and by lemma 33, $\vec{w}$ is not elementary. Therefore, we cannot use $\vec{w}$ neither as a premise for an inference, nor for any new analogy (the two are identical by definition of $(\mathfrak{g}2, P2)$).

From this, it follows that if $I \subseteq J \subseteq \mathfrak{g}2_{P2}(I)$, then $\mathfrak{g}2_{P2}(I) = \mathfrak{g}2_{P2}(J)$. □

4.6.5 Monotonicity

We will now consider the issue of monotonicity, which we have already mentioned before:

Definition 54 *An analogical map P is monotonic, if from $J \subseteq I$ it follows that $P(J) \subseteq P(I)$. A pre-theory $(\mathfrak{f}, P)$ is monotonic, if from $I \subseteq J$ follows that $\mathfrak{f}_P(I) \subseteq \mathfrak{f}_P(J)$.*

Note that if P is monotonic, it follows that if $J \subseteq I$, then $\mathfrak{f}_P(J) \subseteq \mathfrak{f}_P(I)$. The inverse however is not necessarily true: $(\mathfrak{f}, P)$ can be monotonic without P being monotonic – we could get less analogies, but all of them are inessential, in that they make no difference for the language derived. Regarding Pr, we can easily show the following:

Lemma 55 *Pr is not monotonic;* $(\mathfrak{g}, Pr)$ *is not monotonic.*

Proof. Take $I := \{a, bac\}$; $I' = I \cup \{dae\}$. This works as counterexample for both. □

The main point why Pr fails to be monotonic is that it is quite restrictive: the criteria are implicational, and in this sense they do not only refer to what has to be in the language, but also implicitly to what must not be in a certain finite language in order to allow for an analogy. The same holds for the simple $P1$:

Lemma 56 *$P1, (\mathfrak{g}, P1)$ are not monotonic.*

Proof. Recall that we have $\vec{w} \approx_I^{P1} \vec{v}$ if $\vec{w} \sqsubseteq \vec{v}$ and $\vec{w} \leq_I \vec{v}$. So just consider $I := \{a, bac\}$, where $P1(I) = \{(a, bac)\}$; and $I' = I \cup \{xbacy\}$, where $P1(I') = \{(bac, xbacy), (a, abacy)\}$. Consequently, we have $h \circ \mathfrak{g}_{P1}(I) = \{b^n ac^n : n \in \mathbb{N}\}$; $h \circ \mathfrak{g}_{P1}(I') = \{x^n bacy^n : n \in \mathbb{N}\} \cup \{(xb)^n a(cy)^n : n \in \mathbb{N}\}$. □

But what kind of analogical maps/pre-theories are monotonic, and how do we construct them? We will first look at analogical maps. As it turns out, there is an easy and reliable way to construct them. Given any analogical map P, we can use a **powerset construction** to immediately get a monotonic extension of P:

Definition 57 *The powerset extension $\wp(P)$ of an analogical map P is defined by $\wp(P)(I) := \{(\vec{x}, \vec{y}) : (\vec{x}, \vec{y}) \in P(J)$ for some $J \subseteq I\}$.*

If P is an analogical map, then so is $\wp(P)$, and moreover:

Lemma 58 *If P is an analogical map, then $\wp(P)$ is an analogical map with the same domain and range, and $\wp(P)$ is monotonic.*

The proof is an easy exercise. Now we can ask: is this construction a reasonable one, is it desirable from a (meta-)linguistic point of view? This question is of course hard to answer. Of course, monotonicity is desirable in some sense; as a matter of fact it solves one of the main problems we stated in the beginning: in view of the fact we can only observe fragments of the observable language, we are unsure about our "language", but with a monotonic analogical map, we can at least give a partial answer, in that we know: a statement of the form $\vec{w}$ IS PART OF OUR "LANGUAGE" will never be falsified by new evidence (still we remain unsure about statements of the form: $\vec{w}$ IS NOT PART OF OUR "LANGUAGE", see the discussion on negative evidence).

But whereas before, our problem was that an analogical map such as Pr is probably rather too restrictive, we might now think that we are too liberal. In particular, there is no evidence which might make a certain analogy illegitimate, which is to say for the linguist that there is no evidence which can make a certain projection implausible. I am not too sure whether this is desirable. But note that this is a fault which is intrinsic to *any* monotonic analogical map.

We will not settle the question on whether monotonicity is necessary or even desirable for pre-theories; this will remain, as many issues, a matter of taste. However, what we can show is that, given that the problem of monotonic pre-theories is that they are rather too liberal, the powerset construction is an optimal solution in a strong sense:

Definition 59 *Given two analogical maps P, P', say that P is smaller than P' (in symbols $P \leq P'$), if for all finite languages I, $P(I) \subseteq P'(I)$.*

Theorem 60 *Given an analogical map P, $\wp(P)$ is the smallest monotonic analogical map which is larger than P.*

Proof. Take an analogical map Q which is larger than P and monotonic, and an arbitrary finite language I. As Q is larger than P, we know that for every $J \subseteq I$, $Q(J) \supseteq P(J)$. Furthermore, as Q is monotonic, we know that for every $J \subseteq I$, $Q(J) \subseteq Q(I)$. Therefore, assume $\mathbf{a} \in \wp(P)(I)$. Then $\mathbf{a} \in P(J)$ for some $J \subseteq I$. Therefore, $\mathbf{a} \in Q(J)$; and by monotonicity, $\mathbf{a} \in Q(I)$. Therefore, for any finite language I, $Q(I) \supseteq \wp(P)(I)$. □

So the powerset construction is the *smallest monotonic extension* for any given analogical map. So if one thinks that a given analogical map is linguistically justified, and one thinks that monotonicity is necessary/desirable, then one just has to use the powerset construction. Given two analogical maps P, P', we denote **extensional equality** by $P \equiv P'$, by which we mean that for all finite I, we have $P(I) = P'(I)$.

Corollary 61 *Given a monotonic analogical map P, we have $\wp(P) \equiv P$.*

Actually, this follows immediately from the last lemma; for the sake of exposition, we will give another proof.

Proof. Assume that P is monotonic. Then for any finite language I, $J \subseteq I$, we have $P(J) \subseteq P(I)$. Therefore, we have $\bigcup_{J \subseteq I} P(J) \subseteq P(I)$. Conversely, as $I \subseteq I$, equality follows. □

Now that we have seen that there are very satisfying solutions for providing monotonic analogical maps, we will see whether this can be transferred to pre-theories. Obviously, for any pre-theory $(\mathfrak{f}, P)$, we can construct a monotonic pre-theory $(\mathfrak{f}, \wp(P))$. So for the projection we obtain the following equality:

$$\mathfrak{f}_{\wp(P)}(I) = \mathfrak{f}_{\wp(P)(I)}(I) = \mathfrak{f}_{\bigcup_{J \subseteq I} P(J)}(I) \tag{4.22}$$

But it turns out that $\mathfrak{f}_{\wp(P)}$ is *not* the smallest projection which is larger than $\mathfrak{f}_P$ and monotonic. We can show that it is possible to extend $\mathfrak{f}_P$ to a monotonic map using a powerset construction in at least one smaller way:

$$\wp(\mathfrak{f})_P(I) := \bigcup_{J \subseteq I} \mathfrak{f}_P(J) \tag{4.23}$$

This actually gives us an implicit definition of a pre-theory $(\wp(\mathfrak{f}), P)$. It is easy to see that this pre-theory is monotonic. We can also easily show that:

Lemma 62 *1. For all finite languages I, pre-theories P, $\bigcup_{J \subseteq I} \mathfrak{f}_P(J) \subseteq \mathfrak{f}_{\wp(P)(I)}(I)$. Furthermore,*
2. there exist finite languages I and pre-theories $(\mathfrak{f}, P)$ such that $\bigcup_{J \subseteq I} \mathfrak{f}_P(J) \subsetneq \mathfrak{f}_{\wp(P)(I)}(I)$.

Proof. 1. $\subseteq$: Assume that for some $J \subseteq I$, $\mathfrak{f}_P(J) = L$. As $P(J) \subseteq \wp(P)(I)$, and $J \subseteq I$, we have $L \subseteq \mathfrak{f}_{\wp(P)(I)}(I)$. As this holds for all $J \subseteq I$, the claim follows.

2. $\subsetneq$: Take $(\mathfrak{g}, Pr)$ and $I_1 = \{axbyc, ax_1xx_2byc, axby_1yy_2c\}$. Then we have $\bigcup_{J \subseteq I_1} \mathfrak{g}_{Pr}(J) = \{a(x_1)^n x(x_2)^n byc : n \in \mathbb{N}_0\} \cup \{axb(y_1)^n y(y_2)^n c : n \in \mathbb{N}_0\}$, whereas $\mathfrak{g}_{\wp(Pr)(I_1)}(I_1) = \{a(x_1)^n x(x_2)^n b(y_1)^m y(y_2)^m c : n, m \in \mathbb{N}_0\}$, which is clearly a superset. □

So which definition is preferable? As the problem with monotonicity is that it is pre-theories become rather too permissive on occasions, we would opt for the smaller. This is confirmed by the following result:

Theorem 63 *Given a pre-theory* $(\mathfrak{f}, P)$*, projection* $\mathfrak{f}_P$*, then* $\wp(\mathfrak{f})_P$ *is the smallest projection which is 1. monotonic and 2. larger than* $\mathfrak{f}_P$*.*

The proof is standard set-theoretic and almost identical to the one of theorem 59, and therefore omitted. So $(\wp(\mathfrak{f}), P)$ should be preferable to $(\mathfrak{f}, \wp(P))$. I have the impression though that $(\mathfrak{f}, \wp(P))$ is much more elegant in its definition and intuitive in its application.

We should at this point remark that there is an important property regarding the interaction of upward normality and monotonicity:

Lemma 64 *Let* $(\mathfrak{f}, P)$ *be a pre-theory which is upward normal and (weakly) monotonic. Then we have for all finite languages* I, J*, if* $I \subseteq J \subseteq \mathfrak{f}_P(I)$*, then* $\mathfrak{f}_P(I) = \mathfrak{f}_P(J)$*.*

Proof. Assume $I \subseteq J \subseteq \mathfrak{f}_P(I)$, and $\mathfrak{f}_P(I) \neq \mathfrak{f}_P(J)$. By (weak) monotonicity, we have $\mathfrak{f}_P(I) \subsetneq \mathfrak{f}_P(J)$. But also, for all $J' \supseteq J$, we have $\mathfrak{f}_P(J) \subseteq \mathfrak{f}_P(J')$. This contradicts upward normality. □

So we get an immediate corollary:

Corollary 65 *If* $I \subseteq J \subseteq \mathfrak{g}2_{P2}(I)$*, then* $\mathfrak{g}2_{P2}(I) = \mathfrak{g}2_{P2}(J)$*.*

So pre-theories being upward normal and monotonous are very predictable.

4.6.6 A Weaker Form of Monotonicity

As we have already said, the problem of monotonicity is that it is very strong and makes pre-theories very permissive. We now define a related, but somewhat weaker notion:

Definition 66 *A pre-theory* $(\mathfrak{f}, P)$ *is* ***weakly monotonic****, if from* $I \subseteq J \subseteq \mathfrak{f}_P(I)$*, it follows that* $\mathfrak{f}_P(I) \subseteq \mathfrak{f}_P(J)$*.*

Some remarks are in order: note that, contrary to the definition of monotonicity, this makes reference to the entire pre-theory; it does not seem to make sense for the analogical map. Apart from this, it is a restriction of monotonicity, in that we say that monotonicity with respect to a set I should only apply to strings in $\mathfrak{f}_P(I)$. It is clear that if $(\mathfrak{f}, P)$ is monotonic, then it is also weakly monotonic. Weak monotonicity means, that for a finite language I, only if we add derivable strings the resulting language must not decrease. That is in fact considerably weaker than monotonicity, and a requirement which only few will find too strong, or put differently, as making pre-theories too liberal. The condition shows an immediate relation to the "normalizing maps" p and q we introduced for upward normality. And in fact, we can show the following result:

Theorem 67 *Let* $(\mathfrak{f}, P)$ *be a pre-theory. Then the map* $\mathfrak{f}_P \circ p_{(\mathfrak{f},P)}$*, induced by the pre-theory* $(p(\mathfrak{f}), P \circ p_{(\mathfrak{f},P)})$ *is weakly monotonic.*

Proof. We have to show that if $I \subseteq J \subseteq \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I)$, then $\mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I) \subseteq \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(J)$. But in fact, we have already seen above (corollary 47), that in this case we have the stronger result that $\mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(I) = \mathfrak{f}_P \circ p_{(\mathfrak{f},P)}(J)$. So the claim follows. □

This is a positive result, but also a bit disappointing: as we said, we might want more richness in our pre-theories, we might want them to be less expectable. We therefore also introduced the normalizing map q. As it turns out, this map is more flexible, but fails to meet the requirements of weak monotonicity.

Lemma 68 *$\mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}$ is not not weakly monotonic.*

Proof. Take $I := \{ab, aabb, aaabbb, aaaxbbby\}$. We have $q_{(\mathfrak{g},Pr)}(I) = I$. Take a $J \subseteq \mathfrak{g}_{Pr}(I)$, where $J := I \cup \{aaaabbbb\}$. In J, we do not get $bbb \approx_J^{Pr} xbbby$; therefore $q_{(\mathfrak{g},Pr)}(J) = \{ab, aabb, aaaxbbby\}$. It is now easy to see that $\mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}(J) \subsetneq \mathfrak{g}_{Pr} \circ q_{(\mathfrak{g},Pr)}(I)$. □

This also shows that strong upward normality does not entail weak monotonicity. This is a negative result, but also has the positive effect of showing that q is more rich and flexible than p.

There is a last important result on the relation of upward normality and weak monotonicity. We see that upward normality does not entail weak monotonicity. Also the converse entailment does not obtain: we can easily think of a projection which is weakly monotonic, yet not upward normal - just think of any (trivial) pre-theory as $(\mathfrak{f}, P)$, where we have $I \subsetneq J \subseteq \mathfrak{f}_P(I) \Rightarrow \mathfrak{f}_P(I) \subsetneq \mathfrak{f}_P(J)$, and where $\mathfrak{f}_P$ is injective. But there is a very precise characterization for the class of pre-theories which are both weakly monotonic and upward normal, which strengthens the result of the last lemma in the section on monotonicity:

Theorem 69 *$\mathfrak{f}_P$ is weakly monotonic and upward normal, if and only if from $I \subseteq J \subseteq \mathfrak{f}_P(I)$ it follows that $\mathfrak{f}_P(I) = \mathfrak{f}_P(J)$.*

If. Assume that from $I \subseteq J \subseteq \mathfrak{f}_P(I)$ it follows that $\mathfrak{f}_P(I) = \mathfrak{f}_P(J)$. This clearly gives upward normality, because if $\mathfrak{f}_P(I)$ is infinite, every finite $J : I \subseteq J \subseteq \mathfrak{f}_P(I)$ induces $\mathfrak{f}_P(I)$. Also, it gives weak upwards normality, because $\mathfrak{f}_P(I) = \mathfrak{f}_P(J)$ entails $\mathfrak{f}_P(I) \subseteq \mathfrak{f}_P(J)$.

Only if: Assume we have $I \subseteq J \subseteq \mathfrak{f}_P(I)$, and $\mathfrak{f}_P(I) \neq \mathfrak{f}_P(J)$. By weak monotonicity, we know that $\mathfrak{f}_P(I) \subseteq \mathfrak{f}_P(J)$, and so, $\mathfrak{f}_P(I) \subsetneq \mathfrak{f}_P(J)$. But then it holds for all $J' : J \supseteq J' \subseteq \mathfrak{f}_P(J)$ that $\mathfrak{f}_P(I) \subsetneq \mathfrak{f}_P(J')$, by weak monotonicity, and *a fortiori*, the same holds for $J' : J \subseteq J' \subseteq \mathfrak{f}_P(I)$. But this contradicts strong upward normality. □

This is a very interesting result, because it shows that pre-theory which are both upward normal and weakly monotonic are quite predictable in their behavior.

4.6.7 Fixed-point Properties

We have not yet considered another important point. We say that a pre-theory P, projection $\mathfrak{f}_P$ is **closed**, if $\mathfrak{f}_P(I) = \mathfrak{f}_P(\mathfrak{f}_P(I))$. This is generally well-defined, because we have adopted the convention that for L infinite, we have $P(L) = \emptyset$, and therefore, $\mathfrak{f}_P(L) = L$. Note that nonetheless, this notion is not vacuous, because if $\mathfrak{f}_P(I)$ is finite, it might well be that $\mathfrak{f}_P(\mathfrak{f}_P(I)) \neq \mathfrak{f}_P(I)$. We consider it a favorable property for our pre-theories to be closed.

We can easily check that all the pre-theories we have considered so far are closed, because we have either $\mathfrak{f}_P(I) = I$, or $\mathfrak{f}_P(I)$ infinite, and this entails closure. This holds because our analogies are necessarily such that they derive infinite languages; so we have either no analogies at all, or we obtain infinite languages. So this property is even stronger than being closed. However, we will later on allow pre-theories and analogies which lack this property, as we allow analogies which do not necessarily derive infinite languages. In these cases, the matter of closure is serious, we have to take care of it. This will become a point when we consider what we call transformational pre-theories.

4.6.8 Closure under Morphisms

We now come to the question, whether the projections our pre-theories define are closed under various morphisms. This is a very important question, because we have been deliberately vague on whether we want to apply them directly on the words we observe, or on the categories we assign. To move from one conception to the other, we need some morphism, so closure under certain morphisms is quite desirable. For simplicity, we will now adopt the following convention:

Definition 70 *Assume* $(\mathfrak{f}, P)$ *is a pre-theory. Then by* $\mathfrak{C}(\mathfrak{f}, P)$ *we denote the class of all languages* L *such that there is a finite language* I *where* $\mathfrak{f}_P(I) = L$. *By* $\mathfrak{C}^\infty(\mathfrak{f}, P)$ *we denote the class of all infinite languages* L *such that there is a finite language* I *where* $\mathfrak{f}_P(I) = L$.

We get the following result:

Lemma 71 $\mathfrak{C}(\mathfrak{g}, Pr)$,$\mathfrak{C}(\mathfrak{g}, P1)$ *are not closed under (*ϵ*-free) homomorphism.*

Proof. $\mathfrak{C}(\mathfrak{g}, Pr)$ is clear. For $\mathfrak{C}(\mathfrak{g}, P1)$ take the language $I = \{aa, ccc\}$, and $h(a) = h(c) = a$. This can be easily transferred to infinite languages by adding an alphabetically disjoint language as above. □

Note that the fact that a class $\mathfrak{C}(\mathfrak{f}, P)$ is not closed under homomorphism entails that in general, $\mathfrak{f}_P(h(I)) \neq h(\mathfrak{f}_P(I))$. Conversely, the implication is not true: if $\mathfrak{C}(\mathfrak{f}, P)$ is closed under homomorphism, then we do not necessarily have $\mathfrak{f}_P(h(I)) = h(\mathfrak{f}_P(I))$. What is obviously true by contraposition is that if $\mathfrak{f}_P(h(i)) = h(\mathfrak{f}_P(I))$, then $\mathfrak{C}(\mathfrak{f}, P)$ is closed under homomorphism. We urge the reader to keep this in mind, as we will only present the strongest results we can obtain.

What we should ask now is: do we even get closure under isomorphism? This is usually taken for granted, as it is one of the most basic properties of families of languages (usually, one defines a family of languages by closure under isomorphism).

Lemma 72 *For a letter isomorphism* i, *we have* $\mathfrak{g}_{Pr}(i(I)) = i(\mathfrak{g}_{Pr}(I))$, $\mathfrak{g}_{P1}(i(I)) = i(\mathfrak{g}_{P1}(I))$ *and* $\mathfrak{g}2_{P2}(i(I) = i(\mathfrak{g}2_{P2}(I))$.

The proof consists in a straightforward checking of the definitions, which is quite tedious and therefore omitted. A more interesting question is whether our pre-theories are closed under more liberal bijections, namely so-called **codes** (see [46] for reference).

A code over Σ^* is a set $X \subseteq \Sigma^*$, such that for all $\vec{x}_1\vec{x}_2...\vec{x}_i : \vec{x}_1, ..., \vec{x}_i \in X$, $\vec{y}_1...\vec{y}_j : \vec{y}_1, ..., \vec{y}_j \in X$, it holds that if $\vec{x}_1...\vec{x}_i = \vec{y}_1...\vec{y}_j$, then $i = j$ and for

each $1 \leq k \leq i$, $\vec{x}_k = \vec{y}_k$. The closure of X under concatenation with itself is denoted by X^*. One therefore can say that every string in X^* has a unique X-factorization, that is, a unique decomposition into factors in X. An alternative and equivalent definition is the following: a code is a set $X \subseteq \Sigma^*$ such that for every alphabet $T : |T| = |X|$, every bijection $\phi : T \to X$, the homomorphic extension of $\phi : T^* \to X^*$ to strings is a bijection. So ϕ maps simple letters to strings; but if X is a code, the mapping will be one-to-one. We will refer to the homomorphic extension of a map ϕ as a **coding**.

Lemma 73 *1.* $\mathfrak{C}(\mathfrak{g}, Pr)$ *is not closed under coding. 2.* $\mathfrak{C}(\mathfrak{g}2, P2)$ *is not closed under coding.*

Proof. 1. Take the language $I = \{a, bab\}$, and the coding $\phi(a) = a, \phi(b) = bab$; see above. 2. Take an arbitrary alphabet Σ, an arbitrary language I having an infinite image, and define $\phi(\sigma) = \sigma^5$ for all $\sigma \in \Sigma$. $\phi(I)$ does not contain elementary strings. □

Lemma 74 *There is a coding* ϕ *such that* $\phi(\mathfrak{g}_{P1}(I)) \neq \mathfrak{g}_{P1}(\phi(I))$.

Proof. Put $I = \{a, ab\}$, $\phi(a) = bab, \phi(b) = aba$. $\phi(I) = \{bab, bababa\}$. We then have $P1(\phi(I)) = (bab, bababa)$, and we can derive $ba(bababa)a$, which is not in $\phi(\mathfrak{g}_{P1}(I))$. □

To improve this apparently bad situation we can do the following: we define a very restricted notion of a code, namely so-called *infix*-code (as analogical to the well-known prefix-codes, see [46]). This is a very strict notion, and in fact it is more restricted than any notion of code which is known to me from the literature. Unfortunately, we cannot obtain the following results with any weaker notion.

Definition 75 *An* ***infix code*** *is a code* $X \subseteq \Sigma^*$ *such that from* $\vec{w}\vec{x}\vec{v} \in X^*$, $\vec{x} \in X$, *it follows that for all* $\vec{x}_1, ..., \vec{x}_i \in X$ *such that* $\vec{x}_1...\vec{x}_i = \vec{w}x\vec{v}$, *we have a* j *such that* $\vec{x}_1...\vec{x}_j = \vec{w}, \vec{x}_{j+2}...\vec{x}_i = \vec{v}$.

So we can recognize any infix in the code, and "translate" it accordingly, without having to consider any of its context. An example of an infix code is any code, where beginning and ending of strings in X is uniquely marked. For example, we can encode an alphabet T with $|T| = n$ in $\{0, 1\}^*$ with the infix code $\{00, 010, 0110, ..., 01^{n-1}0\}$.

Lemma 76 *Let* ϕ *be an infix coding. Then for all finite languages* I, *we have 1.* $\mathfrak{g}_{Pr}(i(I)) = i(\mathfrak{g}_{Pr}(I))$. *2.* $\mathfrak{g}_{Pr}(i(I)) = i(\mathfrak{g}_{Pr}(I))$.

Proof. We can easily reduce this case to the letter isomorphism case, because we know that $\vec{w}(i(\vec{x}))\vec{v} \in i[I]$ if and only if $i^{-1}(\vec{w})\vec{x}i^{-1}(\vec{v}) \in I$. □

Note that this lemma does *not* obtain for $(\mathfrak{g}2, P2)$!

Lemma 77 $\mathfrak{C}^\infty(\mathfrak{g}2, P2)$ *is not closed under infix coding.*

Proof. See the example above, where $\phi(\sigma) = \sigma^5$. This is an infix code. □

4.7 Methodological Universals

4.7.1 Which Languages Do We (Not) Obtain?

All structural pre-theories we considered so far yield context-free languages, so we have an upper bound for the class of languages we induce. However, we do not get all context-free languages, as can be easily deduced from the fact that 1. all finite languages are context-free, 2. we have finite languages which are projected, so not induced by themselves, and 3. all induced languages are infinite. This tells us that as a lower bound for the languages we obtain, we cannot consider a class containing the finite languages.

But this result is not only very unspecific, it is also in some sense trivial, as we only are interested in infinite languages (as candidates for "language"), so the fact that we do not obtain certain finite languages is of no concern to us. What should be a concern to us are the infinite languages which *are* context-free, yet not induced by any finite language and some pre-theory under consideration. We will first try to bring some order in the relation of languages induced by $(\mathfrak{g}, Pr), (\mathfrak{g}, P1), (\mathfrak{g}2, P2)$, and the normalized pre-theories. Then we show some interesting examples of languages we cannot obtain by them. This will then also shed a better light on the properties of languages we *can* obtain. We will restrict our attention mostly to $\mathfrak{C}^\infty(\mathfrak{f}, P)$, because firstly the finite languages are the ones which remain invariant under our pre-theories, and secondly they do not have any relevance for us.

Lemma 78 *We have* $\mathfrak{C}^\infty(\mathfrak{g}, Pr) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}2, P2)$ *and* $\mathfrak{C}^\infty(\mathfrak{g}2, P2) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, Pr)$ *and*

Proof. $\mathfrak{C}^\infty(\mathfrak{g}, Pr) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}2, P2)$ Take the language $\{a^n b^n : n \geq 4\}$. This is in $\mathfrak{C}^\infty(\mathfrak{g}, Pr)$ but not in $\mathfrak{C}^\infty(\mathfrak{g}2, P2)$ because of the elementary string condition.

$\mathfrak{C}^\infty(\mathfrak{g}2, P2) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, Pr)$ Conversely, take the language $\{aaabbb, aaaabbbb\} \cup L$, where L is any infinite language in $\mathfrak{C}^\infty(\mathfrak{g}2, P2)$ over an alphabet Σ such that $a, b \notin \Sigma$. □

So we have a relation of incomparability. The second part of the proof shows that results of this kind are however not very meaningful, because we can always recur to finite, alphabetically distinct sublanguages. So the case of the finite languages falls back on us, and we have to be aware that inclusions are only meaningful if they do not use this sort of argument. The reason why arguments of this kind will always work with our pre-theories is the following general property of our pre-theories so far, which we will have it for all pre-theories we look at.

Definition 79 *A pre-theory* $(\mathfrak{f}, P)$ *is* ***alphabetically innocent****, if for* $I \subseteq \Sigma^*$, $J \subseteq T^*$, $\Sigma \cap T = \emptyset$, $\mathfrak{f}_P(I \cup J) = \mathfrak{f}_P(I) \cup \mathfrak{f}_P(J)$.

Lemma 80 $(\mathfrak{g}, Pr), (\mathfrak{g}, P1), (\mathfrak{g}2, P2)$ *are alphabetically innocent.*

This is immediate to see. So in a word, all our pre-theories are alphabetically innocent.

Lemma 81 *We have* $\mathfrak{C}^\infty(p\mathfrak{g}, Pr \circ p_{(\mathfrak{g}, Pr)}) \subsetneq \mathfrak{C}^\infty(\mathfrak{g}, Pr)$.

Proof. $\subseteq$ Assume we have $L = p\mathfrak{g}_{Pr \circ p_{(\mathfrak{g}, Pr)}}(I)$. Then we put $I' = p_{(\mathfrak{g}, Pr)}(I') = I$, and then $\mathfrak{g}_{Pr}(I') = L$.

$\subsetneq$ Take a language as $I = \{ab, aabb, aaaabbbb\}$, such that $\mathfrak{g}_{Pr}(I) = \{ab, (aa)^n(bb)^n : n \in \mathbb{N}\} := L$. Obviously, we have $p_{(\mathfrak{g},Pr)}(I) = \{ab, aabb\}$. Furthermore, for any finite I' such that $\{ab, aabb\} \subseteq I' \subseteq L$, we will have $p_{(\mathfrak{g},Pr)}(I') = \{ab, aabb\}$; so we cannot induce L. □

So the normalizing map p comes with a decrease in "inducing power". The same result can be obtained if we substitute Pr with $P1$. The results regarding $P1$ are the following:

Lemma 82 *We have* $\mathfrak{C}^\infty(\mathfrak{g}, P1) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}2, P2)$ *and* $\mathfrak{C}^\infty(\mathfrak{g}2, P2) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, P1)$.

Proof. See the proof of the corresponding lemma for Pr. □

Lemma 83 $\mathfrak{C}^\infty(\mathfrak{g}, Pr) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, P1)$, $\mathfrak{C}^\infty(\mathfrak{g}, P1) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, Pr)$.

Proof. $\mathfrak{C}^\infty(\mathfrak{g}, P1) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, Pr)$: Take $L = \{((bab)^n a(bab)^n : n \in \mathbb{N}\}$. We have $L = \mathfrak{g}_{P1}(\{a, bababab\})$. But assume we have I' such that $\mathfrak{g}_{Pr}(I') = L$. We need $a \in I'$, consequently $bababab \in I'$. But then we also need $bbabababbabab \in I'$ etc., so I' is infinite.

$\mathfrak{C}^\infty(\mathfrak{g}, Pr) \not\subseteq \mathfrak{C}^\infty(\mathfrak{g}, P1)$. Put $I = \{ab, aabb, xaby\} \cup L$, where L is an infinite language in $\mathfrak{C}^\infty(\mathfrak{g}, Pr)$ over Σ such that $a, b, x, y \notin \Sigma$ (again, we see that this part of the lemma is quite meaningless, whereas the former is not). □

Why should we be interested in the languages we do not induce, or, more generally, why should we be interested in the classes we induce, given they are very unnatural from the point of view of formal language theory? In my view, there is a very strong and good motivation for scrutinizing their properties, even though to the "normal linguist" this motivation will seem a bit queer at the first sight. They provide a first example of what we might call **methodological universals**. These are universal properties of "language", which are artefacts of our projection. So assume we say that $(\mathfrak{g}, Pr)$ is the right pre-theory to adopt, we formalize our linguistic observations and perform the projection under this assumption. Then we might observe some universal properties of "language" (recall that, after all, "language" is the proper subject of linguistics!). The most obvious one is: "languages" are context-free. In addition, if we work with strong "languages", we will say that we only find phrase-structure style dependencies. But these, obviously, are not properties of the observed languages; these are properties due to our methodology, which will obtain no matter what we observe.

Formally, a methodological universal of a pre-theory $(\mathfrak{f}, P)$ is a property of the class of languages which are induced by some finite language under $(\mathfrak{f}, P)$, that is, a property of $\mathfrak{C}(\mathfrak{f}, P)$. So it is important to know the methodological universals of pre-theories we consider, for two main reasons: for the metalinguist these are interesting in itself, as he can decide whether they make a pre-theory preferable or not. For example, he might opt in favor of context-free or mildly context-sensitive "languages", or in favor of phrase-structure style dependencies. For the normal linguist who simply applies a pre-theory it is also very important: he has to know its methodological universals in order to exclude them from the "linguistic" observations he makes, that is, his empirical observations. For example, if he notices that all the "languages" he considers have a certain property $\mathcal{P}$, he should make sure that it is *not* a methodological universal of his pre-theory – because otherwise his observation is void of content. If on the other side $\mathcal{P}$ is not a methodological universal of his pre-theory, then he can make the claim that he has made a meaningful, empirical observation (still taken

"modulo the pre-theory"; we will work out what that means in the next section). As I have tried to point out in the second chapter, the concern that we take properties of pre-theories to be properties of languages is quite realistic.

So we have already presented some methodological universals regarding our pre-theories; we will now go a bit more into detail. Obviously, by the fact that there are finite languages which our pre-theories cannot induce, it is quite easy to construct infinite languages which cannot be induced either: just take a finite language I which cannot be induced, an infinite language L which can be induced, such that $I \subseteq \Sigma^*$, $L \subseteq T^*$, and $\Sigma \cap T = \emptyset$, and put $L' = I \cup L$. We have applied this argument repeatedly, which is based on our requirement of "alphabetical conservativity": our pre-theories must not introduce new letters, and on the even stricter requirement of alphabetical innocence, namely that sublanguages which do not share some letters do not interact in any way, which our pre-theories satisfy.

Another point to note is the following: take the language $L_7 := \{a^n a a^n : n \in \mathbb{N}_0\}$. Is this language induced by some finite language under $(\mathfrak{g}, Pr)$? The answer is negative, the reason is as follows: if we have $I_7 := \{a, aaa\}$, then we have the same a in a new, non-recursive context, because we cannot distinguish the a in the context (a, a) from the ones in context (ϵ, aa), and thus violate the weak Pr-condition. So take $I_7' := \{a, aaa, aaaaa\}$. Also the analogy $(aaa, aaaaa)$ is prevented for the same reason, and so on, and so, $L_7 \notin \mathfrak{C}(\mathfrak{g}, Pr)$. How about $L_8 := \{a^n a b^n : n \in \mathbb{N}_0\}$? This is in $\mathfrak{C}(\mathfrak{g}, Pr)$, but is only obtained by using larger analogies, as in the language $I_8 = \{a, aab, aaabb\}$, where we have $aab \approx_{I_8}^{Pr} aaabb$.

4.7.2 Unreasonable Restrictions of the String Case

We now come to a final characteristic of the classes of languages induced by our pre-theories, which in fact is an unreasonable restriction and will directly lead to the first major extension of our linguistic universe. Assume there is a finite language I, where we have $\vec{y} \leq_I \vec{x}$, as well as $\vec{x} \approx_I^{Pr} \vec{x}_1\vec{x}\vec{x}_2$. From this it does of course *not* follow that $\vec{y} \approx_I^{Pr} \vec{x}_1\vec{y}\vec{x}_2$; in fact, this *only* follows in a very particular case, which almost amounts to $\vec{x} \sim_I \vec{y}$ (though not exactly, $\vec{x}$ and $\vec{y}$ might have distinct contexts, as long as they are all recursive).

Consider $I = \{wxv, wx_1xx_2v, wyv, w_1yx_2v, yz\}$. In this setting, we have $x \approx_I^{Pr} x_1xx_2$, but $y \not\approx_I^{Pr} x_1yx_2$. This means in particular that the relation $\leq_I$ is not preserved over projection, not even for the elements of Σ^* in the *strong* language. This is a problem to our intuition. That it is not preserved for the weak language should not bother us, as it is undecidable anyway. But obviously, for the strong language the relation $\leq_L \subseteq \Sigma^* \times \Sigma^*$, not containing any brackets, is decidable, because it remains finite. For this reason, the fact $\leq_{\mathfrak{g}_{Pr}(I)}$ does not extend $\leq_I$ should bother us, because intuitively, we know that y has a more liberal distribution in I than x, and so it should have in $\mathfrak{g}_{Pr}(I)$ for its free occurrences.

This is not a problem of the more liberal pre-theories we considered before Pr (as the simple pre-theory $P1$); it is a problem of the restrictive Pr-family. Now the question is: can we be somewhat more liberal, yet not as liberal as $P1$? We will answer this question positively in the sequel.

The last problem is a consequence and particular instance of a more general problem of the pre-theories defined on sets of strings. We can only speak of strings, not about strings in a *certain distribution*. For example, in I as defined

above, we might say that the $\vec{y}$ in the word $\vec{y}\vec{z}$ "means" something entirely different from the $\vec{y}$ in the position where also $\vec{x}$ can occur (as far as we can say something like this in a purely syntactic approach; linguistically speaking, we would say it belongs to different categories). If we were able to say something like this, then we would get rid of our problem: we would consider the *set* of strings $\{x, y\}$, with respect to the contexts in which *both* can occur. This would also solve some more general problems. Consider for example the language:

$$J := \{a, ab, abb, abbb\} \cup \{d, db, dbb, dbbb\} \cup \{ac, de\} \tag{4.24}$$

In this language, there is no pseudo-recursion; but clearly, one would say that there *is* a pseudo-recursive pattern in there, because it is only the strings $\{ac, de\}$ which spoil the pseudo-recursion. But as we said, for us there is no way to speak of strings in certain positions. The extension we will introduce later on will allow us to do so in a certain way, which is still based on purely language-theoretic notions.

4.7.3 Linguistic Reason

Here we will consider the inverse question to the question we asked above. The above question was: which of the properties that we ascribe to "language", are necessary, that is, methodological universals? Here we are interested in the question: on the basis of our epistemological concerns, which empirical claims can we make about "language"? As a first point, if we want to claim that "language" has property $\mathcal{P}$ independently of any pre-theory, we must be careful that there exists a finite language I such that I does not have P. In this case, we say that $\mathcal{P}$ is finitary. Every property which is not finitary and which we ascribe to "language" depends on a projection. Contrary to what people tend to think, there are many interesting properties which satisfy this constraint. We will discuss this at length in the section on linguistic finitism, so there is no need to duplicate this discussion at this point.

What I want to point our here is the following: there are certain empirical claims we can make on *infinite languages*, which are based on the observation that for certain datasets I, $\mathfrak{f}_P(I)$ always shows a certain property $\mathcal{P}$, even though $\mathcal{P}$ is *not* a methodological universal of $(\mathfrak{f}, P)$. As an example, let us reconsider the family of pre-theories $Pr\text{-}k$ we considered above. We have claimed that if we choose k large enough, then $\mathfrak{g}_{Pr\text{-}k}(I)$ will be regular for *any* dataset I corresponding to a natural language dataset. This is a property in the above sense, but only if we consider a certain pre-theory. So there could be a dataset J such that $\mathfrak{g}_{Pr\text{-}k}(J)$ is not regular, but empirically, we do not find any. We say in this case that "language" has the property of being regular *modulo* $(\mathfrak{g}, Pr\text{-}k)$. In general, we can say that "languages" have property $\mathcal{P}$ modulo $(\mathfrak{f}, P)$, if there is no dataset I corresponding to an observed language, such that $\mathfrak{f}_P(I)$ does not have $\mathcal{P}$, and if there is some finite language J such that $\mathfrak{f}_P(J)$ does not have property $\mathcal{P}$.

Take another example: assume there is a pre-theory $(\mathfrak{f}, P)$ such that $\mathfrak{C}(\mathfrak{f}, P) \not\subseteq CFL$. Assume then we do not observe any linguistic dataset corresponding to a finite language I such that $\mathfrak{f}_P(I)$ is not context-free. Then we can claim that natural languages are context free modulo $(\mathfrak{f}, P)$. Or to reverse the example: surely, we can make observations to the point that under the pre-theory $(\mathfrak{g}, P1)$,

natural languages are *not regular*. Also this (negative) property is an instance of a property modulo $(\mathfrak{g}, P1)$, as there are finite languages I such that $\mathfrak{g}_{P1}(I)$ is not regular.

As is easy to see, the notion of a property modulo a pre-theory is fundamental for linguistics, and its importance can hardly be overrated. I think in a more formally rigid foundation of linguistics, in most cases it has to take the place of blunt statements of the form: natural languages have property $\mathcal{P}$, because these statements can only obtain if $\mathcal{P}$ is finitary.

Note however that the concept of $\mathcal{P}$ modulo $(\mathfrak{f}, P)$ does not allow us to use just any property: it must be a property which remains falsifiable under the assumption of $(\mathfrak{f}, P)$. This concerns in particular "universal properties" as: natural language rules are structure dependent (whatever that is supposed to mean exactly) under assumption of the structural inference $\mathfrak{g}$.

This small subsection could be said to be the germ of a critique of linguistic reason, that is, an investigation on which properties of language we can see, and which ones we cannot. But in order to achieve something which could carry this name, one would have to do much more work than I am currently able to.

4.8 Extension I: Pre-Theories on Powersets

The pre-theories presented so far have some drawbacks. In particular, the Pr-family seems to be somewhat too restrictive. We have so far considered analogical maps as maps $P : \Sigma^* \to \Sigma^* \times \Sigma^*$, and we have called these maps simple string based. We will now consider a new class. Unfortunately, the distinction by range and domain, substitution, structure are really orthogonal to each other, so it seems unavoidable to me that there is something arbitrary in the grouping; also, because we do not scrutinize all possible pre-theories, but only those which are of some mathematical and linguistic interest.

What we will now consider are analogical maps $P : \wp(\Sigma^*) \to \wp(\Sigma^* \times \Sigma^*)$, and accordingly modified inference rules. The main idea in this approach is the following: as we said, we have been vague on whether we intend our simple string-based pre-theories to work on words we observe or categories we already posit. Possibly, people will be more sympathetic to the latter case. But then the question is: who makes the categorization, and according to which criteria? This problem is addressed in this section, as syntactic concepts can be seen as categories, which are defined by the intrinsic distributional structure of a language. So the syntactic concept lattice provides us with a sort of "automatic" pre-categorization, which is then the object of projection, rather than the language itself. We will later on define under which conditions such a categorization can be said to be meaningful. We have to go through some definitions before we can present the next full pre-theory.

4.8.1 Syntactic Concepts

Syntactic concept lattices arise from the distributional structure of languages. Their main advantage is that they can be constructed on distributional relations which are weaker than strict equivalence. [6] has shown how these lattices can be enriched with a monoid structure to form residuated lattices. Syntactic concept lattices originally arose in the structuralist approach to syntax, back

when syntacticians tried to capture syntactic structures purely in terms of distributions of strings (see, e.g. [24]). An obvious way to do so is by partitioning strings/substrings into *equivalence classes*: we say that two strings $\vec{w}, \vec{v}$ are equivalent in a language $L \subseteq \Sigma^*$, in symbols

$$\vec{w} \sim_L^0 \vec{v} \text{ iff for all } \vec{x} \in \Sigma^*,\ \vec{w}\vec{x} \in L \Leftrightarrow \vec{v}\vec{x} \in L. \tag{4.25}$$

This defines the well-known Nerode-equivalence. We can use a richer equivalence relation, by considering not only left contexts, but also right contexts:

$$\vec{w} \sim_L^1 \vec{v} \text{ iff for all } \vec{x}, \vec{y} \in \Sigma^*,\ \vec{x}\vec{w}\vec{y} \in L \Leftrightarrow \vec{x}\vec{v}\vec{y} \in L. \tag{4.26}$$

Of course, this can be arbitrarily iterated for tuples of strings. The problem with equivalence classes is that they are too restrictive for many purposes: if we want to induce our grammar on the basis of a given dataset; then it is quite improbable that we get the equivalence classes we would usually desire as linguists, as we have pointed out in the beginning. But even apart from the intrinsic restrictions of finiteness there are notorious problems: there might be constructions, collocations, idioms which ruin equivalences which we would intuitively consider to be adequate.

Syntactic concepts provide a somewhat less rigid notion of equivalence, which can be conceived of as equivalence restricted to a given set of contexts. This at least partly resolves the difficulties we have described above.

4.8.2 Syntactic Concepts: Definitions

For a general introduction to lattices, see [12]; for background on residuated lattices, see [19]. Syntactic concept lattices form a particular case of what is well-known as formal concept lattice (or formal concept analysis) in computer science. In linguistics, they have been introduced in [64]. They were brought back to attention and enriched with residuation in [6], [7], as they turn out to be useful representations for language learning. In this section, we follow the presentation given in [6].

Given a language $L \subseteq \Sigma^*$, we define two maps: a map $\triangleright : \wp(\Sigma^*) \to \wp(\Sigma^* \times \Sigma^*)$, and $\triangleleft : \wp(\Sigma^* \times \Sigma^*) \to \wp(\Sigma^*)$, which are defined as follows:

$$\text{for } M \subseteq \Sigma^*,\ M^{\triangleright} := \{(\vec{x}, \vec{y}) : \forall \vec{w} \in M, \vec{x}\vec{w}\vec{y} \in L\}; \tag{4.27}$$

and dually

$$\text{for } C \subseteq \Sigma^* \times \Sigma^*,\ C^{\triangleleft} := \{\vec{x} : \forall (\vec{v}, \vec{w}) \in C, \vec{v}\vec{x}\vec{w} \in L\}. \tag{4.28}$$

That is, a set of strings is mapped to the set of contexts, in which all of its elements can occur. The dual function maps a set of contexts to the set of strings, which can occur in all of them. Obviously, $\triangleleft$ and $\triangleright$ are only defined with respect to a given language L, otherwise they are meaningless. As long as it is clear of which language (if any concrete language) we are speaking, we will omit however any reference to it. For a set of contexts C, $C^{\triangleleft}$ can be thought of as an equivalence class with respect to the contexts in C; but not in general: there might be elements in $C^{\triangleleft}$ which can occur in a context $(\vec{v}, \vec{w}) \notin C$ (and conversely).

The two compositions of the maps, $\triangleleft\triangleright$ and $\triangleright\triangleleft$, form a closure operator on subsets of $\Sigma^* \times \Sigma^*$ and Σ^*, respectively, that is:

1. $M \subseteq M^{\triangleright\triangleleft}$,
2. $M^{\triangleright\triangleleft} = M^{\triangleright\triangleleft\triangleright\triangleleft}$,
3. $M \subseteq N \Rightarrow M^{\triangleright\triangleleft} \subseteq N^{\triangleright\triangleleft}$,

for $M, N \subseteq \Sigma^*$. The same holds for contexts, where we simply exchange the order of the mappings, and use subsets of $\Sigma^* \times \Sigma^*$. We say a set M is **closed**, if $M^{\triangleright\triangleleft} = M$. The closure operator $\triangleright\triangleleft$ gives rise to a lattice $\mathcal{L}_S := \langle \mathfrak{B}_S, \leq \rangle$, where the elements of $\mathfrak{B}_S$ are the closed sets, and $\leq$ is interpreted as $\subseteq$. The same can be done with the set of closed contexts. Given these two lattices, $\triangleright$ and $\triangleleft$ make up a Galois connection between the two:

1. $M \leq N \Leftrightarrow M^{\triangleleft} \geq N^{\triangleleft}$, and
2. $C \leq D \Leftrightarrow C^{\triangleright} \geq D^{\triangleright}$.

Furthermore, for $\mathcal{L}_S$ the lattice of closed subsets of strings, $\mathcal{L}_C$ the lattice of contexts, it is easy to show that $\mathcal{L}_S \cong \mathcal{L}_C^{\partial}$, where by $[-]^{\partial}$ we denote the **dual** of a lattice, that is, the same lattice with its order relation inverted; and by $\cong$ we denote that there is an isomorphism between two structures. Therefore, any statement on the one lattice is by duality a statement on the other. Consequently, we can directly conceive of the two as a single lattice, whose elements are **syntactic concepts**:

Definition 84 *A syntactic concept A is an (ordered) pair, consisting of a closed set of strings, and a closed set of contexts, written $A = \langle S, C \rangle$, such that $S^{\triangleright} = C$ and $C^{\triangleleft} = S$.*

Note also that for any set of strings S and contexts C, $S^{\triangleright} = S^{\triangleright\triangleleft\triangleright}$ and $C^{\triangleleft} = C^{\triangleleft\triangleright\triangleleft}$. Therefore, any set M of strings gives rise to a concept $\langle M^{\triangleright\triangleleft}, M^{\triangleright} \rangle$, and any set of C contexts to a concept $\langle C^{\triangleleft}, C^{\triangleleft\triangleright} \rangle$. Therefore, we denote the concept which is induced by a set M, regardless of whether it is a set of strings or contexts, by $\mathcal{C}(M)$. We speak of the *extent* of a concept A as the set of strings it contains, which we denote by S_A; the *intent* of A is the set of contexts it contains, denoted by C_A. For example, given a language L, we have $S_{\mathcal{C}((\epsilon,\epsilon))} = L$, as all and only the strings in L can occur in L in the context (ϵ, ϵ).

We define the partial order $\leq$ on concepts by

$$\langle S_1, C_1 \rangle \leq \langle S_2, C_2 \rangle \qquad \Longleftrightarrow \qquad S_1 \subseteq S_2; \tag{4.29}$$

this gives rise to the syntactic concept lattice $\mathcal{L}$:

Definition 85 *The lattice of concepts of a language L, $SCL(L) = \langle \mathfrak{B}, \wedge, \vee \rangle$, with the partial order $\subseteq$, is called the **syntactic concept lattice**, where $\top = \mathcal{C}(\Sigma^*)$, $\bot = \mathcal{C}(\Sigma^* \times \Sigma^*)$, and for $\langle S_i, C_i \rangle, \langle S_j, C_j \rangle \in \mathfrak{B}$, $\langle S_i, C_i \rangle \wedge \langle S_j, C_j \rangle = \langle S_i \cap S_j, (C_i \cup C_j)^{\triangleleft\triangleright} \rangle$, and $\vee$ as $\langle (S_i \cup S_j)^{\triangleright\triangleleft}, C_i \cap C_j \rangle$.*

It is easy to verify that this forms a complete lattice. Note the close connection between intersection of stringsets and union of context sets, and vice versa. For a given language, we obviously have $\mathcal{L} \cong \mathcal{L}_S$, which we defined before.

4.8.3 Monoid Structure and Residuation

As we have seen, the set of concepts of a language forms a lattice. In addition, we can also give it the structure of a monoid: for concepts $\langle S_1, C_1\rangle, \langle S_2, C_2\rangle$, we define:

$$\langle S_1, C_1\rangle \circ \langle S_2, C_2\rangle = \langle (S_1S_2)^{\triangleright\triangleleft}, (S_1S_2)^{\triangleright}\rangle, \tag{4.30}$$

where $S_1S_2 = \{\vec{x}\vec{y} : \vec{x} \in S_1, \vec{y} \in S_2\}$. Obviously, the result is a concept. $'\circ'$ is associative on concepts:

$$\text{for } X, Y, Z \in \mathfrak{B},\ X \circ (Y \circ Z) = (X \circ Y) \circ Z. \tag{4.31}$$

This follows from the fact that $[-]^{\triangleright\triangleleft}$ is a *nucleus*,[4] that is, it is a closure operator and in addition it satisfies

$$S^{\triangleright\triangleleft}T^{\triangleright\triangleleft} \subseteq (ST)^{\triangleright\triangleleft}. \tag{4.32}$$

Using this property and the associativity of string concatenation, the result easily follows. Furthermore, it is easy to see that the neutral element of the monoid is $\mathcal{C}(\epsilon)$. This monoid structure respects the partial order of the lattice, that is:

Lemma 86 *For concepts $X, Y, Z, W \in \mathfrak{B}$, if $X \leq Y$, then $W \circ X \circ Z \leq W \circ Y \circ Z$.*

We can extend the operation $\circ$ to the contexts of concepts:

$$(\vec{x}, \vec{y}) \circ (\vec{w}, \vec{z}) = (\vec{x}\vec{w}, \vec{z}\vec{y}). \tag{4.33}$$

This way, we still have $f \circ (g \circ h) = (f \circ g) \circ h$ for singleton contexts f, g, h. The operation can be extended to sets in the natural way, preserving associativity. For example, $C \circ (\epsilon, S) = \{(\vec{x}, \vec{a}\vec{y}) : (\vec{x}, \vec{y}) \in C, \vec{a} \in S\}$. We will use this as follows:

Definition 87 *Let $X = \langle S_X, C_X\rangle, Y = \langle S_Y, C_Y\rangle$ be concepts. We define the right residual $X/Y := \mathcal{C}(C_1 \circ (\epsilon, S_2))$, and the left residual $Y\backslash X := \mathcal{C}(C_1 \circ (S_2, \epsilon))$.*

For the closed sets of strings S, T, define $S/T := \{\vec{w} : \text{for all } \vec{v} \in T, \vec{w}\vec{v} \in S\}$. We then have $S_X/S_Y = S_{X/Y}$. So residuals are unique and satisfy the following lemma:

Lemma 88 *For $X, Y, Z \in \mathfrak{B}$, we have $Y \leq X\backslash Z$ iff $X \circ Y \leq Z$ iff $X \leq Z/Y$.*

For a proof, see [6]. This shows that the syntactic concept lattice can be enriched to a residuated lattice. Note that every language, whether computable or not, has a syntactic concept lattice. An important question is whether it is finite or not. This question can be answered in the following way.

Proposition 89 *The syntactic concept lattice for a language L is finite if and only if L is regular.*

[4] See [19], p.173 for more on this notion.

This is a rather immediate consequence of the Myhill-Nerode theorem. In the sequel, we will denote by SCL the class of all syntactic concept lattices, that is, the class of all lattices of the form $SCL(L)$ for some language L, without any further requirement regarding L itself except the ones stated above. There are two important conventions we urge the reader to keep in mind. We have denoted the set of concepts (for a language L) by $\mathfrak{B}_L$. For simplicity, if we want to say that X, Y are concepts of the lattice $SCL(L) = (\mathfrak{B}_L, \vee, \wedge, \circ, /, \backslash)$, we just write $X, Y \in SCL(L)$. The second convention is the following: concepts are ordered pairs. Given the language of reference, we can however recover all their relevant information from their extend, that is, their first component. Therefore, we will be sometimes treat concepts as if they were simple sets of strings rather than pairs. When we do so, we always refer to the first component of a concept.

We say a set (of strings) W is **downward closed** (with respect to $\leq_L$), if from $\vec{w} \in W$ and $\vec{v} \leq_L \vec{w}$ it follows that $\vec{v} \in W$.

Lemma 90 *Given a language $L \subseteq \Sigma^*$ and a set of strings $W \subseteq \Sigma^*$, if $W = W^{\triangleright\triangleleft}$, then W is downward closed wrt. $\leq_L$.*

Proof. Assume $W = W^{\triangleright\triangleleft}$, $\vec{v} \leq_L \vec{w}$, $\vec{w} \in W$. We know that for all $(\vec{a}, \vec{b}) \in W^{\triangleright}$, $\vec{a}\vec{w}\vec{b} \in L$. By $\vec{v} \leq_L \vec{w}$ it follows that also $\vec{a}\vec{v}\vec{b} \in L$, if $awb \in L$. Consequently, $\vec{v} \in W^{\triangleright\triangleleft}$, and so $W^{\triangleright\triangleleft}$ is downward closed. □

The converse implication does not obtain, that is: not every downward closed set is closed under $[-]^{\triangleright\triangleleft}$. This is because the $[-]^{\triangleright\triangleleft}$-closure considers only the L-contexts which are common to *all* strings in W. An interesting notion is the following:

Definition 91 *A language $L \subseteq \Sigma^*$ is **distributionally simple**, if for the set of concepts $SCL(L)$, we have $|SCL(L)| \leq |\Sigma|$.*

So distributional simplicity means: we have less concepts than letters. If we go to natural language, where letters become words, the obvious conjecture would be that our natural language datasets (just words!) are distributionally simple. This is the assumption, under which the application of concepts is an advantage – otherwise it can be arguable. Obviously, only regular languages can be distributionally simple, provided we have a finite alphabet – but for us this is no reason to worry, as we stick to finite languages.

4.9 Analogies and Inferences with Powersets

We start by providing some basic properties of the concept lattices of finite languages.

Lemma 92 *In any finite language $I \neq \emptyset$, we have $\{\epsilon\} = \{\epsilon\}^{\triangleright\triangleleft}$.*

Proof. Assume we have a $\vec{w} \in \Sigma^+$, $\vec{w} \in \{\epsilon\}^{\triangleright\triangleleft}$. As we have some $\vec{v} \in I$, we have $\vec{v}\epsilon \in I$, and so $\vec{v}\vec{w} \in I$, and so $v\vec{w}\epsilon \in I$, and so $\vec{v}\vec{w}\vec{w} \in I$ and so on. □

Assume $\leq$ is an arbitrary partial order. We say that a **covers** b (in $\leq$), if $a \leq b$, $a \neq b$, and if $a \leq c \leq b$, then $a = c$ or $b = c$.

Lemma 93 *Assume $I \subseteq \Sigma^+$ is a non-empty, finite language. Then there are concepts of the form $\mathcal{C}(a) : a \in \Sigma$ which cover $\perp$.*

Proof. Assume for every $\mathcal{C}(a) : a \in \Sigma$ there is an $X \leq \mathcal{C}(a)$. As a is the $\leq_I$-largest string in $\mathcal{C}(a)$, $X \subseteq \mathcal{C}(a)$, we have some $\vec{w} \leq_I a$ with $|\vec{w}| > 1$ for all $a \in \Sigma$. As we have some $\vec{x}a\vec{y} \in I$, this means we have $\vec{x}\vec{w}\vec{y} \in I$; as $\vec{w} = \vec{x}'a'\vec{y}'$, and as some $\vec{w}' \leq a'$, we have $\vec{x}\vec{x}'\vec{w}'\vec{y}'\vec{y} \in I$; as $\vec{w}' = \vec{x}''a''\vec{y}''$ etc., and I is infinite. □

This result is strongly related to the fact that for every $a \in \Sigma$, if there is a $\vec{w} \leq_I a$, then $a \not\sqsubseteq \vec{w}$.

Lemma 94 *In a finite language I, $\mathcal{C}(\epsilon)$ covers $\perp$; in other words: there is no $\vec{w} \leq_I \epsilon$.*

Proof is immediate after all. We now come to the definition of the analogical maps. We first define the map $\mathcal{C}P1$, the conceptual counterpart of $P1$.

Definition 95 *Given a finite language I, $V, W \in SCL(I)$, $W \neq V$, we have $(V, W) \in \mathcal{C}P1(I)$ if and only if*

1. *$V \leq_{SCL(I)} W$, and*
2. *there are $X, Y \in SCL(I)$ such that $X \circ V \circ Y = W$.*

Moreover, we have $(W, V) \in \mathcal{C}Pr(I)$ if and only if

1. *$V \leq_{SCL(I)} W$*
2. *there are concepts X, Y such that $X \circ V \circ Y = W$,*
3. *for $X' \neq Z_1 \circ X, Y' \neq Y \circ Z_2$, we have $X' \circ V \circ Y' \leq \mathcal{C}(I) \Leftrightarrow X' \circ W \circ Y' \leq \mathcal{C}(I)$.*

We also write $V \approx_I^{\mathcal{C}P1} W$ for $(V, W) \in \mathcal{C}P1(I)$, and if $(V, W) \in \mathcal{C}Pr(I)$, we also write $V \approx_I^{\mathcal{C}Pr} W$ or say that V, W are pseudo-recursive in I.

This differs somewhat from the definition on strings; we quickly show that the two nonetheless coincide in the case we talk about concepts on strings.

Lemma 96 *Assume $\vec{w} \sqsubset \vec{v}$, and I is a finite language. Then we have $(\vec{w}, \vec{v}) \in P1(I)$, if and only if $(\mathcal{C}(\vec{w}), \mathcal{C}(\vec{v})) \in \mathcal{C}P1(I)$.*

Proof. We prove the bi-implication by proving bi-implications of the two conditions.

1. Assume $\vec{w} \leq_I \vec{v}$. Then it follows that if $\vec{x} \in \mathcal{C}(\vec{w})$, then $\vec{x} \in \mathcal{C}(\vec{v})$, and so $\mathcal{C}(\vec{w}) \leq \mathcal{C}(\vec{v})$.

Conversely, $\mathcal{C}(\vec{x}) = \{\vec{y} : \vec{y} \leq_I (\vec{x})\}$. So it follows that if $\mathcal{C}(\vec{w}) \leq \mathcal{C}(\vec{v})$, then $\vec{w} \leq_I \vec{v}$.

2. Assume $\vec{w} \sqsubseteq \vec{v}$. Then $\vec{x}\vec{w}\vec{y} = \vec{v}$ for some $\vec{x}, \vec{y}$. Consequently, there are $\mathcal{C}(\vec{x}), \mathcal{C}(\vec{y})$, and $\mathcal{C}(\vec{x}) \circ \mathcal{C}(\vec{w}) \circ \mathcal{C}(\vec{y}) = \mathcal{C}(\vec{v})$ (this equation generally holds in a SCL).

Conversely, assume there are concepts X, Y such that $X \circ \mathcal{C}(\vec{v}) \circ Y = \mathcal{C}(\vec{w})$. By assumption, $\vec{v} \sqsubseteq \vec{w}$, and $\vec{w} \neq \vec{v}$, because otherwise $\mathcal{C}(w) = \mathcal{C}(v)$. □

So we have a proper generalization of $P1$, in that we allow $P1$ also for *sets of strings*, whereas the two coincide, if used for simple strings. For the case of Pr, things are slightly more complicated, we do not get a correspondence which is so simple, because we cannot characterize the recursive contexts in terms of strings. Still, there is an implication:

Lemma 97 *Assume I is a finite language. If $(\vec{w}, \vec{x}\vec{w}\vec{y}) \in Pr(I)$, then $(\mathcal{C}(\vec{w}), \mathcal{C}(\vec{x}) \circ \mathcal{C}(\vec{w}) \circ \mathcal{C}(\vec{y})) \in \mathcal{C}Pr(I)$.*

Proof. The first two conditions are clear from the previous proof. Assume $\vec{x}\vec{w}\vec{y} = \vec{v}$, and if $(\vec{a}\vec{w}\vec{b}) \in I$, $(\vec{a}, \vec{b}) \neq (\vec{a}'\vec{x}, \vec{y}\vec{b}')$, then $\vec{a}\vec{v}\vec{b} \in I$. Now assume there are concepts Z_1, Z_2 such that $Z_1 \circ \mathcal{C}(\vec{w}) \circ Z_2 \leq \mathcal{C}(I)$ and $Z_1 \circ \mathcal{C}(\vec{v}) \circ Z_2 \not\leq \mathcal{C}(I)$. Then there is $\vec{z}_1\vec{v}\vec{z}_2 \in Z_1 \circ \mathcal{C}(\vec{v}) \circ Z_2$, such that $\vec{z}_1\vec{v}\vec{z}_2 \notin I$, because $\vec{v}$ is $\leq_I$-maximal in $\mathcal{C}(\vec{v})$. At the same time, $\vec{z}_1\vec{w}\vec{z}_2 \in I$, because $Z_1 \circ \mathcal{C}(\vec{w}) \circ Z_2 \leq \mathcal{C}(I)$. Therefore, by assumption we must have $\vec{z}_1 = \vec{z}_1'\vec{x}, \vec{z}_2 = \vec{y}\vec{z}_2$. Therefore, $Z_1 \geq Z_1' \circ \mathcal{C}(\vec{x})$, $Z_2 \geq \mathcal{C}(\vec{y}) \circ Z_1'$. So we just have to show equality. Assume $Z_1' \circ \mathcal{C}(\vec{x}) < Z_1$ for arbitrary Z_1'. It follows that $Z_1 \circ A \geq Z_1' \circ \mathcal{C}(x) \circ A$ for all concepts A; same for Z_2. So there is a string $\vec{z}_1 \in Z_1$, $\vec{z}_1 \neq \vec{z}'\vec{x}$, $\vec{z}_1 \in Z_2$, $\vec{z}' \neq \vec{y}\vec{z}''$, such that both $\vec{z}_1\vec{v}\vec{z}_2 \notin I$, $\vec{z}_1\vec{w}\vec{z}_2 \in I$. Contradiction. □

What we also get another characterization of recursive contexts in $\mathcal{C}Pr$:

Lemma 98 *Let X, Y be concepts of $SCL(L)$. We have $W \leq V$, $X \circ W \circ Y = W$, iff and only iff $\mathcal{C}(\epsilon) \leq X \wedge Y$.*

Proof. If: $W \leq V$ if and only if $W \subseteq V$ (speaking of extents rather than concepts). Now assume $X \circ W \circ Y = V$. If $\mathcal{C}(\epsilon) \leq X \wedge Y$, then $\epsilon \in X \cap Y$. So we have $W \subseteq XWY \subseteq X \circ W \circ Y$.

Only if: Conversely, assume we have $W \leq V$ and $X \circ W \circ Y = V$, and assume $\epsilon \notin X \cap Y$. Then every string in V must be strictly longer than a string in W. But this contradicts $W \subseteq V$. □

Note that this property is not restricted to the finite, but also holds for infinite languages. What about a pre-theory as $(\mathfrak{g}2, P2)$? Here we see the full disadvantage of the conceptual approach: there does not seem to be a meaningful way to transfer it to this level. For example, try the following:

We have $A \circ C \circ E \approx_I^{P2} A \circ B \circ C \circ D \circ E$, if and only if $(A \circ C \circ E) \vee (A \circ B \circ C \circ D \circ E) \leq \mathcal{C}(I)$.

For the elementary condition, there is no reasonable way of transferring it, because the composition of a concept by $\circ$ tells us little if anything about the shape of the strings it contains – a concept of the form $(A \circ B \circ C \circ D \circ E)$ might contain strings of one letter etc. There is however some hope:

Lemma 99 *In a finite language I, concepts $X_1 \neq \mathcal{C}(\epsilon) \neq X_2$, of $SCL(I)$, we always have $X_1 \circ X_2 \not\leq X_1 \wedge X_2$.*

Proof. Assume $X_1 \circ X_2 \leq X_1$. As $X_1 \neq \mathcal{C}(\epsilon) \neq X_2$, for each string in X_1, we find a string in $X_1 \circ X_2$ which is strictly longer. As there is an upper bound to string length, we cannot have $X_1X_2 \subseteq X_1$. Same for X_2, and the claim follows. □

So in a sense, in finite languages the $\circ$-operation is more informative on the vertical structure than in infinite languages, but still there are many open problems we cannot address here.

We now come to our inference rules. Obviously, these have to look differently, as they proceed over sets of strings. For $\vec{w}, \vec{v} \in \Sigma^*$, $A \subseteq \Sigma^*$, write $\vec{w}A\vec{v}$ for $\{\vec{w}\vec{a}\vec{v} : \vec{a} \in A\}$; similarly for $\vec{w}A\vec{v}B\vec{u}$ etc. To get in line with the previous formats, we now have inference rules of the following kind:

$$\frac{\vdash \vec{w}\vec{x}_1\vec{v} \in I \quad ... \quad \vdash \vec{w}\vec{x}_i\vec{v} \in I}{\vdash \vec{w}\{\vec{x}_1, ..., \vec{x}_i\}\vec{v} \subseteq \mathfrak{f}_P(I)} \tag{4.34}$$

We allow this for all finite sequences of premises. Thereby, we get the premises we need to apply our analogies on concepts. We also need to be able to get rid of sets and go back to simple linguistic judgments; we do this as follows:

$$\frac{\vdash \vec{w}V\vec{v} \subseteq \mathfrak{f}_P(I) \quad \vec{x} \in V}{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}_P(I)} \tag{4.35}$$

By the requirement that the number of premises be finite (though arbitrarily large), we only allow finite sets to figure in our inferences. Now we can come to the "major" inference rules. To motivate our treatment, we first present a scheme which is inadequate. The problem is that by using only concepts, we lose the structure inherent in the strings, or put differently, we do not see it any more and thus cannot refer to it when making inferences. Take the following inference rule:

$$\frac{\vdash \vec{w}B\vec{v} \subseteq L \quad B \Leftarrow_L^P A}{\vdash w(A)\vec{v} \subseteq L} \tag{4.36}$$

That does not work recursively, because we do not "see" B within A. So we have to find another solution. We will use the following: we write analogies on explicit terms (which is sufficient by our conditions), that is, for $A = C \circ B \circ D$, instead of $B \Leftarrow_L^P A$ we write $B \Leftarrow_L^P C \circ B \circ D$, which is just another equivalent way of writing the same concept. Now, inference rules do *not* introduce the resulting concepts, but rather the concatenation of their extents:

$$\frac{\vdash wB\vec{v} \subseteq L \quad B \Leftarrow_L^P C \circ B \circ D}{\vdash w(CBD)\vec{v} \subseteq L} \tag{4.37}$$

The interpretation – that is, string denotation – of (CBD) is *not* $C \circ B \circ D = (CBD)^{\triangleright\triangleleft}$, but simply CBD, the pointwise concatenation without closure. This makes subconcepts accessible and allows us to refer to the same concept recursively. But more than this, it also warrants us from unwanted inferences. Assume we have the language $I = \{x, axb, y\}$. Here we have $\mathcal{C}(x) \approx_I^{\mathcal{C}Pr} \mathcal{C}(axb)$, and $y \in \mathcal{C}(axb)$. However, we do not want to make inferences on y, as it is not involved in any pseudo-recursive constellations. So our solution is both necessary and preferable on conceptual grounds. So we denote the inference rules – inference rules in (4.34),(4.35),(4.37), plus standard inference of analogies from similarity – by $\mathcal{C}\mathfrak{g}$, so that we get two new pre-theories: $(\mathcal{C}\mathfrak{g}, \mathcal{C}P1)$ and $(\mathcal{C}\mathfrak{g}, \mathcal{C}Pr)$.

4.9.1 Upward Normality and (Weak) Monotonicity

Obviously, things are more complicated with concepts than with strings. What are the main factors in this complication? The main reason is easily identified as the following: the monoid of concepts with $\circ$, $(\mathfrak{B}_L, \circ, \mathcal{C}(\epsilon)$ is not free, and so we cannot think of $\circ$ as in terms of concatenation. This means we lose the uniqueness of decomposition property. What consequences does this have? First of all, if we strive for characteristicity or downward normality, this makes the search much more complicated. For us, the main step was the proof of downward normality was that from the derivability of a certain string in $\mathfrak{f}_P(I)$, we could conclude that there was a certain string in I. Now this is much more complicated,

if possible at all, because analogies do not directly reveal the strings which are involved.

Upward normality, weak and strong monotonicity on the other side are unproblematic: the normalizing maps can be adapted accordingly, and the powerset construction will still work; we can also easily make sure that our projections are closed.

Regarding upward normality, things are quite easy again. We can take the normalizing maps $p_{(\mathcal{C}\mathfrak{g},\mathcal{C}P1)}$, $p_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}$, Recall that from the definitions of the last section, it follows that

$$p_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}(I) = I - max_{rad^*}(per_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}(I)), \tag{4.38}$$

where $per_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}(I) := \{M \subseteq I : I \subseteq \mathcal{C}\mathfrak{g}_{\mathcal{C}Pr}(I-M)\}$; and $p\mathcal{C}\mathfrak{g}_{\mathcal{C}Pr\circ p_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}}(I) = \mathcal{C}\mathfrak{g}_{\mathcal{C}Pr} \circ p_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}(I)$.

This yields an upward normal pre-theory and projection, as already follows from the previous results. The proof of upward normality in the last section did not make reference to the pre-theory itself, just to the set-theoretic properties of the order rad^*. The same holds for weak monotonicity. So whereas our ontology in terms of analogy has changed, the resulting projections itself still map sets of strings to sets of strings, so everything which works on the level of sets and projections works fine as before. So we have the following corollary:

Lemma 100 *$(p\mathcal{C}\mathfrak{g}, \mathcal{C}Pr \circ p_{(\mathcal{C}\mathfrak{g},\mathcal{C}Pr)}), (p\mathcal{C}\mathfrak{g}, \mathcal{C}P1 \circ p_{(\mathcal{C}\mathfrak{g},\mathcal{C}P1)})$ are upward normal and weakly monotonic.*

More problematic is the map q. Already above, we could not prove that q yields upward normality, but only under the assumption that for a pre-theory $(\mathfrak{f}, P)$, we find for every language $I, J : I \subseteq J \subseteq \mathfrak{f}_P(I)$, a language $J' \supseteq J$ such that $\mathfrak{g}_P(I) \supseteq \mathfrak{g}_P(J')$. If we could not even prove this for Pr, we cannot prove it for $\mathcal{C}Pr$ either, because of the following theorem:

Theorem 101 *Given a finite language I, we have 1. $\mathfrak{g}_{P1}(I) \subseteq \mathcal{C}\mathfrak{g}_{\mathcal{C}P1}(I)$ and 2. $\mathfrak{g}_{Pr}(I) \subseteq \mathcal{C}\mathfrak{g}_{\mathcal{C}Pr}(I)$.*

Proof. 1. Follows from lemma 96 above: assume we have $(\vec{x}, \vec{y}) \in P1(I)$, $\vec{w}\vec{x}\vec{v} \in I$. Then we have $\mathcal{C}(\vec{w})\mathcal{C}(\vec{x})\mathcal{C}(\vec{v}) \subseteq I$, and $(\mathcal{C}(\vec{x}), \mathcal{C}(\vec{y})) \in \mathcal{C}P1(I)$. So the claim follows.

2. This follows by the same argument from lemma 97. □

So even if q is normalizing for Pr, it is not guaranteed that it is normalizing for $\mathcal{C}Pr$; but the converse is true.

Regarding monotonicity, there is nothing which has changed considerably, that is: the powerset construction works as before, and all results can be obtained as before. The reason is that all results on monotonicity and the powerset construction for pre-theories and projections were obtained by purely set-theoretic methods, not by language-theoretic techniques, so they hold regardless of the pre-theory and its primary objects. As we have given the definitions in their full generality already in the preceding section, we urge the critical reader to verify our claims checking the section on monotonicity.

4.9.2 Reducing Lattices to Languages

The main problem is that we cannot treat concepts like letters in a language. We will now show some preliminary results on how we might nonetheless reduce them in some way to simple languages. The technical side will be quite self-explaining, but one has to be careful what these results mean. They do *not* mean that results from string-based pre-theories can be carried over to concepts. They mean that the results can be transferred if we are happy with thinking of our concepts as the basic letters of a new language, that is, the basic language-theoretic objects, which only in the very final spell-out are substituted by strings. This might be quite desirable: we can think of our concepts as syntactic categories (or rather: pre-categories), and project not the string language, but the language of syntactic categories. I guess most linguists would be fine with this procedure, and many would prefer it over a procedure which simply works with strings. The main reason is: presumably, it makes language less messy than it is on the level of visible strings. And from a language-theoretic point of view, our procedure of pre-categorization is very well-defined, so on grounds of formality there is nothing to object. Still we have to be aware that we lose the very immediate touch with the "visible language" we had before, regardless of whether we consider this an advantage or rather a disadvantage.

Given a language $L \subseteq \Sigma^*$, we denote by $S(L)$ the partially ordered monoid $([\Sigma^*]_{\sim_L}, \leq_L, \bullet)$, where $\leq_L$ is the linguistic order, which is extended from strings, where it forms a pre-order, to equivalence classes of the form $[\vec{w}]_L := \{\vec{v} : \vec{v} \sim_L \vec{w}\}$, where it forms a partial order; and $\bullet$ is defined by $[\vec{w}]_L \bullet [\vec{v}]_L = [\vec{w}\vec{v}]_L$. Note that the monoid operation $\bullet$ is also not free, and in general, we might have $[\vec{w}]_L[\vec{v}]_L \neq [\vec{w}\vec{v}]_L$. (see [9] on this problem).

We now present the extensionality lemma, which correlates the extension of concepts (set-theoretically) with their combinatorial properties in the lattice. This result does not hold in all residuated lattices, but only for syntactic concept lattices! The reason that this can be despite the completeness theorem in [72] is that it is not an inequation of the lattice itself, but a statement in our meta-language.

Lemma 102 *(Extensionality Lemma)*

1. *For $X, Y \in SCL(L)$, if $X \neq Y$, then there are $X_1, X_2 \in SCL(L)$, such that $X_1 \circ X \circ X_2 \leq \mathcal{C}(L)$ and $X_1 \circ Y \circ X_2 \not\leq \mathcal{C}(L)$ or vice versa.*

2. *If for all $X_1, X_2 \in SCL(L)$, we have $X_1 \circ X \circ X_2 \leq X_1 \circ Y \circ X_2$, then $X \leq Y$.*

Proof. 1. Assume we have $X \neq Y$. Then we also have $X^\triangleright \neq Y^\triangleright$. Assume wlog that there is $(\vec{x}, \vec{y}) \in X^\triangleright, (\vec{x}, \vec{y}) \notin Y^\triangleright$. Then it follows that $\mathcal{C}(\vec{x}) \circ X \circ \mathcal{C}(\vec{y}) \leq \mathcal{C}(L)$, because $\vec{x}X\vec{y} \subseteq L$, and for all $\vec{x}' \in \mathcal{C}(x)$, $\vec{x}' \leq_L x$, for all $\vec{y}' \in \mathcal{C}(\vec{y})$, $\vec{y}' \leq_L \vec{y}$. Conversely, as $(\vec{x}, \vec{y}) \notin Y^\triangleright$, we have $\vec{x}Y\vec{y} \not\subseteq L$, and so $\mathcal{C}(\vec{x}) \circ X \circ \mathcal{C}(\vec{y}) \not\leq \mathcal{C}(L)$.

2. Contraposition: assume we have $X \not\leq Y$. Then by the Galois-connection, we have $Y^\triangleright \not\subseteq X^\triangleright$, and so there exists $(\vec{x}, \vec{y}) \in Y^\triangleright, (\vec{x}, \vec{y}) \notin X^\triangleright$. As in 1., we know that $\mathcal{C}(\vec{x}) \circ Y \circ \mathcal{C}(\vec{y}) \leq \mathcal{C}(L)$, but $\mathcal{C}(\vec{x}) \circ X \circ \mathcal{C}(\vec{y}) \not\leq \mathcal{C}(L)$. From this it follows that $\mathcal{C}(\vec{x}) \circ X \circ \mathcal{C}(\vec{y}) \not\leq \mathcal{C}(\vec{x}) \circ Y \circ \mathcal{C}(\vec{y})$. □

In the sequel, we will embed (monoid reducts of) concept lattices in languages. In all of what is to follow, we are interested in (combinations of) concepts being

smaller than $\mathcal{C}(L)$, the concept of the strings of the language with respect to which we form our concepts. There is one problem about this approach: let $L \subseteq \Sigma^*$. There is always a largest concept $\mathcal{C}(\Sigma^*)$; and it can easily happen (though not necessarily) that we have $(\Sigma^*)^\triangleright = \emptyset$. In that case, there are no concepts X, Y such that $X \circ \mathcal{C}(\Sigma^*) \circ Y \leq \mathcal{C}(L)$, and consequently, the techniques presented below do not work for this concept. In particular, there is no general way to translate this concept into an equivalence class, because there might be a language, where each string in Σ^* figures as some substring, yet the language does not equal Σ^* – just think of the palindrome or copy languages! So for all that is to follow, we exclude this concept from consideration, which does no harm as it is trivial and syntactically uninformative anyway.

Definition 103 *We say a SCL over a language L' is embedded in a language $L \subseteq \Sigma^*$, if there is a injective map $i : SCL(L') \to [\Sigma]_{\sim_L}$, such that $i(\mathcal{C}_1 \circ \mathcal{C}_2) = [i(\mathcal{C}_1)i(\mathcal{C}_2)]_L$, and $\mathcal{C}_1 \leq \mathcal{C}_2 \Leftrightarrow i(\mathcal{C}_1) \leq_L i(\mathcal{C}_2)$. In other words, we require there to be an embedding of the reduct $(\mathfrak{B}_{L'}, \circ, \mathcal{C}(\epsilon))$ in $S(L)$. If in addition, i is surjective, we write $S(L) \cong SCL(L')$.*

A first reduction result goes as follows:

Theorem 104 *For each finite language I, concept lattice $SCL(I)$, there is a finite language J over a finite alphabet such that $S(J) \cong SCL(I)$.*

Proof. Let I be a finite language and $SCL(I)$ its concept lattice. We construct Σ as follows: for every $X \in SCL(I)$, we have a letter $x \in \Sigma$, such that there is a bijection $i : SCL(I) \to \Sigma$. We now define $I_\mathcal{C} \subseteq \Sigma^*$ by $x_1 x_2 ... x_i \in I_\mathcal{C}$, if and only if we have $i^{-1}(x_1) \circ i^{-1}(x_2) \circ ... \circ i^{-1}(x_i) \leq \mathcal{C}(I)$.

We now prove that $S(I_\mathcal{C}) \cong SCL(I)$. First, we extend i^{-1} to strings in the usual fashion: $i^{-1}(a\vec{w}) = i^{-1}(a) \circ i^{-1}(\vec{w})$; we thus have a homomorphism i^{-1} from Σ^* onto the syntactic concepts.

1. There is a bijection from $SCL(I)$ to $[\Sigma^*]_{\sim_{I_\mathcal{C}}}$.

Let $Y \in [\Sigma^*]_{\sim_{I_\mathcal{C}}}$ be an equivalence class over $I_\mathcal{C}$, and let $Y^\triangleright$ be the set of its contexts in $I_\mathcal{C}$. Then by definition, for $(x_1...x_i, x_j...x_k) \in Y^\triangleright$, $x_l...x_n \in Y$, we have $x_1..x_i x_l...x_n x_j...x_k \in I_\mathcal{C}$, and thereby $i^{-1}(x_1..x_i x_l...x_n x_j...x_k) \leq \mathcal{C}(I)$. In turn, by the extensionality lemma, this means that for every $Y \in [\Sigma^*]_{\sim_{I_\mathcal{C}}}$, we have a separate concept. Now assume we have two concepts X, X'. Then by the extensionality lemma, we have a distinguishing context in $I_\mathcal{C}$. This shows that there is a bijection between concepts of I and equivalence classes of $I_\mathcal{C}$.

2. $\circ$

We just have to show that if $X \circ Y = Z$, then we have $i(X)i(Y) \sim_{I_\mathcal{C}} i(Z)$. This is straightforward, because we have $i^{-1}(xy) \leq \mathcal{C}(I)$ if and only if $i^{-1}(z) \leq \mathcal{C}(I)$.

3. $\leq$

We show that $X \leq Y$ if and only if $i(X) \leq_{I_\mathcal{C}} i(Y)$.

Only if: assume $X \leq Y$. Then it follows that if have $\overline{X} \circ Y \circ \overline{X}' \leq \mathcal{C}(I)$, then $\overline{X} \circ X \circ \overline{X}' \leq \mathcal{C}(I)$. So if $i(\overline{X})i(Y)i(\overline{X}') \in I_\mathcal{C}$, then $i(\overline{X})i(X)i(\overline{X}') \in I_\mathcal{C}$.

If: Assume we have $\vec{z} \leq_{I_\mathcal{C}} \vec{u}$. Then we have $\vec{x}\vec{u}\vec{x}' \in I_\mathcal{C} \Rightarrow \vec{x}\vec{z}\vec{x}' \in I_\mathcal{C}$; consequently, $i^{-1}(\vec{x}) \circ i^{-1}(\vec{u}) \circ i^{-1}\vec{x}' \leq \mathcal{C}(I) \Rightarrow i^{-1}(\vec{x}) \circ i^{-1}(\vec{z}) \circ i^{-1}\vec{x}' \leq \mathcal{C}(I)$.

By the extensionality lemma part 2, it follows that $i^{-1}(\vec{z}) \leq i^{-1}(\vec{u})$. □

Note that the proof works equally well with an infinite language L; but if the language is not regular, we will have an infinite alphabet for $L_\mathcal{C}$. This is a very

good result, but note that the usage of equivalence classes puts some difficulties to us:

Lemma 105 *Let $I \subseteq \Sigma^*$ be a finite language, and assume that we have $\vec{x} \sim_I \vec{y}, \vec{x}_1 \sim_I \vec{y}_1, \vec{x}_2 \sim_I \vec{y}_2$, where $\vec{x} \neq \vec{y}$ and $\vec{x}_i \neq \vec{y}_i$ for $i \in \{1, 2\}$. Then $\vec{x} \not\approx_I^{Pr} \vec{x}_1\vec{x}\vec{x}_2$, and $\vec{y} \not\approx_I^{Pr} \vec{y}_1\vec{y}\vec{y}_2$.*

Proof. Assume wlog that $\vec{x}_1 \neq \vec{y}_1$. First of all, we have to make a premiss explicit, which is implicit in the assumption that I is finite, namely that $\vec{x} \not\sqsubseteq \vec{y}$, and $\vec{x}_1 \not\sqsubseteq \vec{y}_1$, and vice versa. That follows from a basic lemma on finite languages.

By assumption, that $\vec{x} \approx_I^{Pr} \vec{x}_1\vec{x}\vec{x}_2$, it follows that we have some $\vec{w}\vec{x}\vec{v} \in I$, where $(\vec{w}, \vec{v})$ is not recursive for $(\vec{x}, \vec{x}_1\vec{x}\vec{x}_2)$. Then we have $\vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v} \in I$. By $\vec{x} \sim_I \vec{y}$ we can infer $\vec{w}\vec{x}_1\vec{y}\vec{x}_2\vec{v} \in I$. Now, $(\vec{w}\vec{x}_1, \vec{x}_2\vec{v})$ is not a recursive context for $(\vec{y}, \vec{y}_1\vec{y}\vec{y}_2)$ (otherwise $\vec{x}_1$ would be a substring of $\vec{y}_1$ or vice versa); therefore we have $\vec{w}\vec{x}_1\vec{y}_1\vec{y}\vec{y}_2\vec{x}_2\vec{v} \in I$. Therefore, we have $\vec{w}\vec{x}_1\vec{y}_1\vec{x}\vec{y}_2\vec{x}_2\vec{v} \in I$, and for the same reason as above, $(\vec{w}\vec{x}_1\vec{y}_1, \vec{y}_2\vec{x}_2\vec{v})$ is not recursive for $(\vec{x}, \vec{x}_1\vec{x}\vec{x}_2)$, therefore we have...and so on, so I must be infinite, contradiction. □

So this shows us: we cannot just talk about equivalence classes as we can talk about strings; the notion of pseudo-recursion is fundamentally at odds with the notion of equivalence classes. This result also puts us in guard: in interpreting concepts as equivalence classes we have to be very careful, and the first reduction theorem is not as useful as it seems.

Now the question is: assuming we can perform a reduction back to strings without any substantial loss, we are (almost) back where we have been before; maybe we have not lost anything substantial with respect to the simple approach – but what have we gained? The answer to this question lies in the properties of our concept language, which has particular properties corresponding to the concept lattice. For example, we recognized the problem that with string-based pre-theories we cannot distinguish between different distributions of strings, for example: $\vec{x}$ in the position where both $\vec{x}, \vec{y}$ occur. Concepts obviously solve this problem. How is the solution preserved in the language reduction? The key is a property of the reduction language.

Definition 106 *We say a language $L \subseteq \Sigma^*$ is **distributional**, if every $X \in SCL(L)$, there is a $\vec{w} \in \Sigma^*$ such that $X = \downarrow_L \vec{w} := \{\vec{v} : \vec{v} \leq_L \vec{w}\}$.*

This means: for every distribution (closed set of contexts), which is characterized by the occurrence of some set of strings, we find a single word which characterizes it; more explicitly: for every $S \subseteq \Sigma^*$, there is a word $\vec{w} \in \Sigma^*$ such that $\vec{x}\vec{w}\vec{y} \in L$ iff $\vec{x}S\vec{y} \subseteq L$ (excluded the special case where there is no context for S). For distributional languages, we find the following nice property.

Given a set $S \subseteq \Sigma^*$, we have $S^u := \{\vec{t} : \text{for all } \vec{s} \in S,\ \vec{s} \leq_L \vec{t}\}$; and similarly, $S^l := \{\vec{t} : \text{for all } \vec{s} \in S,\ \vec{t} \leq_L \vec{s}\}$. In the context of the order induced by a language, the set S^u denotes the set of all strings $\vec{t}$, such that for any $\vec{x}, \vec{y} \in \Sigma^*$, if $\vec{x}\vec{t}\vec{y} \in L$, then $\vec{x}\vec{s}\vec{y} \in L$ for all $\vec{s} \in S$. So it is the set of substrings, for which we can substitute all $\vec{s} \in S$. S^l conversely is the set of strings, which we can take as substitute for all $\vec{s} \in S$: if $\vec{w} \in S^l$, $\vec{s} \in S$, then from $\vec{x}\vec{s}\vec{y} \in L$ it follows that $\vec{x}\vec{w}\vec{y} \in L$.

The DM-completion of a partially ordered set $(P, \leq)$ is the lattice $(\{A \subseteq P : A^{ul} = A\}, \subseteq)$. For a partially ordered monoid, it respects the monoid operation, and creates unique preserves meets and joins, where we define for

A, B closed sets, $A \wedge B = A \cap B$, $A \vee B = (A \cup B)^{ul}$, and $A \cdot B = (AB)^{ul}$, where $AB := \{a \cdot b : a \in A, b \in B\}$ (for reference on this procedure, consider [19], p.173). We will show the following:

Lemma 107 *Let $L \subseteq \Sigma^*$ be a distributional language. The syntactic concept lattice of L is isomorphic to the DM-closure of the partially ordered monoid $(\Sigma^*_{/\sim}, \leq_L)$.*

(It is only isomorphic because the elements of concept lattices are pairs of sets of strings and sets of contexts given by a Galois connection; the elements of the DM completion are only sets of strings. These, however, are identical.)

Proof. We only show that the two closure operators $[-]^{ul}$ and $[-]^{\triangleright\triangleleft}$ coincide on any set $S \subseteq \Sigma^*$; the rest follows from properties of the connectives in terms of closure. So what we show is: $S^{\triangleright\triangleleft} = S^{ul}$.

$\subseteq$ Let $S \subseteq \Sigma^*$, and $\vec{w} \in S^{\triangleright\triangleleft}$. We show that $\vec{w} \in S^{ul}$. In case $\vec{w} \in S$, this is trivial as $[-]^{ul}$ is a closure operator. So assume $\vec{w} \notin S$. If $\vec{w} \in S^{\triangleright\triangleleft}$, this means that for all contexts $(\vec{a}, \vec{b})$, such that for all $\vec{s} \in S, \vec{a}\vec{s}\vec{b} \in L$, we also have $\vec{a}\vec{w}\vec{b} \in L$. S^u is the set of all words, for which all $\vec{s} \in S$ can function as substitute. So regarding the contexts $(\vec{a}, \vec{b}) \in S^{\triangleright}$, S^u consists of all and only the words which occur only (i.e., if at all) in the contexts in $S^{\triangleright}$. Now as $\vec{w} \in S^{\triangleright\triangleleft}$, we know that $\vec{w}$ can occur in all these contexts. Consequently, we can substitute $\vec{w}$ for all words $\vec{t} \in S^u$; therefore, $\vec{w} \in S^{ul}$.

$\supseteq$ As L is distributional, there is $\vec{u} \in \Sigma^*$ such that $\vec{x}\vec{u}\vec{y} \in L$ iff $\vec{x}S\vec{y} \subseteq L$, and so $\vec{u} \in S^u$. Now, every $\vec{w} \in S^{ul}$ must satisfy $\vec{w} \leq_L \vec{u}$; so this means by the above bi-implication: if $\vec{x}\vec{w}\vec{y} \in L$, then $\vec{x}S\vec{y} \subseteq L$; consequently, $\vec{u} \in S^{\triangleright\triangleleft}$. □

Note that we used the premise of L being distributional only on the proof of $\supseteq$; so otherwise, we still get an inclusion. On the other side, the lemma can be proved to be wrong without this additional premise. Furthermore, we have the following:

Lemma 108 *Given a language L, syntactic concept lattice $SCL(L)$, the concept language $L_{\mathcal{C}}$ is distributional.*

Proof. As SCL are complete lattices, we have, for any set $\mathbf{X} \subseteq SCL(L)$, $\bigvee \mathbf{X} \in SCL(L)$. This means that $i(W)i(X_i)i(V) \in L_{\mathcal{C}}$ for all $X_i \in \mathbf{X}$, if and only if $i(W)i(\bigvee \mathbf{X})i(V) \in L_{\mathcal{C}}$. □

So the distributionality of the concept language makes sure: even though we just speak about words, not about strings, there is nothing we lose, because every distributional property of a set of strings is the distributional property of a single string. I have to admit at this point that the reduction is still incomplete and does not permit substantial insights. Still it is an interesting way to go for further research. Now instead we will look at the first pre-theory which brings us beyond the context-free languages.

4.10 Context-freeness and Beyond: SCL_n

The extension based on syntactic concept lattices was mainly useful to make pre-theories more liberal and to capture additional patterns; it does not make us transcend the bounds of context freeness:

Lemma 109 *Given any finite language I, the language $\mathcal{C}\mathfrak{g}_{\mathcal{C}Pr}(I)$ is a context-free language.*

The proof is immediate and can be done using grammar construction as in the proof of theorem 20. So we have $\mathfrak{C}(\mathcal{C}\mathfrak{g}, \mathcal{C}Pr) \subseteq CFL$. So we are still in the situation that context-freeness of "languages" is a methodological universal, a result which goes against intuition of most researchers in the field of formal linguistics. We will now make our first attempt to transcend this boundary.

Syntactic concept lattices can be extended in a very natural way (see for example [8],[53]). The change is first of all in the underlying monoid. Whereas in the simple case, we had the monoid $(\Sigma^*, \cdot, \epsilon)$ underlying the concept analysis, we can now switch to the monoid $((\Sigma^*)^2, \cdot, (\epsilon, \epsilon))$, where we have $(\vec{w}_1, \vec{w}_2) \cdot (\vec{v}_1, \vec{v}_2) = (\vec{w}_1\vec{v}_1, \vec{w}_2\vec{v}_2)$. Obviously, this can be generalized to a monoid $((\Sigma^*)^n, \cdot, (\epsilon, ..., \epsilon))$ for any $n \in \mathbb{N}$. For sake of brevity we will denote $((\Sigma^*)^n, \cdot, (\epsilon, ..., \epsilon))$ simply by $(\Sigma^*)^n$. All what is to follow can be read as holding for all $n \in \mathbb{N}$, though we will mostly exemplify it with $n = 2$.

One might ask now the following question: why do we introduce this extension only for concepts, why do we not devise string-based pre-theories using this extended monoid? There is a good answer to this question: the problem with the monoids $(\Sigma^*)^n : n \geq 2$ is that it is no longer free either. To see this, take $a, b \in \Sigma$; then we have in $(\Sigma^*)^2$: $(a, b) = (a, \epsilon) \cdot (\epsilon, b) = (\epsilon, b) \cdot (a, \epsilon)$. So we lose the property of freeness, which in turn makes it very hard with our current methods to find characteristic or downward normal pre-theories.[5] So the situation is similar to the situation with concepts, which also yield an unfree monoid. So we can conclude: once we have lost freeness, the extension from (Σ^*) to $(\Sigma^*)^n$ gives us a clean gain. But as this step makes us lose freeness anyway, it does not seem to be very worthwhile to pursue this monoid in simple string-based pre-theories, as the main advantage of the simple string based pre-theories was that it was based on a free monoid.

We thus now extend syntactic concepts over $(\Sigma^*)^n$, exemplifying the construction with $n = 2$. As extents of syntactic concepts, we take subsets of $\Sigma^* \times \Sigma^*$ (instead of subsets of Σ^*), and as intents, we take subsets of $\Sigma^* \times \Sigma^* \times \Sigma^*$ (instead of $\Sigma^* \times \Sigma^*$), such that we have $[-]^\triangleright : \Sigma^* \times \Sigma^* \to \Sigma^* \times \Sigma^* \times \Sigma^*$, and dually $[-]^\triangleleft : \Sigma^* \times \Sigma^* \times \Sigma^* \to \Sigma^* \times \Sigma^*$. Given a language $L \subseteq \Sigma^*$, and $M \subseteq \Sigma^* \times \Sigma^*$, we put $M^\triangleright := \{(\vec{x}, \vec{y}, \vec{z}) : \forall (\vec{a}, \vec{b}) \in M, \vec{x}\vec{a}\vec{y}\vec{b}\vec{z} \in L\}$; $[-]^\triangleleft$ is defined inversely. Obviously, this can be easily generalized to sets of arbitrary $n, n+1$ tuples.

A syntactic concept is a pair (M, N) such that $M = N^\triangleleft$, $N = M^\triangleright$; but for convenience (as before), we sometimes simply ignore the second component. We define $\vee, \wedge$ as usual: $(M_1, N_1) \vee (M_2, N_2) = ((M_1 \cup M_2)^{\triangleright\triangleleft}, N_1 \cap N_2)$, and dually $(M_1, N_1) \wedge (M_2, N_2) = (M_1 \cap M_2, (N_1 \cup N_2)^{\triangleright\triangleleft})$. And finally, we define $(M_1, N_1) \circ (M_2, N_2) = ((M_1M_2)^{\triangleright\triangleleft}, (M_1M_2)^\triangleright)$, where $M_1M_2 = \{\vec{w}_1 \cdot \vec{w}_2 : \vec{w}_1 \in M_1, \vec{w}_2 \in M_2\}$; but note that $\cdot$ here denotes the generalized concatenation in $(\Sigma^*)^n$!

Denote the class of (generalized) syntactic concept lattices, where extents are sets of n-tuples, intents sets of $n+1$-tuples, by SCL_n. On the level of the

[5] As a side note, and to make visible of what kind the complications are, I want to point out that all complications, which arise when we go from simple finite automata to finite state transducers, can be traced back to this simple fact that the underlying monoid of transducers is not free in exactly the same way we described above. This leads, among other, to: no closure under intersection and complement, undecidability of inclusion and equivalence.

concepts themselves, everything is the same as before, and we would hardly see the difference if we would not know that the underlying monoid is different. But there are some differences. For example, the neutral element of a lattice $SCL_n(L)$ is $\mathcal{C}((\epsilon_1, .., \epsilon_n))$. This is fine; what is more problematic is that $\mathcal{C}(L) = (\epsilon_1, .., \epsilon_{n+1})^\triangleleft = \mathcal{C}(\{(\vec{w}, ..., \vec{w}_n) : \vec{w}_1 ... \vec{w}_n \in L\})$. This illustrates how there is not one concept corresponding to a single word, but rather one for each of its decompositions. Recall that in SCL, we had $\mathcal{C}(\vec{w}) =\downarrow \vec{w}$. In SCL_n, things are a bit different, and it is undefined what $\mathcal{C}(\vec{w})$ even means: we rather get a possibly different concept for every decomposition $\vec{w}_1 ... \vec{w}_n = \vec{w}$, namely $\mathcal{C}(\vec{w}_1, ..., \vec{w}_n)$.

So whereas on the level of concepts, everything looks neat as before, the relation from concepts to strings is considerably more complicated. And this is a complication with real consequences, because our ultimate point of reference is a language L, that is, a set of strings. So the question is: can our results be generalized from SCL (that is, SCL_1) to SCL_n for any $n \in \mathbb{N}$? The most important result for the relation from strings to the abstract concept language is the extensionality lemma. We will now show that a generalized version holds. We therefore need a new symbol: for $M \subseteq (\Sigma^*)^n$, $N \subseteq \Sigma^*$, we write $M \Subset N$ if for all $(\vec{w}_1, ..., \vec{w}_n) \in M$, $\vec{w}_1 ... \vec{w}_n \in M$.

Lemma 110 *(Generalized Extensionality Lemma)*

1. *For $X, Y \in SCL_n(L)$, if $X \neq Y$, then there are $X_1, X_2 \in SCL_n(L)$, such that $X_1 \circ X \circ X_2 \leq \mathcal{C}(L)$ and $X_1 \circ Y \circ X_2 \not\leq \mathcal{C}(L)$ or vice versa.*

2. *If for all $X_1, X_2 \in SCL_n(L)$, we have $X_1 \circ X \circ X_2 \leq X_1 \circ Y \circ X_2$, then $X \leq Y$.*

Proof. 1. Assume we have $X \neq Y$. Then again we have $X^\triangleright \neq Y^\triangleright$, because all laws of Galois connections still hold in the general case. Assume wlog that there is $(\vec{x}_1, ..., \vec{x}_n) \in X^\triangleright, (\vec{x}_1, ..., \vec{x}_n) \notin Y^\triangleright$. Then it follows that $(\mathcal{C}((\vec{x}_1, \epsilon, ..., \epsilon)) \circ \mathcal{C}((\epsilon, \vec{x}_2, ..., \epsilon)) \circ ... \circ \mathcal{C}((\epsilon, ..., \vec{x}_{n-1}))) \circ X \circ \mathcal{C}((\epsilon, ..., \epsilon, \vec{x}_n)) \leq \mathcal{C}(L)$, because $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_{n-1}) \cdot X \cdot (\epsilon, ..., \epsilon, \vec{x}_n) \Subset L$. Conversely, as $(\vec{x}_1, ..., \vec{x}_n) \notin Y^\triangleright$, we have $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_{n-1}) \cdot Y \cdot (\epsilon, ..., \epsilon, \vec{x}_n) \not\Subset L$, and so $(\mathcal{C}((\vec{x}_1, \epsilon, ..., \epsilon)) \circ \mathcal{C}((\epsilon, \vec{x}_2, ..., \epsilon)) \circ ... \circ \mathcal{C}((\epsilon, ..., \vec{x}_{n-1}))) \circ Y \circ \mathcal{C}((\epsilon, ..., \epsilon, \vec{x}_n)) \not\leq \mathcal{C}(L)$.

2. Contraposition: assume we have $X \not\leq Y$. Then by the Galois-connection, we have $Y^\triangleright \not\subseteq X^\triangleright$, and so there exists $(\vec{x}_1, ..., \vec{x}_n) \in Y^\triangleright, (\vec{x}_1, ..., \vec{x}_n) \notin X^\triangleright$. As in 1. we know that $\mathcal{C}((\vec{x}_1, \epsilon, ..., \epsilon)) \circ \mathcal{C}((\epsilon, \vec{x}_2, ..., \epsilon)) \circ ... \circ \mathcal{C}((\epsilon, ..., \vec{x}_{n-1})) \circ X \circ \mathcal{C}((\epsilon, ..., \epsilon, \vec{x}_n)) \not\leq \mathcal{C}(L)$, but $\mathcal{C}((\vec{x}_1, \epsilon, ..., \epsilon)) \circ \mathcal{C}((\epsilon, \vec{x}_2, ..., \epsilon)) \circ ... \circ \mathcal{C}((\epsilon, ..., \vec{x}_{n-1})) \circ Y \circ \mathcal{C}((\epsilon, ..., \epsilon, \vec{x}_n) \leq \mathcal{C}(L)$. From this it follows that there are X_1, X_2 such that $X_1 \circ X \circ X_2 \not\leq X_1 \circ Y \circ X_2$. □

So we can transfer this important result, which makes syntactic concept lattices more well-behaved for our purposes than residuated lattices in general. We now define the pre-theories and projections based on SCL_n. We can now define $\mathcal{C}_n P1$ as follows:

Definition 111 *Given a finite language I, $V, W \in SCL_n(I)$, $W \neq V$, we have $(V, W) \in \mathcal{C}_n P1(I)$ if and only if*

1. *$V \leq W$, and*

2. *there are $X, Y \in SCL_n(I)$ such that $X \circ V \circ Y = W$.*

One now might think that we can define $\mathcal{C}_nPr$ in the same fashion:

Definition 112 *(Preliminary Definition) Given a finite language I, $V,W \in SCL_n(I)$, $W \neq V$, we have $(V,W) \in \mathcal{C}_nPr(I)$, if and only if*

1. $V \leq W$
2. *there are concepts X,Y such that $X \circ V \circ Y = W$, and*
3. *for $X' \neq Z_1 \circ X, Y' \neq Y \circ Z_2$, we have $X' \circ V \circ Y' \leq \mathcal{C}(I) \Leftrightarrow X' \circ W \circ Y' \leq \mathcal{C}(I)$.*

On the level of concepts, this is perfectly fine, because concepts behave as before. If we go down to the string level however, we find that this does not do what we intend it to do: because for concepts over Σ^*, we had for every $M \subseteq \Sigma^*$ a unique $N \subseteq \Sigma^*$ such that $M \cdot N = MN$ (in fact, $M \cdot N$ and MN are notational variants). So we do have uniqueness up to $[-]^{\triangleright\triangleleft}$ closure. This does not obtain for $(\Sigma^*)^n$; we have already demonstrated this for elements of $(\Sigma^*)^n$; and the same holds *a fortiori* for subsets of $(\Sigma^*)^n$. This in turn means that for every concept X, we might have a number of distinct concepts $X_1, ..., X_i$ and $Y_1, ..., Y_i$ such that for $1 \leq j, j' \leq i$, we have $X_j \circ X \circ Y_j = X_{j'} \circ X \circ Y_{j'}$, because we already have $X_j \cdot X \cdot Y_j = X_{j'} \cdot X \cdot Y_{j'}$. So the definition of $\mathcal{C}_nPr$, as it stands above, is useless, it *cannot* be satisfied.

We therefore must introduce a new notion, namely the notion of $\circ$-equivalence. For SCL_n, we define $\sim_\circ \subseteq (SCL_n(L))^4$, and write $(X_1, X_2) \sim_\circ (Y_1, Y_2)$, if for all $Z \in SCL_2(L)$, we have $X_1 \circ Z \circ X_2 = Y_1 \circ Z \circ Y_2$. $\sim_\circ$ is thus an equivalence relation; note that in our convention, it does not make explicit reference to L or $SCL_n(L)$, though in principle it should. The same however already holds for $[-]^\triangleright$ etc., so we just use the concept in a way that avoids any possible confusion. Note that for SCL_1, $\circ$-equivalence is not necessarily vacuous, that is, $(X_1, X_2) \sim \circ(Y_1, Y_2)$ does not entail $X_1 = Y_1, X_2 = Y_2$. It is however quite restricted and bound to cases which are not relevant to pseudo-recursion. In the case of $SCL_n : n \geq 2$, it is absolutely crucial, because there is no way to do without.

Definition 113 *Given a finite language I, $V,W \in SCL_2(I)$, $W \neq V$, we have $(V,W) \in \mathcal{C}_2Pr(I)$ iff*

1. $V \leq W$
2. *there are concepts $X,Y \in SCL_2(I)$ such that $X \circ V \circ Y = W$, and*
3. *for $Z_1 \neq Z_1' \circ X', Z_2 \neq Y' \circ Z_2'$, where $(X', Y') \sim_\circ (X, Y)$, we have $Z_1 \circ V \circ Z_2 \leq L \Leftrightarrow Z_1 \circ W \circ Z_2 \leq L$.*

This gives us the notion we need. Take for example a language $I := \{abcd, axbyczd\}$. Here we have $\mathcal{C}_2Pr(I) = \{(\mathcal{C}(b,c), \mathcal{C}(xby, cz)), (\mathcal{C}(b,c), \mathcal{C}(xb, ycz))\}$, as can be checked (checking such conditions becomes more tedious with growing n, but is a good exercise though). Note that the two analogies illustrate a further problem: assume y in the above example is a word $\vec{y}$ of length n. Then if we have *one* analogy, we already get $n+1$ analogies, namely one for each decomposition of $\vec{y}$. This is another complication we have to be aware of. So problems regarding closure under coding etc. will get much more complicated.

We also have to adapt our inference rules; the old ones will no longer work, because our inferences now proceed over sets of *tuples*, and we do not have primitive judgments of the form $\vdash (\vec{w}, \vec{v}) \in I$. Rather, we first have to "deconstruct" judgments of the form $\vdash \vec{w} \in I$. Also, we have to take into account the fact that inferences now proceed over an unfree monoid, and therefore, "representation matters"; so we need to implement the equivalence of $((\Sigma^*)^n, \cdot)$-terms in inferences. We do this by the map γ, where $\gamma(\vec{x}_1, ..., \vec{x}_i) = \vec{x}_1 ... \vec{x}_i$; so γ is nothing but the classical concatenation function.

$$\frac{\vdash \vec{w} \in I \quad w = \gamma((\vec{x}_1, \vec{y}_1) \cdot ... \cdot (\vec{x}_i, \vec{y}_i))}{\vdash \gamma((\vec{x}_1, \vec{y}_1) \cdot ... \cdot (\vec{x}_i, \vec{y}_i)) \in \mathfrak{f}_P(I)} , \tag{4.39}$$

$$\frac{\gamma((\vec{x}_1, \vec{y}_1) \cdot ... \cdot (\vec{x}_i, \vec{y}_i)) = \gamma((\vec{x}'_1, \vec{y}'_1) \cdot ... \cdot (\vec{x}'_j, \vec{y}'_j)) \quad \vdash \gamma((\vec{x}'_1, \vec{y}'_1) \cdot ... \cdot (\vec{x}'_j, \vec{y}'_j)) \in \mathfrak{f}_P(I)}{\vdash \gamma((\vec{x}_1, \vec{y}_1) \cdot ... \cdot (\vec{x}_i, \vec{y}_i)) \in \mathfrak{f}_P(I)} \tag{4.40}$$

Keep in mind that these decompositions are by no means unique, not even the maximal ones! So to accommodate this extension in the general notion of pre-theories, we have to extend our language-theoretic structures; we leave this implicit, as it is nothing but an exercise in formalization, without being of any immediate usage for our purposes. Furthermore, of course we need the rules to infer sets, as above:

$$\frac{\vdash \gamma((\vec{w}_1, \vec{v}_1) \cdot (\vec{x}_1, \vec{y}_1) \cdot (\vec{w}_2, \vec{v}_2)) \in \mathfrak{f}_P(I) \quad ... \quad \vdash \gamma((\vec{w}_1, \vec{v}_1) \cdot (\vec{x}_i, \vec{y}_i) \cdot (\vec{w}_2, \vec{v}_2)) \subseteq \mathfrak{f}_P(I)}{\vdash \gamma((\vec{w}_1, \vec{v}_1) \cdot \{(\vec{x}_1, \vec{y}_1), ..., (\vec{x}_1, \vec{y}_1)\} \cdot (\vec{w}_2, \vec{v}_2)) \in \mathfrak{f}_P(I)} \tag{4.41}$$

And in addition the rules to go back from sets of terms to terms, which are completely parallel (check $\mathcal{C}P1, \mathcal{C}P1$). Moreover, we need the inferences for concepts; but luckily, these look exactly like the ones for $\mathcal{C}P1, \mathcal{C}P1$, because they only refer to sets, not to underlying entities; so they can be taken over without change. We will denote this set of (meta-)rules by $\mathcal{C}_n\mathfrak{g}$ for n-tuples, so that this way, we get the pre-theories $(\mathcal{C}_n\mathfrak{g}, \mathcal{C}_nP1), (\mathcal{C}_n\mathfrak{g}, \mathcal{C}_nPr)$. We will not further scrutinize the properties of this generalization; we just mention the following: all constructions working set-theoretically also work in this case without loss of generality, such as upward normality using p and monotonicity using the powerset construction. Properties relying on language-theoretic constructions are beyond reach with our current methods. An issue we have to settle at this point, however, is the complexity of the class of languages induced by $(\mathcal{C}_n\mathfrak{g}, \mathcal{C}_nPr)$. We can provide an upper bound as follows:

Theorem 114 *For any finite language I, $\mathcal{C}_n\mathfrak{g}_{\mathcal{C}_nPr}(I)$, $\mathcal{C}_n\mathfrak{g}_{\mathcal{C}_nPr}(I)$ are n-multiple context-free languages.*

For the (somewhat tedious) definitions of multiple context-free grammars and languages, see [63], also [38]). We only prove the claim for $\mathcal{C}_nPr$, for $\mathcal{C}_nP1$ it is exactly the same. In the proof, we write $\vec{w}$ for strings, $\overline{w}$ for tuples of strings, and $\overline{x}$ for tuples of variables. Note that for variable tuples, we have the same convention as for string tuples: $(x_1, x_2) \cdot (x_3, x_4) = (x_1x_3, x_2x_4)$.

Proof. We make the usual grammar construction: for each $X \in SCL_n(I)$, we grant us a non-terminal N_X; We add have $N_X(\vec{w}_1, ..., \vec{w}_i) \Leftarrow$ if and only

if $(\vec{w}_1, ..., \vec{w}_i) \in X$; and we have rules $X(\overline{x}_1 \cdot ... \cdot \overline{x}_i \Leftarrow Y_1(\overline{x}_1), ..., Y_n(\overline{x}_n)$ iff $Y_1 \circ ... \circ Y_n \leq X$. Finally, we add the rule $S(\gamma(\overline{x})) \Leftarrow N_{\mathcal{C}(I)}(\overline{x})$, which reduces tuples to strings and gives us the language the grammar generates. Call this (preliminary) grammar PG_i^I, where i refers to the tuple size. It can be easily seen that we have $L(PG_i^I) = I$, and moreover, any string can be derived in a large number of ways according to the concepts to which it belongs.

Now we simply add rules $N_W(\overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3) \Leftarrow N_X(\overline{x}_1)N_W(\overline{x}_2)N_Y(\overline{x}_3)$ if and only if $(W, X \circ W \circ Y) \in \mathcal{C}Pr(I)$. Call the resulting grammar G_n^I; this obviously is an MCFG.

As it might not be immediately clear that G_i^I does the job, consider the following. Checking inclusions in both direction would be very tedious, but luckily, there is a simpler way: as all our (non-terminal) MCFG-rules have the form $N_W(x_1y_1z_1, ..., x_ny_nz_n) \Leftarrow N_X(x_1, ..., x_n)N_W(y_1, ...y_n)N_Y(z_1, ..., z_n)$, we can read it as a simple context-free grammar, with the only difference that it generates terms of $(\Sigma^*)^n$ instead of Σ^*, but where juxtaposition is interpreted as normal concatenation in $((\Sigma^*)^n, \cdot)$ (note that this does not hold in general, only for the grammars we construct)! So we obtain the proof from theorem 20 (and lemma 109) and the following considerations: we have $\vec{w} \in \mathcal{C}_n\mathfrak{g}_{\mathcal{C}_nPr}(I)$ if and only if there is $\overline{v}$ such that $\gamma(\overline{v}) = \vec{w}$ and $N_{\mathcal{C}(I)}(\overline{v})$ is derivable; this holds if and only if $\vec{w} \in L(G_n^I)$. □

So we get a clear upper bound for the complexity of languages induced by $\mathcal{C}_nPr$. The question of lower bounds can be answered as follows:

Theorem 115 *For every $n \in \mathbb{N}$, there is an $m \in \mathbb{N}$ and a finite language I, such that $\mathfrak{g}_{\mathcal{C}_mPr}(I)$ is not an n-MCFL.*

Proof. Fix an $n \in \mathbb{N}$. We now take the language $I := \{a_1...a_m, a_1b_1a_2...b_{m-1}a_m\}$, where $m > 2n$. Then we obtain $\mathfrak{g}_{\mathcal{C}_mPr}(I) = \{a_1(b_1)^ia_2...a_{m-1}(b_{m-1})^ia_m : i \in \mathbb{N}\}$. This is not an n-MCFL. □

So we here have a pre-theory which brings us into the realm of what is known as mild context-sensitivity. Most linguists consider this to be the class of languages which contains all possible natural languages (or formal models thereof), though there is a lively discussion on that issue. So could we just be happy with this result? In my view we cannot; because if we stay with these results, then the fact that natural languages are mildly context sensitive is a *methodological universal*; and we want it to be an *empirical fact*, at least partially; in our terminology: we would be satisfied if the mild context-sensitivity would follow from some reasonable pre-theory $(\mathfrak{f}, P)$ as a property *modulo* $(\mathfrak{f}, P)$. But for this to be the case, the pre-theory must necessarily induce languages which are not mildly context-sensitive! So there is still plenty of work to do.

4.11 Transformational Pre-Theories

4.11.1 Ontological Questions

So far, we have only used the mechanism of substitution, even though we lifted it from simple strings to sets of strings. In a sense, this is the "structuralist heritage", as substitution was the main tool of structuralism. However, pre-theories are by no means necessarily restricted to the mechanism of substitution, as we will show in this chapter: regarding the techniques and functions we can

use, there are virtually no restrictions. The main question, to which we have already pointed before, is rather which operations on strings are "linguistically meaningful". This is a difficult question, to which the answer will be rather a matter of taste than one of conclusive argument, and we will be lucky enough if we yield a very broad agreement.

This question also has a long history; this relieves us from discussing it extensively, because we think the main arguments have been exchanged, and our focus is very different: for us, the main question is whether a technique yields satisfying results in a purely mathematical fashion; whether it is linguistically meaningful is of secondary importance. We will however shortly sketch in how far this history is relevant for us. The switch from structuralism to the generative paradigm proceeded along several "dimensions": firstly it was ontological one, as scholars went from considering language as an extensional, almost physical object to considering it as a cognitive capacity (see [31]). A second switch was in the formal methods linguists used: the main method of structuralists was substitution, whereas generativists used the more powerful techniques of phrase structure grammars and transformations. This change in methods in turn came with a change in methodology: while methods of generativists became much richer, they gave up on the strict structuralist methodology of gaining linguistic insight; finding the "correct description" was more a matter of linguistic intuition than the application of a rigid methodology. This switch in methodology seems to be strongly related to the more elaborate methods generativists used, as they make it much harder to formulate a precise methodology for gaining linguistic insights. Though this correlation is fairly obvious, I do not know whether it has been acknowledged explicitly. Be that as it may: one of the points of this section is to try to untighten this correlation, and the question is: can we keep up our strict methodology while enriching the techniques we use? Of course we are not doing linguistics, but rather metalinguistics, but the problems are very similar. We will put to use classes of functions within the framework of linguistic metatheory. We formulate a precise methodology for handling transformation-style functions in a language-theoretic context, where "transformation style" should be read in the very broad sense of any functions beyond mere substitution. As we will see, to make this work, we *can* use functions beyond substitutions, but we have to make very essential restrictions on them.

Let us return to the first question, namely the linguistic meaning of language-theoretic observations. Everyone will agree that substitution is in fact meaningful for linguistics, as it seems to be the most innocent of all of the linguists tools. (Though maybe not in the somewhat excessive way in which we have used it up to this point.) The main theme of this chapter is the following: If we decide to go beyond the concept of substitution, we open up a whole new world of possibilities. On the downside, it seems to me that there will be broad agreement that most of these possibilities are "linguistically meaningless", because they will be much too liberal, they will allow to project patterns to which we would never attribute any linguistic relevance. So the main goal is here to find reasonable restrictions of transformational pre-theories, which yet are powerful enough to provide us some substantial gain. One main source of inspiration, as the title suggests, will be transformational syntax, both for the things we have to allow, and things we must not allow.

But as we will see, the issue of allowing certain operations is by no means only of linguistic nature, it also has a mathematical content: allowing larger

classes of functions to make inferences does not necessarily result in larger classes of induced languages; quite the contrary can be the case. This (not being very deep) paradox will be the main motivation for us to consider more restricted classes of functions.

Let us say a pre-theory is simple transformational, if it is structural and based on strings. Our ontology thus contains strings, and in addition it contains unary functions on strings. Our analogical maps map finite languages onto sets of functions, that is, we have $P : \wp(\Sigma^*) \to (\Sigma^*)^{\Sigma^*}$. Inferences have broadly the form:

$$\frac{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}_P(I) \quad f \in P(I)}{\vdash \vec{w}(f(\vec{x}))\vec{v} \in \mathfrak{f}_P(I)} \tag{4.42}$$

Regarding the above scheme, note that we are quite close to the substitutional structural pre-theories on strings, in fact, these appear to be special cases of the transformational pre-theories. Therefore, there is obviously no reason to consider the non-structural case, because we will get the same undecidability results. This scheme needs however some explanation; in particular, there are two ways to read it. These two readings are as follows: *reading 1*: $\vec{w}(f(\vec{x}))\vec{v}$ is just our abstract variable notation for a simple *string*, where $(f(\vec{x}))$ denotes the *value* of $f(\vec{x})$. So for $f(\vec{x}) = \vec{y}$, we thus just use $f(\vec{x})$ instead of $\vec{y}$, because in the general case we have to somehow correlate the two. *Reading 2*: we literally read $f(\vec{x})$ as a *term*. We thereby get a new distinction of weak and strong language: for the weak language, we need to "spell this term out", that is, replace it by its value, because we want strings rather than terms. But for the strong language, we can make inferences on simple strings, or on terms, and both of them make sense. Call these two options, in analogy with some linguistic usage, "early spell-out" and "late spell-out".

It is easy to see that there can be a proper difference between the two. For example, assume $f(\vec{x}) = \vec{y}\vec{z}\vec{y}'$. In the early spell-out, we can now apply some function to $\vec{z}$. In late spell-out, this is impossible (in general): we do not see this string, until we spell-out, and then it is too late. In particular, note that in LAN spell-out, we always evaluate "inside out"; that is, if we derive a term $f_1(...f_i(\vec{x})...)$, then we have to apply first f_i, ... and then f_1; otherwise functions are undefined; this even though f_1 was introduced first in the derivation, and f_i last. In early spell-out, it is quite the contrary: whatever is introduced first, is applied first. Both options seem to be interesting for some, and problematic for other reasons; so we will consider them both with some care.

In favor of "early spell-out" we have the following: there is no "hidden layer". So we do not apply functions to strings we do not really see. For example, one can verify that $f(\vec{x})$ is be well-defined before introducing it. In late spell-out, on the other side, one does not immediately know what $f(\vec{x})$ yields, so it is unclear whether $f(f(\vec{x}))$ is even defined. On the other side, if we assume $f(\vec{x})$ to be a variable in the sense of placeholder for a string, where $f(\vec{x}) = y$, there is no problem.

In favor of late spell-out, let us recall that the main motivation for our functional analogies was to provide some sort of recursive inferences beyond substitution. However, with early spell-out this not always happens as we would expect. Let us illustrate this with an example. Assume we have a function $f(a^n) = a^{2n}$. In a finite language we might have the analogy $a \approx_I^P f(a)$ in some

analogical map P. Now if we have just a single word a as premise, late spell out yields the language $\{a^{2^n} : n \in \mathbb{N}\}$. Immediate spell out yields the language a^*, because the possible inputs of f are always the single as!

This is clearly not what we intend, and this simple example already provides us with the main argument for late spell out: as the analogies are always based on finite languages, the early spell out would thus have the consequence that we always instantiate the functions in one of a finite number of ways. Put differently: a consequence of early spell-out is that our functions *effectively* are finite in a set-theoretic sense; they are always instantiated on a finite number of arguments. But this is not a reasonable requirement to us, as one main motivation for introducing functions into our ontology are patters like duplication. To capture these, we have to make sure that functions apply on larger and larger arguments.

This is for us the main argument for preferring inference on terms (late spell out), but as we will see, there are also drawbacks, which lead us to considering both. But first of all, we have to make the notation unambiguous. We have said that there is a necessary distinction between weak and strong language, but now in a different way: the strong language consists of a set of terms over strings and functions from strings to strings; the weak language consists of a set of strings. One is mapped to the other, if each atomic function term is substituted by its value. We define this as follows.

Let Σ be an alphabet, $\mathcal{F}_\Sigma$ be a set of functions $f : \Sigma^* \to \Sigma^*$.

1. If $\vec{w} \in \Sigma^*$, then $\vec{w} \in Term(\mathcal{F}, \Sigma)$.
2. If $t \in Term(\mathcal{F}, \Sigma)$, $f \in \mathcal{F}_\Sigma$, then $f(t) \in Term(\mathcal{F}, \Sigma)$.
3. If $t, t' \in Term(\mathcal{F}, \Sigma)$, then $tt' \in Term(\mathcal{F}, \Sigma)$.

So let $t \in Term(\mathcal{F}, \Sigma)$ be a term. Its value $\|t\|$ is defined as follows:

1. If $t = \vec{w} : \vec{w} \in \Sigma^*$, then $\|\vec{w}\| = \vec{w}$.
2. If $t = f(t')$, then $\|f(t')\| = val(f(\|t'\|))$, where $val(f(\vec{x}))$ is the value of the function.
3. If $t = t't''$, then $\|t't''\| = \|t'\|\|t''\|$.

So we take the convention that in an inference scheme, $f(\vec{x})$ is a term rather the value of the function; and by $\|f(\vec{x})\|$ we denote its *value* $f(\vec{x})$. This allows us to present the unambiguous inference rule for "early spell-out":

$$\frac{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}_P(I) \quad f \in P(I) \quad \|f(\vec{x})\| = \vec{y}}{\vdash \vec{w}\vec{y}\vec{v} \in \mathfrak{f}_P(I)I} \tag{4.43}$$

Conversely, the late spell-out scheme would look as before; but we have to be aware that context consist of terms rather than strings:

$$\frac{\vdash t\vec{x}t' \in \mathfrak{f}_P(I) \quad f \in P(I)}{\vdash tf(\vec{x})t' \in \mathfrak{f}_P(I)I} \tag{4.44}$$

We will denote the two variants of inference rules with $\mathfrak{g}^{early}, \mathfrak{g}^{late}$ respectively. So we have settled the first ontological issue. The next major question is the following: given a transformational pre-theory P, $I \subseteq \Sigma^*$ a finite language, we assume that it maps a language onto a set of functions, so it gives a map

$P : \wp(\Sigma^*) \to (\Sigma^*)^{\Sigma^*}$. Without further restrictions, this might well be an infinite set! This might be problematic given our requirements that all procedures be finitary. But actually, for late spell-out this is entirely unproblematic: we just need a finite specification of the set, and thereby we can derive the term-language without any further computation: we do not actually have to calculate all the functions or even have the slightest idea what they look like. When we then compute the value of terms, we have to compute the functions, but even here, things are unproblematic: we just have to compute their values on a certain, finite set of points, and as long as functions are computable, this is no problem. The same holds for the early spell-out, except that we change order of things. But there is a further simplification we can perform: in fact, we do not to distinguish between different function which do the same thing on some input; therefore, given a transformational pre-theory P, we can define $P_R(I) := \bigcup\{|f| : f \in P(I)\}$, where by $|f|$ we denote the graph of a functions, that is, its set-theoretic interpretation. In the early spell out, we can replace the entire set of functions with the relation $P_R(I)$, and make analogies according to these pairs. Note however that if $P(I)$ is infinite, even if all $f \in P_R(I)$ are computable, the question whether $(\vec{w}, \vec{v}) \in P_R(I)$ might turn out to be undecidable. Still, there are no difficulties which cannot be overcome.

4.11.2 Detour: an Alternative Scheme

There is a choice we have made, and which we should explain. In the above scheme we have assumed that a pre-theory maps a language directly onto a set of functions, and that these functions are universally applicable to any string in $\mathit{fact}(I)$ in inferences. An equivalent way of expressing the same thing would be:

$$\frac{\vdash tyt' \in \mathfrak{f}_P(I) \quad \forall \vec{x} \in \mathit{fact}(I) : \vec{x} \Leftarrow_I^P f(\vec{x})}{\vdash t(f(\vec{y}))t' \in \mathfrak{f}_P(I)} \tag{4.45}$$

Here we make the universal quantification explicit, which was implicit in the first scheme. We allow an analogy f only if for all $\vec{x} \in \mathit{fact}(I)$ it satisfies certain requirements; as gain, we can apply it universally to any substring. Now the question is: do we really need or even want this universal quantification? We could state an inference scheme as follows:

$$\frac{\vdash t\vec{x}t' \in \mathfrak{f}_P(I) \quad \vec{x} \Leftarrow_I^P f(\vec{x})}{\vdash t(f(\vec{x}))t' \in \mathfrak{f}_P(I)} \tag{4.46}$$

or in the early spell-out:

$$\frac{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}_P(I) \quad \vec{x} \Leftarrow_I^P f(\vec{x})}{\vdash \vec{w}(\|f(\vec{x})\|)\vec{v} \in \mathfrak{f}_P(I)} \tag{4.47}$$

In this case, we have no universal requirements for f - the only thing which matters is how it behaves on $\vec{x}$. As a consequence, we can then only apply it to $\vec{x}$. In this case, let $\mathcal{F}$ be a class of functions which our pre-theory allows. P does not map I on a set of functions, but rather on a set of pairs $(\vec{x}, f(\vec{x})) : \vec{x} \in \mathit{fact}(I), f \in F$. This means that our pre-theories have to determine both arguments and functions on the same time. We can however use the same reduction as above. Assume $P : \wp(\Sigma^*) \to (\Sigma^*)^{\Sigma^*}$. We can reduce this without any loss to a relation $P_R(I) = \{(\vec{x}, \|f(\vec{x})\|) : (\vec{x}, f(\vec{x}) \in P(I)\}$.

Under this formulation it is clear that this alternative scheme is much more liberal: assume we have $f \in P(I)$ in the first formulation. Then, under reasonable assumption, it has to satisfy certain requirement on *all* points in $fact(I)$; in the second formulation, we can also apply functions which satisfy constraints only on *some* points. Assume we have $f(\vec{x}) = \vec{y}$, where $\vec{y} \in fact(I)$; but $f(\vec{y})$ is "unjustifiable", that is, f does something with $\vec{y}$ which can never allow us to build an analogy $(\vec{y}, \|f(\vec{y})\|)$. Then in the first approach, we would not have $f \in P(I)$; But in the second approach (using late spell-out), we might derive

$$\frac{\dfrac{\vdash \vec{w}\vec{x}\vec{v} \in \mathfrak{f}_P(I) \quad \vec{x} \approx_P^I f(\vec{x})}{\vdash \vec{w}f(\vec{x})\vec{v} \in \mathfrak{f}_P(I)} \qquad \vec{x} \approx_P^I f(\vec{x})}{\vdash \vec{w}f(f(\vec{x}))\vec{v} \in \mathfrak{f}_P(I)} \tag{4.48}$$

Effectively, we can apply f to $\vec{y}$, though $\vec{y} \in fact(I)$ and there is no justification such as $\vec{y} \approx \|f(\vec{y})\|$. So let us fix conventions. We denote the standard scheme $\mathfrak{g}_F$, the alternative scheme $\overline{\mathfrak{g}}_F$, with the variants $\mathfrak{g}_F^{early}, \mathfrak{g}_F^{late}$ etc. So for any transformational analogical map we get 4 pre-theories. Assume we have a class of functions $\mathcal{F}$, used for all for schemes, and a pre-theory $P : \wp(\Sigma^*) \to \wp(\Sigma^* \times \Sigma^*)$. We expand P to $\mathcal{F}P$ as follows: we say $(\vec{x}, f(\vec{x})) \in \mathcal{F}P(I)$ if $(\vec{x}, \|f(\vec{x})\|) \in P(I)$; to obtain the inference scheme $\mathfrak{g}_{\mathcal{F}}$, we need an additional inference rule of the form

$$\frac{\forall \vec{x} \in fact(I) : x \approx_I^{\mathcal{F}P} f(\vec{x})}{f \in \mathcal{F}P(I)} \tag{4.49}$$

On the other side, for $\overline{\mathfrak{g}}_{\mathcal{F}}$ we only need

$$\frac{x \approx_I^{\mathcal{F}P} f(\vec{x})}{x \Leftarrow_I^{\mathcal{F}P} f(\vec{x})} \tag{4.50}$$

By what we have said it can be seen that $\overline{\mathfrak{g}}_{\mathcal{F}}$ is more liberal than $\mathfrak{g}_{\mathcal{F}}$. However, we get a proper set theoretic inclusion only in the case of late spell-out, because in early spell-out, functions can be applied to substrings which do not even figure in $fact(I)$, and we know nothing about what happens in this case.

Lemma 116 *For all finite languages I, $\mathfrak{g}_{\mathcal{F}P}^{late}(I) \subseteq \overline{\mathfrak{g}}_{\mathcal{F}P}^{late}(I)$.*

This discussion was informal, because it is simply a side note on an option we have not taken. For us, this is the main reason to stick to the "standard" mode of inference: the problem of the transformational pre-theories is in general that they are too liberal and allow too many inferences (cf. the subsequent results). So among the two options, for us it seems the reasonable choice to take the more restrictive one. So in the sequel, we will use neither of the two schemes presented in this section; however, we should be aware of their existence, at the very least because they make it more clear to us what is peculiar about the choice we made.

4.11.3 Legitimate Functions

The next main question we investigate is the following: which functions should we allow for analogies? The following requirement is immediate: all functions

we use must be computable. Otherwise pre-theories become undecidable. A second requirement is the following: all our functions should be **alphabetically conservative**; that is: for a pre-theory P using functions, it should hold that if $f \in P(I)$, $I \subseteq \Sigma^*$, then $f \in (\Sigma^*)^{\Sigma^*}$. There is another basic requirement, which is already implicit in the above requirements: we do not allow for partial functions. This is important, as partial functions can create a lot of problems: if we use terms for analogies, we never really know whether the terms are actually defined. On the other side, the fact that we consider finite "datasets" suggests that our functions may be undefined for maybe infinitely many inputs, just because they are underspecified from our analogies. To remedy, I would propose the following simple solution: given a partial function $f : \Sigma^* \to \Sigma^*$, we define the canonical completion $\hat{f} : \Sigma^* \to \Sigma^*$ by $\hat{f}(x) = f(\vec{x})$ if $f(\vec{x})$ is defined, and $\hat{f}(\vec{x}) = \vec{x}$ otherwise. So can use canonical completions, in order to make sure that all our functions are complete; and we will in the sequel always assume our functions to be complete.

The good thing of our major analogical maps so far is: they can be transferred with some modification to the transformational approach. For example, we can devise $FP1$ by: $\vec{x} \approx_I^{FP1} f(\vec{x})$ iff

1. if $\vec{x} \sqsubseteq \|f(\vec{x})\|$, then $\vec{w}\|f(\vec{x})\|\vec{v} \in I \Rightarrow \vec{w}\vec{x}\vec{v} \in I$; if $f(\vec{x}) \sqsubseteq \vec{x}$, then $\vec{w}\vec{x}\vec{v} \in I \Rightarrow \vec{w}\|f(\vec{x})\|\vec{v} \in I$.

2. otherwise, $\vec{w}\vec{x}\vec{v} \in I \Leftrightarrow \vec{w}\|f(\vec{x})\|\vec{v} \in I$.

Note that the condition $\vec{x} \sqsubseteq f(\vec{x})$ of $P1$ has been changed, as it is rather a particular case than a pre-condition. In our more general case this is neither necessary nor sufficient, so there is no reason to keep this condition. On the other side, we need the new conditional statement, in order to make sure there is a sufficient similarity in distribution. If $\vec{x} \sqsubseteq \|f(\vec{x})\|$ holds, then we cannot have $\vec{x} \sim_I \|f(\vec{x})\|$ for a finite language I; but in case we have $\vec{x} \not\sqsubseteq \|f(\vec{x})\|$, we *can* have $\vec{x} \sim_I \|f(\vec{x})\|$, and so in my view there is no reason not to require it.

We can also adapt the more restrictive Pr, defining FPr as follows: $\vec{x} \approx_I^{FPr} f(\vec{x})$ iff

1. if $\|f(\vec{x})\| = \vec{x}_1\vec{x}\vec{x}_2$, then $\vec{w}\|f(\vec{x})\|\vec{v} \in I \Rightarrow \vec{w}\vec{x}\vec{v} \in I$; and if $(\vec{w}, \vec{v}) \neq (\vec{w}'\vec{x}_1, \vec{x}_2\vec{v}')$, then $\vec{w}\vec{x}\vec{v} \in I \Rightarrow \vec{w}\|f(\vec{x})\|\vec{v} \in I$. If $\vec{x} = \vec{x}_1\|f(\vec{x})\|\vec{x}_2$, then $\vec{w}\vec{x}\vec{v} \in I \Rightarrow \vec{w}\|f(\vec{x})\|\vec{v} \in I$; and if $(\vec{w}, \vec{v}) \neq (\vec{w}'\vec{x}_1, \vec{x}_2\vec{v}')$, then $\vec{w}\|f(\vec{x})\|\vec{v} \in I \Rightarrow \vec{w}\vec{x}\vec{v} \in I$.

2. otherwise, $\vec{w}\vec{x}\vec{v} \in I \Leftrightarrow \vec{w}\|f(\vec{x})\|\vec{v} \in I$.

For both, we have the obvious restriction that f be computable, alphabetically conservative and total. Call such functions **legitimate**. We could read the F in $FP1, FPr$ as representing these legitimate functions. We get the pre-theories $(\mathfrak{g}^{late}, FP1), (\mathfrak{g}^{late}, FPr)$, where $\mathfrak{g}^{late}$ contains the standard inferences and the scheme

$$\frac{\forall \vec{x} \in \mathit{fact}(I) : x \approx_I^{FP} f(\vec{x})}{f \in FP(I)} \tag{4.51}$$

A point to note is that there are infinitely many legitimate functions which behave in exactly the same way on $\mathit{fact}(I)$, which are thus indistinguishable from

the point of view of possible analogies. So a consequence is: we will get infinitely many distinct premises of the form $f \in FP1(I)$, and we will be able to derive many strings.

One could think that this makes the unrestricted transformational pre-theories quite trivial, because they are completely arbitrary: if f is used in an analogy, we must specify the values of f on the set of factors of I. What f does on any other strings is completely unspecified, and could in fact have no relation or similarity to its behavior on $fact(I)$. This is however not exactly true: the conditions above ensure for example, that if $\vec{x} \not\sqsubseteq f(\vec{x})$, $\vec{x} \approx_I^{FPr} f(\vec{x})$, then if $\vec{x} \in fact(I)$, then also $f(\vec{x}) \in fact(I)$. There is thus no way to move out of $fact(I)$ by means of application of functions, as long as we do not satisfy $\vec{x} \sqsubseteq f(\vec{x})$! However, once we have "moved out of" $fact(I)$, that is, derived a term $f(...(f(\vec{x}))...)$ such that $\|f(...(f(\vec{x}))...)\| \notin fact(I)$, the values of f are completely "free", so to speak. We need the substring condition to move out of $fact(I)$, but once we have a term $f(...f(\vec{x})...) \notin fact(I)$, we can basically derive anything we want from it. This is because the values $\|f(\vec{y})\| : \vec{y} \notin fact(I)$ are completely irrelevant to the analogies we draw. This in turn has the following consequence: let Σ be an alphabet to which we restrict f, and $I \subseteq \Sigma^*$. Then we easily obtain:

Lemma 117 *Let $I \subseteq \Sigma^*$ be a finite language containing two words $\vec{x}, \vec{x}_1\vec{x}\vec{x}_2 \in I$ and satisfying $\vec{x} \leq_I \vec{x}_1\vec{x}\vec{x}_2$. Then $\mathfrak{g}^{late}_{FP1}(I) = \mathfrak{g}^{late}_{FPr}(I) = \Sigma^*$.*

Proof. Because we have $\vec{x}, \vec{x}_1\vec{x}\vec{x}_2$, we can devise a function f such that (1) $f(\vec{x}) = \vec{x}_1\vec{x}\vec{x}_2$, (2) $f((\vec{x}_1)^k\vec{x}(\vec{x}_2)^k) = (\vec{x}_1)^{k+1}\vec{x}(\vec{x}_2)^{k+1}$, if $(\vec{x}_1)^{k+1}\vec{x}(\vec{x}_2)^{k+1} \in fact(I)$, and (3) for all $y \in fact(I)$ such that $y \neq (\vec{x}_1)^k\vec{x}(\vec{x}_2)^k$ for all $k \in \mathbb{N}_0$, we put $f(\vec{y}) = \vec{y}$. This surely allows for an analogy, such that regardless of its behavior on other strings, we have $f \in FP1(I)$ and $FPr1(I)$.

We can order $\Sigma^* - fact(I)$ in a linear and well-founded fashion, and so get a bijection $i : \mathbb{N} \to \Sigma^* - fact$. Choose in particular a bijection i such that $i(1) = (\vec{x}_1)^k\vec{x}(\vec{x}_2)^k$ for the smallest k such that $(\vec{x}_1)^k\vec{x}(\vec{x}_2)^k \notin fact(I)$.

Now we complete the definition of f: (4) for all $\vec{y} \in \Sigma^* - fact(I)$, $n \in \mathbb{N}$, we put $f(i^{-1}(n)) = i^{-1}(n+1)$. This means, we can derive all strings in $\Sigma^* - fact(I)$ from $\vec{x}$. Thus we already have $\mathfrak{g}^{late}_{FP1}(I) = \mathfrak{g}^{late}_{FPr}(I) = \Sigma^*$, and by a single function $f \in F(P1)(I)$ and $FPr(I))$. □

This is obviously a very negative result, as it not only strongly contradicts what we think a pre-theory should do, but it trivializes the entire procedure. It is a good example of what we have to be aware of if we include functions into our ontology: if we have a large class of functions at our disposition, this does not mean that we have a large class of languages we induce - quite the contrary can be the case, as the above lemma shows. The reason is: we get too many analogies, and in the derived language, of course we have to take the union over the derivable strings.

In case the condition $\vec{x}, \vec{x}_1\vec{x}\vec{x}_2 \in I$, with $\vec{x} \leq_I \vec{x}_1\vec{x}\vec{x}_2$ is not satisfied, things are a bit more complicated. Nonetheless, we can get an easy generalization of the above result: if we have some words $\vec{w}\vec{x}\vec{v}, \vec{w}\vec{x}_1\vec{x}\vec{x}_2\vec{v} \in I$ with $\vec{x} \leq_I \vec{x}_1\vec{x}\vec{x}_2$, then we have $\vec{w}\Sigma^*\vec{v} \subseteq \mathfrak{g}^{late}_{FPr}(I) \cap \mathfrak{g}^{late}_{FP1}(I)$. The argument is essentially the same, but we obviously do not get an equality, as we do not know about other strings in the language. We can also easily derive the following lemma from what we have demonstrated so far:

Lemma 118 *(1) If* $P1(I) = \emptyset$*, then* $\mathfrak{g}^{late}_{FP1}(I) = I$*. (2) If* $Pr(I) = \emptyset$*, then* $\mathfrak{g}^{late}_{FPr}(I) = I$*.*

Note that this lemma shows the close relation of $P1, Pr$ on the one, and $FP1, FPr$ on the other hand.

Proof. We only show (1), because (2) can be shown in much the same way. Assume that $P1(I) = \emptyset$, a function $f \in FP1(I)$. Then by assumption, if $\vec{x} \in \mathit{fact}(I)$, $f \in FP1(I)$, we have $\vec{x} \not\sqsubseteq f(\vec{x}), f(\vec{x}) \not\sqsubseteq \vec{x}$. From $FP1$ it follows that $\vec{w}\|f(\vec{x})\|\vec{v} \in I \Leftrightarrow \vec{w}\vec{x}\vec{v} \in I$. As the $\vec{x}$ is arbitrary, the same holds for $\|f(\vec{x})\|$; so we have $\vec{w}\|f(\vec{x})\| \in I \Leftrightarrow \vec{w}\|f(f(\vec{x}))\|\vec{v} \in I$. So all strings we can derive are already in I. □

These results together give us the following:

Theorem 119 *For any finite language* I*,* $\mathfrak{g}^{late}_{FPr}(I)$ *and* $g_{FP1}(I)$ *are regular.*

Proof. We can show this by the set of premises I. Assume we have no $\vec{w} \in I$ with an $\vec{x}$ such that $\vec{w} = \vec{w}_1\vec{x}\vec{w}_2$, $\vec{x} \leq_I \vec{x}_1\vec{x}\vec{x}_2$. Then $\mathfrak{g}^{late}_{FP1(I)}(\vec{w})$ is finite by the last lemma. Otherwise, $\mathfrak{g}^{late}_{FP1(I)}(\vec{w}) = \vec{w}_1\Sigma^*\vec{w}_2$ for all (finitely many) decompositions.

This yields a finite union of finite languages and languages of the form $\vec{w}\Sigma^*\vec{v}$. This is a regular language. □

It is quite clear that this bound is not optimal, that is, it can be improved. As a matter of fact, we can see immediately that the claim can be strengthened to:

Theorem 120 *For any finite language* I*,* $\mathfrak{g}^{late}_{FPr}(I)$ *and* $\mathfrak{g}^{late}_{FP1}(I)$ *are star-free languages.*

The proof is exactly the same. Still this bound is not optimal, of course, but I do not see how it can be substantially improved, at least not involving any known class of languages. However, we also get another corollary which is rather positive:

Corollary 121 $\mathfrak{g}^{late}_{FP1}, \mathfrak{g}^{late}_{FPr}$ *are closed.*

Proof. For one direction, see lemma 118; if the conditions of this lemma are not satisfied, it is easy to see that we get an infinite language. □

So we see from these results that we induce only star-free languages, and this not despite, but rather because we allow for any recursive functions in $FP1(I)$ and $FPr(I)$. The main point why this is a restriction is that we must use all functions which satisfy the analogy condition, which are again countably many. So the only reasonable way to get more powerful pre-theories is to restrict the space of possible functions.

4.11.4 Opaque Functions, and Why They Will not Work

So which classes of functions form meaningful restrictions? To consider a first, very important class, take the rational functions, that is, functions computed by finite state transducers. These are quite restrictive, among other because they can be computed with finite resources and in linear time (see [70] for a proof). Nonetheless, this class is quite rich and allows to compute a lot of functions on

strings. So what if we reduce the space of legitimate functions to the rational functions? Still, the above results obtain for the following reason: it is clear that any restriction of a function to a finite domain is rational. Moreover, there is also a rational bijection $i : \mathbb{N} \to \Sigma^* - fact(I)$, even though it is less obvious: we simply choose a representation of $\mathbb{N}$ in base $|\Sigma|$; as $\Sigma^* - fact(I)$ is finite, we only need to perform the subtraction of a finite number in order to get the desired result, which can be easily done by finite-state means. So the above "trivialization" results already obtains for rational functions, and *a fortiori* for every larger class. The same argument as before obtains even for a smaller class, the so called regular or synchronous regular functions (see [60] for reference).

This means we have to look at classes which are not necessarily smaller, but which are by no means larger. What classes should we consider? Of course, there are many of classes which would allow to derive mathematically interesting results. But if we do not see any linguistic meaning in them, it would be quite sterile and off the point for this work scrutinizing them. We will therefore proceed as follows: first, we will provide an exact characterization of classes of functions which are too strong/too large for our purposes. Next, we show that there are some really interesting candidates regarding classes of functions. There are however also fundamental problems before we can put them to use. These lead to our next major pre-theory and next major change in ontology.

By $dom(f)$ we denote the domain of f. We say a function f is **proper infinite**, if there are infinitely many $x \in dom(f)$, such that $f(x) \neq x$. f is **finite**, if it is not proper infinite. So a finite function is not to be understood in the set-theoretical sense! But a finite function is the canonical completion of a function finite in the set-theoretic sense. A **class** $\mathcal{F}$ of functions is proper infinite, if there is a (finite or infinite) set $\{f_1, f_2, ...\} \subseteq \mathcal{F}$, and an infinite set $M \subseteq dom(f_1) \cap dom(f_2) \cap ...$, such that for every finite $N \subseteq M$, we have $f \in \{f_1, f_2, ...\}$ with $f(x) \neq x$ for all $x \in N$. Note that if $\mathcal{F}$ is proper infinite, it need not contain a proper infinite function; but then it has to contain an infinite set of functions. Conversely, however, any class containing a proper infinite function is proper infinite. The reason for this slightly complicated definition is that we are not so much interested in the behavior of single functions, but of possibly infinite sets of functions (satisfying certain criteria). The reason should be clear by the above considerations.

We distinguish between a function f and its graph $|f|$; so for $f : M \to N$, we have $|f| \subseteq M \times N$, $|f| = \{(m, f(m)) : m \in M\}$. We say a class of functions $\mathcal{F}$ is **opaque**, if it contains functions which are infinite in the set-theoretical sense, and if for every finite relation R, such that there is $f \in \mathcal{F}$ with $R \subseteq |f|$, and for each $Q \subseteq R$, we have a $f' \in \mathcal{F}$, with $Q \subseteq |f'|$, and $R - Q \not\subseteq |f'|$. So for each finite subset of a graph of a function, we have a function which behaves differently on this subset. One might say that opaqueness is a functional analogue of the notion of **infinite flexibility** of classes of languages (see [29]). The following is easy to see:

Lemma 122 *If $\mathcal{F}$ is opaque, it is proper infinite.*

Proof. Assume $\mathcal{F}$ is opaque. Assume furthermore, $\mathcal{F}$ does not contain a proper infinite function (otherwise the claim follows immediately). Then we have a function $f \in F$, $M \subseteq dom(f)$, such that $f(x) = x$ for all $x \in M$, and M is an infinite set. Then for each finite $N \subseteq M$, we have a function f' for which

$f(x) = f'(x)$ if and only if $x \in O \subseteq N$. We now take an increasing sequence $N_1, N_2, ...$, with $N_1 \subsetneq N_2 \subsetneq ...$, and put $O_i = N_i$. This way, we get a sequence of functions (f_i), where for all $x \in O_i$, we have $f_i(x) \neq f(x)$. We then have an infinite set $P := \bigcup_{i \in \mathbb{N}} O_i$, and for each finite $Q \subseteq P$, we have an $O_i \subseteq P$ such that $O_i \supseteq Q$, and thus f_i with $f(x) \neq x$ for all $x \in P$. □

The motivation behind these definitions is the following: if a class of functions is opaque, we cannot really do anything with it. The reason is: if we have a finite language, we only see a finite number of "instantiations" of the function. If our class of functions if opaque, then this finite number of instantiations does not tell us enough about the functions in question to identify a unique function; we have to take all the functions compatible with this finite fragment, which are infinitely many. So the whole idea of functions becomes quite problematic, because in inferences in a sense we throw them together to a relation.

On the other side, assume we work out some criteria to restrict the functions of a given opaque class and their application in analogies, that is, to prefer a function f_1 over f_2, even though they coincide on the finite set of factors we observe. This might be possible in principle, but note that it is exactly the same problem we address here in the first place: determining infinite sets from finite sets is basically the same as determining infinite functions from finite subsets of their graphs. I conclude (maybe prematurely) that this way does not lead to a solution for our problem, but rather into a new version of our old problem. So when we look for interesting classes of functions, we have to make sure these classes are not opaque. Note that the classes of functions we mentioned so far, such as rational or regular functions are all opaque, as is easy to show.

We now introduce the dual of opaqueness. A class of functions $\mathcal{F}$ is **transparent**, if for all $f \in \mathcal{F}$, there is a finite relation R, such that $R \subseteq |f|$, and if $f' \in \mathcal{F}$ and $R \subseteq |f'|$, it follows that $f' = f$. We say then that R is characteristic of f in $\mathcal{F}$. Note that being transparent is much stronger than not being opaque: a class of functions $\mathcal{F}$ is not opaque if there is at least one $f \in \mathcal{F}$ satisfying the transparency criterion, and a class $\mathcal{F}$ is not transparent if there is at least one $f \in \mathcal{F}$ satisfying the opaqueness criterion. So the two properties can be seen as opposite extremes.

4.11.5 Polynomial Functions

What we want are functions which are proper infinite, because finite functions do leave us within the realm of finite languages under the inference with "late spell-out":

Lemma 123 *Let $\mathcal{F}$ be a class of finite functions. Then for any transformational pre-theory $(\mathfrak{g}^{late}, \mathcal{F}P)$, finite language I, $\mathfrak{g}^{late}_{\mathcal{F}P}(I)$ is finite.*

Proof of this is immediate: we derive infinitely many terms; however, if our class of functions is finite, each term spells out to a finite number of strings. So how about the alternative scheme with early spell out? Here we get the following result concerning finite functions:

Lemma 124 *Let $\mathcal{F}$ be a class of finite functions. Then for any transformational pre-theory $(\mathfrak{g}^{early}, \mathcal{F}P)$, finite language I, $\mathfrak{g}^{early}_{\mathcal{F}P}(I)$ is a CFL.*

Proof. To see this, just take the grammar construction we used in the last section: every string $\vec{w} \in fact(I)$ is encoded by a nonterminal $N_{\vec{w}}$, and re-writes as $N_{\vec{w}} \to N_{\vec{v}_1 ... \vec{v}_i}$, for all $\vec{v}_1 ... \vec{v}_2 = f(\vec{w})$. As these are finitely many substrings, we have finitely many rules. □

So using finite functions is out of question. On the other side, we want function which are not opaque, or even better, transparent. This leaves still some options; but there is in fact a class of functions satisfying these requirements, which is well-known and well-studied in formal language theory. These are what we call **polynomials functions**, or simply polynomials over an alphabet Σ. These are functions $f : \Sigma^* \to \Sigma^*$, which have representations of the form

$$f(\mathsf{x}) = \vec{w}_1 \mathsf{x} \vec{w}_2 ... \vec{w}_{i-1} \mathsf{x} \vec{w}_i : \ i \geq 0, \tag{4.52}$$

where $\vec{w}_1, ..., \vec{w}_i \in \Sigma^*$, and $\mathsf{x} \notin \Sigma^*$ is a variable, such that for any $\vec{w} \in \Sigma^*$, we have

$$f(\vec{w}) = \vec{w}_1 \vec{w} \vec{w}_2 ... \vec{w}_{i-1} \vec{w} \vec{w}_i \tag{4.53}$$

We denote the above representation of f by $pol(f)$, its polynomial, which is a string over $(\Sigma \cup \{\mathsf{x}\})^*$. Note that polynomials are in principle well-defined for any input string over any alphabet; but we will assume that all functions come with a specified domain; and once its domain of a polynomial function is specified, so is its range. Denote the class of polynomial functions by $\mathcal{P}$. A function in this class thus copies the input string an arbitrary number of times (possibly 0), while putting any constant strings between the copies. It is easy to see that $\mathcal{P}$ is proper infinite. We can also show that this class is not opaque: There is only one function $f \in \mathcal{P}$ such that $f(\vec{w}) = \vec{w}\vec{w}, f(\vec{w}\vec{w}) = \vec{w}\vec{w}\vec{w}\vec{w}$ for any $\vec{w} \in \Sigma^*$. In fact, we can show that the class of polynomial functions is transparent:

Lemma 125 *$\mathcal{P}$ is transparent.*

Proof. Though this seems quite obvious, we need a bit of work to show this. Assume $f \in \mathcal{P}$. We have to find a relation $N \subseteq |f|$ characteristic of f in $\mathcal{P}$. What we first do is we choose two words $a, aa : a \in \Sigma$, and put $N = \{(a, f(a), (aa, f(aa)\}$. This allows us to uniquely identify the number of variable occurrences in the representation of f by a simple numerical argument. Now assume that $\Sigma^* = dom(f)$.

Case 1. Assume $|\Sigma| = 1$. Let $n(f)$ be the number of variables occurrences of $pol(f)$. Then if $n(f') = n(f)$, we have $f(a) = f'(a)$ if and only if $f = f'$, because they always output strings of the same length, and there is only one such string of a given length given a unary alphabet. Note, by the way, that the case where $|\Sigma| = 1$ is the only one where identical functions can have different representations. So for the case that $|\Sigma| = 1$, we might have $pol(f) \neq pol(f')$, but still if $f(a) = f'(a), f(aa) = f'(aa)$, we have $f(\vec{w}) = f'(\vec{w})$ for all $\vec{w} \in \Sigma^*$, and so all distinct functions can be distinguished by their behavior on a finite set $\{a, aa\}$.

Case 2. Conversely, assume there is $b \in \Sigma$, $b \neq a$. we can safely assume that $n(f) = n(f') = k$, otherwise we distinguish the two by $\{a, aa\}$. Now assume $f(a) = f'(b)$. We have $f(a) = \vec{w}_1 a \vec{w}_2 ... \vec{w}_k a \vec{w}_{k+1} = f'(a)$. Now we take the representations of f, f'. Assume the ith letter in $pol(f)$ is a constant; say c; if it is a constant in $pol(f')$, then it has to be c as well; if it is not in Σ, we

have the ith letter in $f'(a)$ different from the ith letter in $f'(b)$ - and so either $f'(a) \neq f(a)$ or $f(b) \neq f'(b)$. Now assume the ith letter in $pol(f)$ is a variable; then the ith letter in $f(a)$ is a, whereas in $f(b)$ is b; as $a \neq b$, either the ith letter in $pol(f')$ is also a variable, or we have either $f(a) \neq f'(a)$ or $f(b) \neq f'(b)$. Conversely, assume the ith letter in $pol(f)$ is a variable.

This shows that in the case of $|\Sigma| \geq 2$, $a \neq b$, from $f(a) = f'(a), f(aa) = f'(aa), f(b) = f'(b)$ it follows that $pol(f) = pol(f')$, and so $f = f'$. So all distinct functions can be distinguished by their behavior on a finite set $\{a, aa, b\}$. □

So we see that $\mathcal{P}$ is an adequate class of functions.

4.11.6 Inferences with Polynomials

Given this encouraging result, we will devise a pre-theory over $\mathcal{P}$. The definitions for $\mathcal{P}P1, \mathcal{P}Pr$ are as before, except for the fact that we have the additional requirement for $f \in \mathcal{P}P1(I)$, namely that $f \in \mathcal{P}$. But before we look at these pre-theories in particular, let us look what we can infer with the help of polynomial functions. For inferences, we now use the rules $\mathfrak{g}^{late}$:

$$\frac{\vdash t\vec{x}t' \in \mathfrak{f}_P(I) \quad f \in P(I)}{\vdash tf(\vec{x})t' \in \mathfrak{f}_P(I)}, \tag{4.54}$$

Here, the premise that $f \in \mathcal{P}$ is meant to be implicit in the choice of P. For example, assume we have a duplicating patterns as in $I_e := \{a, aa, aaaa\}$, and we have $f(\mathsf{x}) = \mathsf{xx}$, and $P(I) = \{f\}$. This looks very much like what we look for when we want to capture this pattern: we get to derive $f(a), f(f(a)), f(f(f(a)))$.... It is clear that this allows us to derive a language which is not semilinear. However, things are much more complicated than one would guess at the first sight. The surprising fact is that even in this case, we have $\mathfrak{g}_P^{late}(I) \neq \{a^{2^n} : n \in \mathbb{N}\}$. The reason is: we also get to derive terms as $f(f(a))a$, $f(f(a))af(a)a$. So if we do not get the language $\{a^{2^n} : n \in \mathbb{N}_0\}$ which we (probably) desire – but what language do we get? Call this language L_e. It takes some patience to recognize its true face; for $n \leq 20$, we can derive all a^n. We can however find a number i such that $a^i \notin L_e$. We find it as follows. First of all, we can transform the terms resulting from derivations into arithmetic terms denoting their length; concatenation is interpreted in addition, as concatenation of strings over a singleton alphabet is commutative; and $f(\vec{x})$ is transformed into $x + x$. So what we look for is a number $k \in \mathbb{N}$, such that there is no equation

1. $2^{n_1} + 2^{n_2} + 2^{n_3} + 2^{n_4} = k$,
2. $2^{n_1} + 2^{n_2} + 2^{n_3} = k$
3. $2^{n_1} + 2^{n_2} = k$
4. $2^{n_1} = k$
5. $2^{n_1}3 + 2^{n_2} = k$,
6. $2^{n_1}3 = k$,

etc. corresponding to a term having a a solution with $n_1, n_2, n_3, n_4 \in \mathbb{N}_0$. Each equation corresponds to a set of derivations in the strong language; for example, the first equation corresponds to all derivations of the form

$f(...f(a)..)f(...f(a)..)f(...f(a)..)f(...f(a)..)$; in equation 5, we derive $f(...f(aaa)..a)f(...f(a)...)$ etc.

We first take the number 3 as the smallest natural number not in $\{2^n : n \in \mathbb{N}_0\}$. Next, take an $l \in \{2^n : n \in \mathbb{N}_0\}$ such that the difference with the 2^{n+1} is larger than 3. We choose the 8. We now iterate this, taking next the 32, and the 128. We add this up, and get 171. My claim is: there is no solution for $k = 171$ for any of the above equations; that means, $a^{171} \notin L$. From construction it follows that there is no solution for the first equation; the third and forth can be subsumed under the first. There is one additional possible derivation we have to check: it might be that a^{171} has the form $f^i(aaa)f^j(a)$; thus we have $2^{n_1}3 + 2^{n_2} = 171$. Again, it can be easily checked manually that there is no solution for $n_1, n_2 \in \mathbb{N}_0$.

From this, it can be easily inferred that the "gaps" in the language, that is, the proportion of strings $a^n \notin L$ become larger for $n \to \infty$, so we have L becoming more and more sparse in an exponential fashion. From this it follows that L is not semilinear. This shows that we are in fact expressive in going beyond certain classes. On the downside, we cannot claim that this approach behaves in a way we would want it to behave. To put it in an intuitive fashion: we are lacking "control"; we cannot determine the contexts in which we ought to apply functions/analogies; and obviously this is what we need: we only want to apply f to entire strings in L, not to its substrings. As a side note, note also that the "early spell-out" and the alternative scheme do not change this!

There is still another substantial problem with $\mathcal{P}$. Assume we have a language I with a duplication pattern, but some other unrelated strings and patterns. By assumption, polynomial functions are total. However, if $f(\mathsf{x}) = \mathsf{xx}$ and we have $f \in P(I)$, then it follows we can really duplicate *everything*. Polynomial functions cannot distinguish between different input strings. We could remedy this by allowing functions to use different polynomials depending on their input - but in the very moment we allow this, we lose transparency. So to get functions to work seems to be difficult.

4.11.7 Polynomial Pre-Theories

So far we have looked at classes of functions and found $\mathcal{P}$ very promising. However, our standard inference rules $\mathfrak{g}^{late}$ lead us away from what we actually desired, and we actually have no good solution for this. We now show that exactly the same problem strikes if we devise our analogical maps, in a way that we cannot say how a reasonable pre-theory for $\mathcal{P}$ should look like. Let us consider $\mathcal{P}P1$ and $\mathcal{P}Pr$, which are like $\mathcal{F}P1, \mathcal{F}Pr$ above, except for the fact that we exchange the class of legitimate functions, and we put the universality requirement into the analogical map:

$\mathcal{P}P1$ is defined by: $f \in \mathcal{P}P1(I)$ iff

1. $f \in \mathcal{P}$;
2. for all $\vec{x} \in \Sigma^*$, if $\vec{x} \sqsubseteq \|f(\vec{x})\|$, then $\vec{x} \leq_I \|f(\vec{x})\|$;
3. for all $\vec{x} \in \Sigma^*$, if $\|f(\vec{x})\| \sqsubseteq \vec{x}$, then $\|f(\vec{x})\| \leq \vec{x}$.
4. for all $\vec{x} \in \Sigma^*$, if neither of the two holds, then $\vec{x} \sim_I \|f(\vec{x})\|$

Note that cases 3.,4. occur only if if we have the case of a constant function $f(\mathsf{x}) = \vec{w}$ for $\vec{w} \in \Sigma^*$. However, constant functions do not allow to derive anything new, in virtue of their being constant, the condition and our inference rules; so we will just ignore them in the sequel. This approach can in principle work. However, it probably does not exactly do what we want it to do: take the language $I_1 := \{a, aa\}$. Here we have three non-trivial functions $f_1, f_2, f_3 \in \mathcal{P}P1(I_1)$, where $f_1(\mathsf{x}) = \mathsf{xx}, f_2(\mathsf{x}) = a\mathsf{x}, f_3(\mathsf{x}) = \mathsf{x}a$. So f_1 captures a duplication scheme, but it is "covered" by f_2, f_3, in the sense that we derive the language a^*. This is not bad all together, as we might say that I_1 does not yet clearly show a duplicating pattern. But now take $I_e := \{a, aa, aaaa\}$. What happens now? Surprisingly, we have $\mathcal{P}P1(I_2) = \emptyset$ (apart from constant functions and the identity function)! Why is that? We can easily check manually that all f_1, f_2, f_3 are out of question; we only demonstrate this for f_1. Assume we have $f_1 \in \mathcal{P}P1(I_2)$. Then $\vec{w}_1 \| f_1(\vec{w}_2) \| \vec{w}_3 \in I_2 \Rightarrow \vec{w}_1\vec{w}_2\vec{w}_3 \in I_2$. Now we have $a \| f_1(a) \| a = aaaa \in I_2$; but $aaa \notin I_2$ – contradiction. So the unsolved problem which we described above now strikes in a different form, prohibiting any analogy in the first place. In particular, it shows us that we cannot simply transfer ideas from the substitutional approach to the functional approach.

What are the possible solutions? I see two options: the first is: we go back to the inference rules in $\overline{\mathfrak{g}}$. We have discarded it because it was too liberal, but now that surely does not hold any more. In this case, we do no longer derive functions, but single analogies, so we have a whole different ontology. Our analogical map thus looks like this:

$(\vec{x}, f(\vec{x})) \in \mathcal{P}P1'(I)$ iff

1. $f \in \mathcal{P}$;
2. if $\vec{x} \sqsubseteq \|f(\vec{x})\|$, then $\vec{x} \leq_I \|f(\vec{x})\|$;
3. if $\|f(\vec{x})\| \sqsubseteq \vec{x}$, then $\|f(\vec{x})\| \leq \vec{x}$.
4. if neither of the two holds, then $\vec{x} \sim_I \|f(\vec{x})\|$

So we have just skipped the universality conditions. Inferences have the following form:

$$\frac{\vec{x} \approx_I^{\mathcal{P}P1'} f(\vec{x})}{\vec{x} \Leftarrow_I^{\mathcal{P}P1'} f(\vec{x}),} \qquad \frac{\vdash \vec{w}\vec{x}\vec{v} \in \overline{\mathfrak{g}}^{late}_{\mathcal{P}P1'}(I) \quad \vec{x} \Leftarrow_I^{\mathcal{P}P1'} f(\vec{x})}{\vdash wf(\vec{x})\vec{v} \in \overline{\mathfrak{g}}_{\mathcal{P}P1'}(I)}. \tag{4.55}$$

This seems reasonable, and we get in fact (restricting ourselves to non-trivial functions)

$$\mathcal{P}P1'(\{a, aa, aaaa\}) = \{(a, f_1(a)), (aa, f_2(aa))\}, \tag{4.56}$$

where $f_1(\mathsf{x}) = \mathsf{xxxx}, f_2(\mathsf{x}) = \mathsf{xx}$. So we see that $\|\overline{\mathfrak{g}}^{late}_{\mathcal{P}P1'}(\{a, aa, aaaa\})\|$ is a language which is not semilinear. But still, it does not behave in the way we would expect it to, yielding a simple language such as $\{a^{2^n}\}$.

It might also be worthwhile to investigate how a pre-theory as $(\mathfrak{g}2, P2)$ can be adapted to polynomial functions; we will however not undertake this at this point. Instead of pursuing the problem of transformational (functional) pre-theories in further ramifications, we will present some additional means to enlarge our language-theoretic ontology. These will also lead to our next major pre-theory.

4.12 Strings as Typed λ-Terms

4.12.1 A Simple Type Theory

The objects on which inferences were based in the previous pre-theories went beyond simple language-theory. However, the conditions for analogies themselves were based on pure language-theoretic criteria; they did not refer to any objects except for strings. This will change now. We will now use what is known as the encoding of strings as typed λ-terms. This has been worked out and attracted considerable attention in the research on **abstract categorial grammars** (ACG). We will not have to do with ACG themselves, but most of the techniques we show have been developed within this research community.

We first have to introduce the basic notions of type theory; the following concepts are standard and can be looked up in many places.[6] Type theory starts with a (usually) finite set of basic types, and a finite, (usually) small set of type constructors. Types are usually interpreted as sets; we denote the set of all objects of type τ by $\|\tau\|$. We will consider only a single type constructor, the usual $\rightarrow$, where for types σ, τ, $\sigma \rightarrow \tau$ is the type of all functions from $\|\sigma\|$ to $\|\tau\|$. Basic objects are assigned some type, and all new objects we can construct in our universe must be constructed in accordance with a typing procedure, that is, we have to make sure that they can be assigned at least one type. Objects which are not well-typed do not exist in the typed universe.

Given a non-empty set A of atomic types, the set of types $Tp(A)$ is defined as closure of A under type constructors: $A \subseteq Tp(A)$, and if $\sigma, \tau \in Tp(A)$, then $\sigma \rightarrow \tau \in Tp(A)$. The **order** of a type is defined as $ord(\sigma) = 0$ for $\sigma \in A$, $ord(\sigma \rightarrow \tau) = max(ord(\sigma) + 1, ord(\tau))$.

We define a higher order signature as $\Theta := (A, C, \phi)$, where A is a finite set of atomic types, C is a set of constants, and $\phi : C \rightarrow Tp(A)$ assigns types to constants. The order of Θ is $max(\{ord(\phi(c)) : c \in C\})$. Let X be a countable set of variables. The set $\mathtt{Tm}(\Lambda(\Theta))$, the set of all λ terms over Θ, is the closure of $C \cup X$ under the following rules: 1. $C \cup X \subseteq \mathtt{Tm}(\Lambda(\Theta))$; 2. if $\mathtt{m}, \mathtt{n} \in \mathtt{Tm}(\Lambda(\Theta))$, then $(\mathtt{mn}) \in \mathtt{Tm}(\Lambda(\Theta))$; 3. if $x \in X, \mathtt{m} \in \mathtt{Tm}(\Lambda(\Theta))$, then $(\lambda x.\mathtt{m}) \in \mathtt{Tm}(\Lambda(\Theta))$.

We omit the outermost parentheses $(,)$ for λ terms, and write $\lambda x_1 ... x_n.\mathtt{m}$ for $\lambda x_1.(\ldots(\lambda x_n.\mathtt{m})...)$; furthermore, we write $\mathtt{m}_1\mathtt{m}_2\ldots\mathtt{m}_i$ for $(\ldots(\mathtt{m}_1\mathtt{m}_2)\ldots\mathtt{m}_i)$. The set of free variables of a term $\mathtt{m}$, $FV(\mathtt{m})$, is defined by 1. $FV(x) = \{x\} : x \in X$, 2. $FV(c) = \emptyset : c \in C$, 3. $FV(\mathtt{mn}) = FV(\mathtt{m}) \cup FV(\mathtt{n})$, and 4. $FV(\lambda x.\mathtt{m}) = FV(\mathtt{m}) - \{x\}$. $\mathtt{m}$ is closed if $FV(\mathtt{m}) = \emptyset$. We write $\mathtt{m}[\mathtt{n}/x]$ for the result of substituting $\mathtt{n}$ for all free occurrences of x in $\mathtt{m}$. α-conversion is defined as $\lambda x.\mathtt{m} \rightsquigarrow_\alpha \lambda y.\mathtt{m}[y/x]$. A β-redex is a term of the form $(\lambda x.\mathtt{m})\mathtt{n}$. We write $\rightsquigarrow_\beta$ for β reduction, so we have $(\lambda x.\mathtt{m})\mathtt{n} \rightsquigarrow_\beta \mathtt{m}[\mathtt{n}/x]$. The inverse of β-reduction is β-expansion. Let $[\mathtt{m}]_\beta$ denote the β normal form of $\mathtt{m}$, that is, the term without any β redex. This term is unique up to α-conversion for every term $\mathtt{m}$. We denote by $=_{\alpha\beta}$ the smallest **congruence** which contains both $\rightsquigarrow_\alpha$ and $\rightsquigarrow_\beta$ (recall that a congruence is an equivalence relation closed under subterms). We thus write $\mathtt{m} =_{\alpha\beta} \mathtt{n}$, if $\mathtt{n}$ can be derived from $\mathtt{m}$ with any finite series of steps of β-reduction, expansion or α-conversion of any of its subterms. Later on, we will extend $=_{\alpha\beta}$ to *sets* of terms; assume $\mathtt{M}, \mathtt{N} \subseteq \mathtt{Tm}(\Lambda(\Theta))$ (we will define this set in a minute); we then write $\mathtt{M} =_{\alpha\beta} \mathtt{N}$, if for every $\mathtt{m} \in \mathtt{M}$, there is a $\mathtt{n} \in \mathtt{N}$ such that $\mathtt{m} =_{\alpha\beta} \mathtt{n}$,

[6] Here I follow in particular my own presentation given in [71], which in turn follows the presentations given in [30], [25].

and for every $\mathtt{n} \in \mathtt{N}$, there is a $\mathtt{m} \in \mathtt{M}$ such that $\mathtt{m} =_{\alpha\beta} \mathtt{n}$.

We now come to the procedure of assigning types to terms.[7] A *type environment* is a (possibly empty) set $\{x_1 : \alpha_1, \ldots x_n : \alpha_n\}$ of pairs of variables and types, where each variable occurs at most once. A λ-term $\mathtt{m}$ with $FV(\mathtt{m}) = \{x_1, \ldots, x_n\}$ can be assigned a type α in the signature $\Theta = (A, C, \phi)$ and type environment $\{x_1 : \alpha_1, \ldots x_n : \alpha_n\}$, in symbols

$$x_1 : \alpha_1, \ldots x_n : \alpha_n \vdash_\Theta \mathtt{m} : \alpha, \tag{4.57}$$

if it can be derived according to the following rules:

(cons) $\vdash_\Theta c : \phi(c)$, for $c \in C$;

(var) $x : \alpha \vdash_\Theta x : \alpha$, where $x \in X$ and $\alpha \in Tp(A)$;

(abs) $\dfrac{\Gamma \vdash_\Theta \mathtt{m} : \beta}{\Gamma - \{x : \alpha\} \vdash_\Theta \lambda x.\mathtt{m} : \alpha \to \beta}$, provided $\Gamma \cup \{x : \alpha\}$ is a type environment;

(app) $\dfrac{\Delta \vdash_\Theta \mathtt{n} : \alpha \quad \Gamma \vdash_\Theta \mathtt{m} : \alpha \to \beta}{\Gamma \cup \Delta \vdash_\Theta \mathtt{mn} : \beta}$, provided $\Gamma \cup \Delta$ is a type environment.

An expression of the form $\Gamma \vdash_\Theta \mathtt{m} : \alpha$ is called a *judgment*, and if it is derivable by the above rules, it is called the *typing* of $\mathtt{m}$. A term $\mathtt{m}$ is called *typable* if it has a typing. If in a judgment we do not refer to any particular signature, we also write $\Gamma \vdash \mathtt{m} : \alpha$. Derivations of judgments have the forms of trees; the derivation tree of a judgment $\Gamma \vdash_\Theta \mathtt{m} : \alpha$ is called its deduction. When $\mathtt{m}$ is β-normal, every typing of $\mathtt{m}$ has a unique deduction. Regarding β-reduction, we have the following well-known result:

Theorem 126 *(Subject Reduction Theorem) If* $\Gamma \vdash \mathtt{m} : \alpha$, $\mathtt{m} \rightsquigarrow_\beta \mathtt{m}'$, *then* $\Gamma' \vdash \mathtt{m}' : \alpha$, *where* Γ' *is the restriction of* Γ *to* $FV(\mathtt{m}')$.

We define the set $\mathtt{WTT}$ to be the set of all well-typed, closed λ-terms, that is, all $\mathtt{n}$ such that $\vdash \mathtt{n} : \alpha$ is derivable for some α (we might refer to a signature Θ by writing $\mathtt{WTT}(\Theta)$. Keep in mind that $\mathtt{WTT}(\Theta) \subsetneq \mathtt{Tm}(\Lambda(\Theta))$; it is $\mathtt{WTT}$ which is most interesting and important for us. Let $\mathtt{m} \rightsquigarrow_\beta \mathtt{m}'$ be a contraction of a redex $(\lambda x.\mathtt{n})\mathtt{o}$. This reduction is *non-erasing* if $x \in FV(\mathtt{n})$, and *non-duplicating* if x occurs free in $\mathtt{n}$ at most once. A reduction from $\mathtt{m}$ to $\mathtt{m}'$ is non-erasing (non-duplicating) if all of its reduction steps are non-erasing (non-duplicating). We say a term $\mathtt{m}$ is linear, if for each subterm $\lambda x.\mathtt{n}$ of $\mathtt{m}$, x occurs free in $\mathtt{n}$ exactly once, and each free variable of $\mathtt{m}$ has just one occurrence free in $\mathtt{m}$. Linear λ-terms are thus the terms, for which each β-reduction is non-erasing and non-duplicating. We will be mainly interested in a slightly larger class. A term $\mathtt{m}$ is a λI term, if for each subterm $\lambda x.\mathtt{n}$ of $\mathtt{m}$, x occurs free in $\mathtt{m}$ at least once. λI terms are thus the terms which do not allow for vacuous abstraction (see [2], chapter 9 for extensive treatment). Another important, well-known result for us is the following: obviously, by our typing procedure a single term might be possibly assigned many types. We call a **type substitution** a map $ts : A \to Tp(A)$, which respects the structure of types: $ts(\beta \to \gamma) = (ts(\beta)) \to (ts(\gamma))$, for $\beta, \gamma \in Tp(A)$.

[7] We adopt what is known as Curry-style typing: in Church-style typing, terms cannot be constructed without types; in Curry-style typing, terms are first constructed and then assigned a type; so there might be the case that there is no possible assignment.

Theorem 127 *(Principal Type Theorem) Let* $\mathtt{m}$ *be a term, and let* $\Theta := \{\alpha : \Gamma \vdash \mathtt{m} : \alpha \text{ is derivable}\}$ *be the set of all types which can be assigned to* $\mathtt{m}$*. If* $\Theta \neq \emptyset$*, then there exists a* ***principal type*** β *for* $\mathtt{m}$*, such that* $\Gamma \vdash \mathtt{m} : \beta$ *is derivable, and for each* $\alpha \in \Theta$*, there is a substitution* ts_α *such that* $\alpha = ts_\alpha(\beta)$*.*

Obviously, β is unique up to isomorphism; we will write $pt(\mathtt{m})$ for the principal type of $\mathtt{m}$. The proof of the theorem is constructive, that is, β can be effectively computed or shown to be nonexistent, see [25].

4.12.2 Strings as λ-Terms

The following, type theoretic encoding of language theoretic entities has been developed in the framework on on **abstract categorial grammars** (introduced in [14]). We follow the standard presentation given in [30]. Given a finite alphabet T, a string $a_1 \dots a_n \in T^*$ over T can be represented by a λ term over the signature $\Theta_T^{string} := (\{o\}, T, \phi)$, where for all $a \in T$, $\phi(a) = o \to o$; we call this a *string signature* (over alphabet T. The term is linear and written as $/a_1 \dots a_n/ := \lambda x.a_1(\dots(a_n x)\dots)$. Obviously, the variable x has to be type o, in order to make the term typable. We then have, for every string $\vec{w} \in T^*$, $\vdash_{\Theta_T^{string}} /\vec{w}/ : o \to o$.

Under this representation, string concatenation is not entirely trivial, and cannot be done by juxtaposition, as the result would not be typable. We can concatenate strings by the combinator $\mathbf{B} := \lambda xyz.x(yz)$, which concatenates its first argument to the left of its second argument, as can be easily checked.[8] We can also represent tuples of strings by terms. Let $/\vec{w}_1/, \dots, /\vec{w}_n/$ represent strings. Then a tuple of these strings is written as $/(\vec{w}_1, \dots, \vec{w}_n)/ := \lambda x.((\dots(x/\vec{w}_1/)\dots)/\vec{w}_n/)$. The type of x here depends on the size of the tuple. We define $\alpha \to_n \beta$ by $\alpha \to_0 \beta = \beta$, $\alpha \to_{n+1} \beta = \alpha \to (\alpha \to_n \beta)$. In general, for a term $\mathtt{m}$ encoding an n-tuple , we have $\vdash_{\Theta_T^{string}} \mathtt{m} : ((o \to o) \to_n (\alpha))) \to \alpha$. So the types get larger with the size of tuples; the *order* of the term however remains invariantly 3.

We indicate how to manipulate tuple components separately. The function which concatenates the tuple components in their order is obtained as follows: Given a tuple $/(\vec{w}, \vec{v})/ = \lambda x.((x/\vec{w}/)/\vec{v}/)$, we obtain $/\vec{w}\vec{v}/$ through application of the term: $\lambda x_1.x_1(\lambda x_2 y.\mathbf{B}x_2 y))$. We can also manipulate tuples to form new tuples: take again $/(\vec{v}_1, \vec{w}_1)/ = \lambda x.((x/\vec{v}_1/)/\vec{w}_1/)$; we want to convert it into a tuple $/(\vec{v}_1\vec{v}_2, \vec{w}_1\vec{w}_2)/ = \lambda x.((x/\vec{v}_1\vec{v}_2/)/\vec{w}_1\vec{w}_2/)$. This is done by the term $\lambda y x_1.y(\lambda x_2 x_3((x_1\mathbf{B}x_2\vec{v}_2)\mathbf{B}x_3\vec{w}_2))$. This term takes the tuple as argument and returns a tuple of the same type. If we abstract over the term $/(\vec{v}_2, \vec{w}_2)/$, this gives us a function which concatenates two 2-tuples componentwise. It is easy to see that this way, we can represent any polynomial function by a λ-term.

However, the general componentwise concatenation of tuples of arbitrary size (considering strings as 1-tuples) cannot be effected by a typed λ-term. The reason is: if we do not fix an upper bound on tuple size, the types of tuples get higher and higher, and there is no finite upper bound. So there is no finite term which could have the appropriate type.[9] This means that in this setting,

[8] See [30] for more examples, also for what is to follow. A *combinator* is in general a function over functions.

[9] On the other side, once we fix an upper bound k to tuple size, it is easy to see how to define $\circ$ as λ term: for $i \leq k$, we simply encode all tuples as k-tuples with all jth components, $i < j$, containing the empty string. Then $\circ$ is simply componentwise concatenation of k-tuples, which is λ-definable, as we have seen.

we must refrain from a notion of general concatenation of any type. This will however do little harm, as we will see.

4.12.3 Using λ-terms for Pre-Theories

Take a finite alphabet T, and fix a language $L_1 \subseteq T^*$. As we have seen in the last section, there is a injective map $i : T^* \to \mathtt{WTT}(\Theta_T^{string})$ from strings in T^* into the set of λ-terms of the signature Θ_T^{string}. Note that i is properly injective and not up to $=_{\alpha\beta}$ equivalence; we map strings only onto their standard encoding, using a standard variable. We thus obtain $i[L_1] \subseteq \mathtt{WTT}$, where $i[-]$ is the pointwise extension of i to sets. We close $i[L_1]$ under $=_{\alpha\beta}$, and obtain $L := \{\mathtt{m} : \text{there is } \mathtt{n} \in i[L] : \mathtt{n} =_{\alpha\beta} \mathtt{m}\}$. This is the language we are working with, the type theoretic counterpart of L_1. In the sequel, for any $M \subseteq T^*$, we will denote the closure of $i[M]$ under $=_{\alpha\beta}$ by M^λ; so we have $L = (L_1)^\lambda$. Given an analogical map P for T^*, now want to devise λP for I^λ, referring to subterms instead of factors of the language.

There is however a fundamental problem with that. Take a finite, non-empty language I. It is the easy to see that I^λ is infinite; so we do not have a finite language to depart from. This in itself is no need to worry: for a set M of terms, let us denote by $[M]_{\alpha\beta}$ the partition of M into $=_{\alpha\beta}$-equivalent subsets. For our language-theoretic purposes, we need to consider terms only up to $=_{\alpha\beta}$, and $[I^\lambda]_{\alpha\beta}$ is (by assumption) finite. But we have to consider not only the terms, but also their *subterms*! Define $sub(I^\lambda) := \{sub(\mathtt{m}) : \mathtt{m} \in I^\lambda\}$. Now unfortunately, the quotient $[sub(I^\lambda)]_{\alpha\beta}$ is in general *infinite*; so we have infinitely many distinct subterms even modulo $=_{\alpha\beta}$! We can show this by a simple example: put $I = \{a\}$. Then we have $I^\lambda = \{\lambda x.ax, (\lambda yx.yx)a, ((\lambda zyx.(zy)x)(\lambda z.z))a, (((\lambda z_1zyx.((z_1z)y)x)(\lambda z_1.z_1))(\lambda z.z))a, ...\}$

It is easy to see that the leftmost closed terms $(\lambda yx.yx)$, $(\lambda zyx.(zy)x)$, $(\lambda z_1zyx.((z_1z)y)x)$ are all not $\alpha\beta$-equivalent, and we can iterate the above expansion as often as we want. So we have a major problem, because the factors we have to consider are infinitely many even modulo $\alpha\beta$-equivalence. So what are we supposed to do? There does not seem to be very good solution at this point, so the only solution I can present is the following: rather than using $=_{\alpha\beta}$, we define a relation $=^k_{\alpha\beta}$, which we define inductively for every $k \in \mathbb{N}$. Let $=_\alpha$ be the smallest congruence containing $\rightsquigarrow_\alpha$.

1. if $\mathtt{m} =_\alpha \mathtt{n}$, then $\mathtt{m} =^0_{\alpha\beta} \mathtt{n}$; (closure under α-conversion)
2. if $\mathtt{m} =^k_{\alpha\beta} \mathtt{n}$, then $\mathtt{m} =^{k+1}_{\alpha\beta} \mathtt{n}$ (monotonicity over k)
3. if $\mathtt{m} =^k_{\alpha\beta} \mathtt{n}$, then $\mathtt{n} =^k_{\alpha\beta} \mathtt{m}$; (symmetry)
4. if $\mathtt{m} \rightsquigarrow_\beta \mathtt{n}$, then $\mathtt{m} =^1_{\alpha\beta} \mathtt{n}$ (definition for $k = 1$)
5. if $\mathtt{m} =^k_{\alpha\beta} \mathtt{n}$, then $\mathtt{o}[\mathtt{m}] =^k_{\alpha\beta} \mathtt{o}[\mathtt{n}]$; (closure under subterms)
6. if $\mathtt{m} =^k_{\alpha\beta} \mathtt{n}$, $\mathtt{n} =^{k'}_{\alpha\beta} \mathtt{o}$, then $\mathtt{m} =^{k+k'}_{\alpha\beta} \mathtt{o}$. (transitivity for addition of k, k')

So $=^k_{\alpha\beta}$ is the rewriting relation which involves k β-reduction or expansion steps, and an arbitrary amount of α-conversions. Of course, we have $(\bigcup_{k\in\mathbb{N}} =^k_{\alpha\beta}$

$) = (\ =^{\omega}_{\alpha\beta}) \ = \ (=_{\alpha\beta})$. To make this restriction sufficient, we will now and for the rest of this chapter reduce our focus to λI-terms.

What we intend to do is: for a finite language I, close $i[I]$ under $=^{k}_{\alpha\beta}$ for some k, rather than close it under $=_{\alpha\beta}$. We denote this closure *and* the intersection with the set of λI terms by $I^{\lambda k}$; $I^{\lambda k}$ does thus only contain λI-terms. It is clear that for a finite set of terms I, $I^{\lambda k} := \{\mathtt{m} : \text{there is } \mathtt{n} \in I : \mathtt{n} =^{k}_{\alpha\beta} \mathtt{m}\}$ modulo $=_{\alpha}$ is a finite set; we can easily show this by induction: $[I^{\lambda 1}]_{\alpha}$ is finite; and if $[I^{\lambda k}]_{\alpha}$, then also $[I^{\lambda k+1}]_{\alpha}$ is finite. By the same argument, we can conclude that for each set $I^{\lambda k}$, I finite, there is a constant $k \in \mathbb{N}$ such that for all $\mathtt{m} \in I^{\lambda k}$, we have $|\mathtt{m}| \leq k$, where $|-|$ denotes the length of the term. From this it easily follows that $sub(I^{\lambda k})$ is a finite set, as actually, as the terms of $I^{\lambda k}$ are constantly bounded in length, so are the terms in $sub(I^{\lambda k})$. It follows that $[sub(I^{\lambda k})]_{\alpha}$ is finite, and *a fortiori*, $[sub(I^{\lambda k})]_{\alpha\beta}$ is finite. Note that all the arguments – except for the very last – are wrong when we do not restrict ourselves to λI.

So there is a solution to the infinity problem for $\lambda P1$: we just have to consider families of the form $\lambda kP1, \lambda kPr$. The problem is of course: there is always something arbitrary to a pre-theory of this form, as there is no real criterion for choosing k.

Our goal is by now clear: we want to use the language of terms just as a "normal language", putting our old concepts to work. There are however some things we have to take care of: typed terms are not just strings: we do not have associativity of concatenation in the first place. So we need to respect the bracketings. But there is more: we have to take care that all objects we talk about are indeed typable terms; if not, we would take about things which are non-sense from the point of view of type theory.

Our major asset now is the following: concatenation on the level of strings interpreted as terms is now quite independent of the juxtaposition of terms, and $\mathtt{mn}$ corresponds to applying a function $\mathtt{m}$ to an argument $\mathtt{n}$. For us, this means we can restrict our focus to analogies of the form $(\mathtt{n}, \mathtt{mn})$. But instead of writing $\vec{w}\vec{x}\vec{v}$ to indicate substrings, we will use the subterm notation: by $\mathtt{m}[\mathtt{n}]$ we denote a term $\mathtt{m}$ which contains a term $\mathtt{n}$ as a subterm; importantly, we always require that $\mathtt{n}, \mathtt{m} \in \mathtt{WTT}$; we thus only refer to closed, well-typed terms as terms and subterms. Another important thing is that we denote a single *occurrence* of $\mathtt{n}$ within $\mathtt{m}$; but there are possibly many of them. And if we use $\mathtt{m}[\mathtt{n}]$ subsequently, we always refer to the same occurrence of $\mathtt{n}$. We could make this explicit by adding subscripts to $[,]$, so that they refer to positions in the term; explicitly, this is written as $\mathtt{m}[_{\mathtt{i}}\mathtt{n}]_{\mathtt{j}}$. Importantly, we count the left index from the left, and the right index from the right. We omit this for simplicity and because it is common usage. Then, by $\mathtt{m}[\mathtt{o}/\mathtt{n}]$ we denote the term which results in a substitution of this subterm $\mathtt{n}$ in $\mathtt{m}[\mathtt{n}]$ by $\mathtt{o}$ (so this usage is somewhat different from $\mathtt{m}[\mathtt{n}/x]$, by which we intend the substitution of all free variables). To indicate multiple occurrences of subterms, we simply write $\mathtt{m}[\mathtt{n}_1, ..., \mathtt{n}_{\mathtt{i}}]$, where we make no statement on position and order of the $\mathtt{n}_1, ..\mathtt{n}_{\mathtt{i}}$ Now we can define the following analogical map:

Definition 128 *Given a finite language I, we have $\mathtt{n} \approx^{\lambda P1k}_{I} \mathtt{mn}$, if and only if $\mathtt{o}[\mathtt{mn}] \in I^{\lambda k} \Rightarrow \mathtt{o}[\mathtt{n}/\mathtt{mn}] \in I^{\lambda k}$.*

Some notes are in order. There does not seem to be a reasonable type-theoretic variant of Pr. From the basic results of type theory it follows that

$I^\lambda \subseteq \mathtt{WTT}$. By notation, we already required that $\mathtt{m}, \mathtt{n} \in \mathtt{WTT}$. So all our objects are in the universe of well-typed terms over the string signature of a given alphabet. This also takes the worries from us that our objects are actually meaningless; to all terms in WTT we can – at least in a very broad sense – assign some meaning. In what consists this meaning? We know that all entities will be functions, so they will be in the end some higher order functions from (functions from functions...to) strings to (functions to functions ..to) strings. Another problem for which now there is a unique and natural solution is the question of weak and strong language: the weak language - the first input and final output - consists of strings over an alphabet. The strong language, which is used for inferences, consists of terms in $\mathtt{WTT}(\Theta^\Sigma_{string})$, for Σ an alphabet.

Recall that we have the bijection $i : \Sigma^* \to \mathtt{WTT}(\Theta^\Sigma_{string})$, and the map $[-]^{\lambda k}$, which is i composed with closure under $=^k_{\alpha\beta}$. I propose the following inference rule, here denoted by $\mathfrak{g}^\lambda$:

$$\frac{\mathtt{o} \approx^{\lambda P}_{I^\lambda} \mathtt{no} \quad \mathtt{m}[\mathtt{o}] \in \mathfrak{g}^\lambda_{\lambda P}(I^{\lambda k})}{\mathtt{m}[\mathtt{no}] \in \mathfrak{g}^\lambda_{\lambda P}(I^{\lambda k})} \tag{4.58}$$

Note how surprisingly simple the schema has become again. But the map getting us back to the weak language is a bit more complicated: we have to take the map $j : \wp(\mathtt{WTT}(\Theta^\Sigma_{string})) \to \wp(\Sigma)$, where $j(M) = i^{-1}[M \cap i[\Sigma^*]]$. In words, we first intersect the set of terms with the set of terms $i[\Sigma^*]$, the terms representing strings in canonical form, and then take the inverse image under i. This leads us back to a language, and frees us from a deep worry. This worry, on which we will speak more explicitly later on, is the following: assume we derive a term $\mathtt{m}$. The premise we use is of course in $(\Sigma^*)^\lambda$; but how about the conclusion? We might derive terms which are not $\alpha\beta$-equivalent to any string encoding. This is surely not a virtue, but by our map j, this also will do no harm: these entities are simply sorted out.

So there is no problem if we derive terms that do not reduce to strings. a negative answer. However, a positive answer would allow us to derive a couple of positive results; the first one being: we could say that $\lambda P1$ is *closed*, in the sense that it either generates an infinite language or the identity. The second result we could derive would be that there are some non-trivial boundaries for languages we can generate, by means of growth ratio. But so far there is little hope.

We have said that given a finite language I, it is clear that I^λ is *infinite*. This is problematic for analogical maps; for inferences, it rather seems desirable to close inferences under $=_{\alpha\beta}$, because we want to be able to reduce as much as possible in the end. Here, it turns out that we can simply include α-conversion and β-reduction/expansion into the rules of our calculus. This means, in addition to the above rule, we have the following in $\mathfrak{g}^\lambda$:

$$\frac{\mathtt{m} \in \mathfrak{g}^\lambda_{\lambda P}(I^{\lambda k}) \quad \mathtt{m} =_{\alpha\beta} \mathtt{n}}{\mathtt{n} \in \mathfrak{g}^\lambda_{\lambda P}(I^{\lambda k})} \tag{4.59}$$

For those who object to this scheme that the relation $=_{\alpha\beta}$ is not finitary, we answer that we might read this scheme as a shorthand for complex derivations where each single step consists of α-conversion, β-reduction or β-expansion. So we can reduce the infinite set to a finite set by introducing additional inferences

for terms, which again are finitary in nature. We thus just make the derivation steps of the λ-calculus part of our inference mechanism. This seems to be a good solution.

So our procedure works as follows: given a finite language I, we first take $I^{\lambda k}$, and let our analogical map compute the analogies. Then we depart from I^{λ} then form the deductive closure under the inference rules above, thereby yielding $\mathfrak{g}^{\lambda}_{\lambda P1k}(I^{\lambda})$. So whereas for $\lambda P1$, we take the $I^{\lambda k}$, the premises for inferences are not restricted in a similar way. Finally, given a finite language I, the infinite language we are after is $j \circ \mathfrak{g}^{\lambda}_{\lambda P1} \circ [I]^{\lambda}$.

What class of languages do we induce, that is, what is the class of languages L such that there is a finite I with $L = j \circ \mathfrak{g}^{\lambda}_{\lambda P1} \circ [I]^{\lambda}$? It might be worth a try to show how the resulting language can be generated by an abstract categorial grammar; however, as there are no known non-trivial upper bounds for the expressive power of ACGs, this would be quite uninformative, and so there would be little point in this exercise. Also for other upper possible bounds, there does not seem to be any class which would be worth the effort one has to put in for a result. We can however easily see that this approach generalizes most of our previous approaches: we can encode tuples of arbitrary size without any problem, as we have seen; we can also encode things as duplication with a term as $\lambda.(\mathbf{B}x)x$. In fact, we can easily represent any polynomial function in this way, as well as tuple concatenation etc.

The most urgent problem is however: our original problem regarding $L_i := \{a^{2^n} : n \in \mathbb{N}\}$ still remains: we do not induce the language L_i from any fragment thereof such as $I_e := \{a, aa, aaaa\}$, for the same reasons as above. We encounter exactly the same problem as before, in that we cannot control to which strings we should apply a function and to which not. We will therefore by re-introduce formal concept analysis.

4.13 Concepts and Types

4.13.1 A Context of Terms

We now take a more general perspective on formal concepts. A **context** is a triple $(\mathcal{G}, \mathcal{M}, \mathcal{I})$, where $\mathcal{G}, \mathcal{M}$ are sets and $\mathcal{I} \subseteq \mathcal{G} \times \mathcal{M}$. In FCA, the entities in $\mathcal{G}$ are thought of as objects, the objects in $\mathcal{M}$ as attributes, and for $m \in \mathcal{M}$, $g \in \mathcal{G}$, we have $(g, m) \in I$ if the object g has the attribute m. This is all we need as basic structure to get the machine of FCA going. For $A \subseteq \mathcal{G}, B \subseteq \mathcal{M}$, we put $A^{\triangleright} := \{m \in \mathcal{M} : \forall a \in A, (a, m) \in \mathcal{I}\}$, and $B^{\triangleleft} := \{g \in \mathcal{G} : \forall m \in B, (a, m) \in \mathcal{I}\}$. A concept is a pair (A, B) such that $A^{\triangleright} = B, B^{\triangleleft} = A$. We call A the **extent** and B the **intent**. A is the extent of a concept iff $A = A^{\triangleright\triangleleft}$, dually for intents. The maps $[-]^{\triangleright}, [-]^{\triangleleft}$ are called **polar maps**. As before, we order concepts by inclusion of extents, that is, $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$.

Definition 129 *Given a context $\mathsf{C} = (\mathcal{G}, \mathcal{M}, \mathcal{I})$, we define the concept lattice of C as $\mathcal{L}(\mathsf{C}) = \langle \mathfrak{B}, \wedge, \vee, \top, \bot \rangle$, where $\top = (\mathcal{G}, \mathcal{G}^{\triangleright})$, $\bot = (\mathcal{M}^{\triangleleft}, \mathcal{M})$, and for $(A_i, B_i), (A_j, B_j) \in \mathfrak{C}$, $(A_i, B_i) \wedge (A_j, B_j) = (A_i \cap A_j, (B_i \cup B_j)^{\triangleleft\triangleright})$, and $(A_i, B_i) \vee (A_j, B_j) = ((A_i \cup A_j)^{\triangleright\triangleleft}, B_i \cap B_j)$.*

We define our type theoretic context as follows. Recall that $\mathtt{Tm}(\Lambda(\Theta))$ is the set of all λ-terms over Σ. We put $\mathtt{Tm}_c(\Lambda(\Theta)) := \{\mathtt{m} \in \mathtt{Tm}(\Lambda(\Theta)) : FV(\mathtt{m}) = \emptyset\}$,

the set of closed terms, and we call $\mathtt{Tm}_c(\Lambda I(\Sigma))$ the set of all closed λI terms. Furthermore, define $\mathtt{WTT}$ as the set of all closed and well-typed terms, that is, the set of all terms $\mathtt{m}$ such that $\vdash \mathtt{m} : \alpha$ is derivable for some α by our rules; $\mathtt{WTT}_I = \mathtt{WTT} \cap \mathtt{Tm}(\Lambda I(\Sigma))$. Recall that $=_{\alpha\beta}$ is a congruence. Let σ be a given type, and $L' \subseteq \|\sigma\|$ be a distinguished subset of the terms of type σ; we define $L := \{\mathtt{m} : \exists \mathtt{n} \in L' : \mathtt{m} =_{\alpha\beta} \mathtt{n}\}$, that is, as closure of L' under $=_{\alpha\beta}$. Put $\mathcal{G} = \mathcal{M} = \mathtt{Tm}_c(\Lambda(\Theta))$, and define the relation $\mathcal{I} \subseteq \mathtt{Tm}_c(\Lambda(\Theta)) \times \mathtt{Tm}_c(\Lambda(\Theta))$ as follows: for $\mathtt{m}, \mathtt{n} \in \mathtt{Tm}_c(\Lambda(\Theta))$, we have $(\mathtt{m}, \mathtt{n}) \in \mathcal{I}$ if $\mathtt{nm} \in L$. So the relation of objects in $\mathcal{M}$ and $\mathcal{G}$ is that of function and argument, and the relation I tells us whether the two yield a desired value. Same can be done with $\mathtt{Tm}_c(\Lambda I(\Sigma))$.

Obviously, we have $\bot = (\emptyset, \mathtt{Tm}_c(\Lambda(\Theta)))$. Regarding upper bounds, we have to distinguish two important concepts: we first have a concept we denote $\intercal := (\mathtt{WTT}, \Lambda V)$, where ΛV (V for vacuous) is the set of all terms of the form $\lambda x.\mathtt{m}$, where $\mathtt{m} \in L$ and $x \notin FV(\mathtt{m})$. There is however a larger concept $\top \geq \intercal$, which is defined as $\top := (\mathtt{Tm}_c(\Lambda(\Theta)), \emptyset)$. The reason for this slight complication is as follows: we want our terms to be closed, because open terms are meaningless for us. Now, it holds that the concatenation of closed terms is again a closed term; but the concatenation of well-typed terms need not be well-typed: for $\mathtt{m}, \mathtt{n} \in \mathtt{WTT}$, it might be that $\mathtt{nm} \notin \mathtt{WTT}$. Furthermore, there are λ-terms $\mathtt{n}$ with vacuous abstraction such that the set $\{\mathtt{nm} : \mathtt{m} \in \intercal\} \lneq \intercal$; we would however like our $\top$ to be absorbing; and in fact, if $\mathtt{m} \notin \mathtt{WTT}$, then for any term $\mathtt{n}$, $\mathtt{nm}, \mathtt{mn} \notin \mathtt{WTT}$. So for every term $\mathtt{m} \notin \mathtt{WTT}$, $\{\mathtt{m}\}^{\triangleright} = \emptyset$. For all interesting results we have to restrict ourselves to $\mathtt{WTT}$, but for completeness of some operations we have to consider $\mathtt{Tm}_c(\Lambda(\Theta))$. Note however that if we restrict $\mathtt{Tm}_c(\Lambda(\Theta))$ to $\mathtt{Tm}_c(\Lambda I(\Sigma))$, then $\top$ and $\intercal$ coincide.[10]

4.13.2 Concept Structure and Type Structure

We have seen that each term can be assigned a most general type. Importantly, the same holds for sets of types:

Lemma 130 *Let $T \subseteq \mathtt{WTT}$ be a set of terms, such that the set of principal types $\{pt(\mathtt{m}) : \mathtt{m} \in T\}$ is finite. If there is a non-empty set of types Θ, such that for each $\mathtt{m} \in T$ and all $\theta \in \Theta$, $\vdash \mathtt{m} : \theta$ is a derivable judgment, then there is a (up to isomorphism) unique type α, such that for every $\mathtt{m} \in T$, $\vdash \mathtt{m} : \alpha$ is derivable, and every $\theta \in \Theta$ can be obtained by α through a type substitution.*

α is usually called the **most general unifier** of Θ; for a set of terms T, we also directly call it $pt(T)$, the principal type of T; for Θ a set of types, we denote it by $\bigvee \Theta$. A proof for this fundamental lemma can be found in [25]; again the proof is constructive. Note that if we do not assume that the set of principal types of terms $\mathtt{m} \in T$ modulo isomorphism is finite, then there is no upper bound on the length of types, and so there cannot be a finite common unifier. For convenience, we introduce an additional type $\top \notin A$, such that our types are the set $\{\top\} \cup Tp(A)$. If a set of types Θ does not have a common unifier, then we put $\bigvee(\Theta) = \top$, thus making the operation complete and defined in all cases.

This is of immediate importance for us, as it allows us both to speak of the principal type of a set of terms, as well as of the least upper bound of a

[10] This is actually not straightforward; this follows however easily from the results presented in [71].

set of types. From there we easily arrive at the greatest lower bound of two types α, β, which we denote by $\alpha \wedge \beta$, and which intuitively is the amount of structure which α and β share. Write $\alpha \leq \beta$, if there is a substitution ts such that $ts(\alpha) = \beta$. This is, modulo isomorphism, a partial order. We now can simply define $\alpha \wedge \beta := \bigvee\{\gamma : \gamma \leq \alpha, \beta\}$. It is clear that the set $\{\gamma : \gamma \leq \alpha, \beta\}$ modulo isomorphism is finite, so the (finite) join exists in virtue of the above lemma. So $Tp(A)$ is lattice ordered up to isomorphism, even though we do not have an explicit smallest element $\bot$: modulo isomorphism, α is the unique smallest type.

How does type structure behave wrt. concept structure? First of all, if $A \subseteq B$, then $pt(A) \leq pt(B)$. So the inclusion relation reflects type structure. This entails that $pt(A) \leq pt(B^{\triangleright\triangleleft})$. Stronger results are hard to obtain; for example, if we know $pt(A)$, there is nothing we can say in general about an upper bound for $pt(A^{\triangleright\triangleleft})$. Fortunately, there is more we can say about the lattice order of concepts and type order. Define $\vee$ and $\wedge$ on concepts as usual. For a concept (A, B) over the term context, we put $pt_1(A, B) = pt(A)$, $pt_2(A, B) = pt(B)$.

Lemma 131 *For concepts $\mathcal{C}_1, \mathcal{C}_2$ of the term context, the following holds: (1) If $\mathcal{C}_1 \leq \mathcal{C}_2$, then $pt_1(\mathcal{C}_1) \leq pt_1(\mathcal{C}_2)$, and $pt_2(\mathcal{C}_2) \leq pt_2(\mathcal{C}_1)$. (2) $pt_1(\mathcal{C}_1 \wedge \mathcal{C}_2) \leq pt_1(\mathcal{C}_1) \wedge pt_1(\mathcal{C}_2)$, and (3) $pt_1(\mathcal{C}_1) \vee pt_1(\mathcal{C}_2) \leq pt_1(\mathcal{C}_1 \vee \mathcal{C}_2)$.*

Proof. The first claim is immediate by set inclusion. To see the second, consider that for every $\mathtt{m} \in A_1 \cap A_2$, we must have $pt(\{\mathtt{m}\}) \leq pt(A_1), pt(\{\mathtt{m}\}) \leq pt(A_2)$ by set inclusion; and so $pt(\{\mathtt{m}\}) \leq pt_1(\mathcal{C}_1) \wedge pt_1(\mathcal{C}_2)$. To see the third claim, consider the following: we can easily show that $pt(A_1) \vee pt(A_2) = pt(A_1 \cup A_2)$. Then the claim follows from considering that $pt(A_1 \cup A_2) \leq pt((A_1 \cup A_2)^{\triangleright\triangleleft})$. □

Definition 132 *A term $\mathtt{m}$ is a **left equalizer**, if we have $\vdash \mathtt{m} : \theta_1 \to \alpha$, $\vdash \mathtt{m} : \theta_2 \to \alpha$, and $\theta_1 \neq \theta_2$. $\mathtt{m}$ is a **right equalizer**, if $\vdash \mathtt{m} : \alpha_1$, $\vdash \mathtt{m} : \alpha_2$, and $\alpha_1 \neq \alpha_2$.*

Easy examples of left equalizers are terms with vacuous abstraction; easy examples of right equalizers are terms which do not contain constants. Note that every left-equalizer is a right equalizer; and so a term which is both a left and right equalizer is $\lambda yx.x$. The following results are a bit tedious to obtain, yet not very significant; we therefore omit the proof.

Lemma 133 *Let $T \subseteq \mathtt{WTT}$, such that $pt(T) = \top$. Then each $\mathtt{m} \in T^{\triangleright}$ is a left equalizer.*

We can thus also speak of equalizer concepts. If we restrict our context to λI terms, we get a stronger result:

Lemma 134 *Let $\mathtt{m}$ be a left equalizer and λI-term, such that $\vdash_\Theta \mathtt{m} : \theta_1 \to \alpha$ and $\vdash_\Theta \mathtt{m} : \theta_2 \to \alpha$. Then both θ_1, θ_2 must be types inhabited by terms in $\mathtt{Tm}(\Lambda I(\Sigma))$, that is, there are terms $\mathtt{m}_i$, for which $\vdash_\Theta \mathtt{m}_i : \theta_i$ is derivable for $i \in \{1, 2\}$ and $\mathtt{m}_i \in \mathtt{Tm}(\Lambda I(\Sigma))$.*

So when we restrict ourselves to ΛI, we have proper restrictions on the class of possible equalizers, in the general case we do not. For example, assume there is a set T of terms, and $pt(T) \neq \top$. Still, we might have $pt(T^{\triangleright}) = \top$. Conversely, from the fact that $pt(T) = \top$, it does not follow that $T^{\triangleright} = \emptyset$.

Of course, all general results of FCA also hold in this particular setting. For us, the question is not in how far is the type theoretic context interesting as a *particular* context, but rather: in how far can we use type theoretic contexts in order to *generalize* the approach taken before using syntactic concept lattices? There we had the (implicit) context $\mathsf{C}_S(L) = (\Sigma^*, (\Sigma^*)^2, \mathcal{I}_L)$ (S is for "string"), where $(\vec{w}, (\vec{x}, \vec{x})) \in \mathcal{I}_L$ iff $\vec{x}\vec{w}\vec{y} \in L$. We will now transfer this conception to type theory, using our type-theoretic encoding of strings.

4.13.3 Generalizing the Language-theoretic Context

Take a finite alphabet T, and fix a language $L' \subseteq T^*$. As we have seen in the last section, there is an injective map $i : T^* \to \mathtt{WTT}$ from strings in T^* into the set λ-terms of the signature Θ_T^{string}. We thus obtain $i[L'] \subseteq \mathtt{WTT}$, where $i[-]$ is the pointwise extension of i to sets. We close $i[L']$ under $=_{\alpha\beta}$, and obtain $L := \{\mathtt{m} : \text{there is } \mathtt{n} \in i[L] : \mathtt{n} =_{\alpha\beta} \mathtt{m}\}$.

We now define a context $\mathsf{C}_T(L) = (\mathcal{G}, \mathcal{M}, \mathcal{I})$, where $\mathcal{G} = \mathcal{M} = \mathtt{Tm}_c(\Lambda(\Theta_T^{string}))$, that is the set of closed terms over the signature Θ_T^{string}; and for $\mathtt{m}, \mathtt{n} \in \mathtt{Tm}_c(\Lambda(\Theta_T^{string}))$, we have $(\mathtt{m}, \mathtt{n}) \in \mathcal{I}$ iff $\mathtt{nm} \in L$. So for S a set of terms, we have $S^{\triangleright} := \{\mathtt{t} : \forall \mathtt{s} \in S : \mathtt{ts} \in L\}$, and $S^{\triangleleft} := \{\mathtt{t} : \forall \mathtt{s} \in S : \mathtt{st} \in L\}$.

Definition 135 *A t-concept is a concept (S, T) over the context $\mathsf{C}_T(L)$, where $S = T^{\triangleleft}$, $T = S^{\triangleright}$. The **syntactic t-concept lattice** of a language L is defined as $\mathcal{L}_T(\mathsf{C}_T(L)) =: SCL_T(L) := \langle \mathfrak{B}_L^T, \wedge, \vee, \top, \bot \rangle$, where $\mathfrak{B}_L^T$ is the set of syntactic t-concepts of L, and with all constants and connectors defined in the usual way.*

What we are still missing is an operator which allows us to define fusion and residuation. Recall that for terms, our primitive objects, juxtaposition is interpreted as function application. We extend this interpretation to sets of terms: for $S_1, S_2 \subseteq \mathtt{Tm}_c(\Lambda(\Theta_T^{string}))$, we define $S_1S_2 := \{\mathtt{mn} : \mathtt{m} \in S, \mathtt{n} \in T\}$. Next, for t-concepts $(S_1, T_1), (S_2, T_2)$, we simply put $(S_1, T_1) \circ (S_2, T_2) := ((S_1S_2)^{\triangleright\triangleleft}, (S_1S_2)^{\triangleright})$. That is, as before we use the closure of concatenation of extents to define $\circ$. But there is an important restriction: concatenation of terms is *not* associative. Consequently, the operation $\circ$ is not associative on concepts, we have, for concepts $M, N, O \in SCL_T(L)$, $(M \circ N) \circ O \neq M \circ (N \circ O)$. For example, $M \circ N$ might be $\top$, because MN contains a term $\mathtt{mn} \notin \mathtt{WTT}$, and consequently we have $(M \circ N) \circ O = \top \circ O = \top$. Still, $M \circ (N \circ O)$ might be well-typed. So the structure of $(\mathfrak{B}_L^T, \circ)$ is not a monoid, but rather a groupoid. We furthermore have a left identity element 1_l, such that for every concept S, $1_l \circ S = S$. This is the concept of the identity function $(\{\lambda x.x\}^{\triangleright\triangleleft}, \{\lambda x.x\}^{\triangleright})$. (By the way, the identity function is also the encoding of the empty string $/\epsilon/$). There is no general right identity, though: for assume we have a term $\mathtt{m} : \alpha$ for a constant atomic type α; then there is no term $\mathtt{n}$ such that $\mathtt{mn}$ can be typed. Consequently, no $\mathtt{n}$ can be the right identity for $\mathtt{m}$.

What are the residuals in this structure? Given the fusion operator, they are already implicitly defined by the law of residuation $O \leq M/N \Leftrightarrow O \circ N \leq M \Leftrightarrow N \leq O\backslash M$; what we have to show that they exist and are unique. In the sequel we will use residuals both on sets of terms and on concepts; this can be done without any harm, as the extent order and the concept order are isomorphic. To see more clearly what residuation means in our context, note that for $S \subseteq \mathtt{Tm}_c(\Lambda(\Theta_T^{string}))$, we have $S^{\triangleright} := L/S$; because $S^{\triangleright}$ is the set of all terms

$\mathtt{m}$, such that for all $\mathtt{n} \in S$, $\mathtt{mn} \in L$. Dually, we have $S^{\triangleleft} := S \backslash L$. Consequently, we have $S^{\triangleright\triangleleft} = (L/S)\backslash L$, and dually, we get $S^{\triangleleft\triangleright} = L/(S\backslash L)$. So we see that the polar maps of our Galois connection form a particular case of the residuals, or conversely, the residuals form a generalization of the polar maps. The closure operators are equivalent to a particular case of what is known as *type raising*. More generally, we can explicitly define residuals over a ternary relation: put $(\mathtt{m}, \mathtt{n}, \mathtt{o}) \in R$ if and only if $\mathtt{mn} =_{\alpha\beta} \mathtt{o}$. Then we define

1. $O/N := \{\mathtt{m} : \forall \mathtt{n} \in N, \exists \mathtt{o} \in O : (\mathtt{m}, \mathtt{n}, \mathtt{o}) \in R\}$; dually:
2. $M\backslash O := \{\mathtt{n} : \forall \mathtt{m} \in M, \exists \mathtt{o} \in O : (\mathtt{m}, \mathtt{n}, \mathtt{o}) \in R\}$.

As is easy to see, $M^{\triangleright} := \{\mathtt{n} : \forall \mathtt{m} \in M, \exists \mathtt{o} \in L : (\mathtt{m}, \mathtt{n}, \mathtt{o}) \in R\}$; and $M^{\triangleleft} := \{\mathtt{n} : \forall \mathtt{m} \in M, \exists o \in L : (\mathtt{m}, \mathtt{n}, \mathtt{o}) \in R\}$. This way, we explicitly define residuals for sets of terms. Given this, it easily follows that residuals also exist and are unique for concepts: $(S_1, T_1)/(S_2, T_2) = ((S_1/S_2), (S_1/S_2)^{\triangleright})$.

So residuals allow us to form the closure not only with respect to L, but with respect to any other concept. This provides us with a much more fine-grained access to the hierarchical structure of languages. On the negative side, the $\circ$ operation and residuals do not tell us anything about directionality of concatenation on the string level. This however is unsurprising, as our treatment of strings as λ-terms serves precisely the purpose of abstracting away from this: concatenation is done by terms automatically, and we need no longer take care or even notice of this. Obviously, concepts of $SCL_T(L)$ provide a vast generalization of the concepts over strings in $SCL(L)$. An immediate question is whether this extension is conservative, in the sense that each set closed in $SCL(L)$ is, under the usual translation, closed in $SCL_T(I)$. This is generally wrong, but holds with some restrictions:

Theorem 136 *Let $M, L \subseteq T^*$; let M^λ, L^λ be their type theoretic counterpart in the signature Θ_T^{string}. If $M = M^{\triangleright\triangleleft}$ is closed wrt. the language theoretic context $\mathfrak{C}_C(L)$, then we have $M^\lambda = (M^\lambda)^{\triangleright\triangleleft} \cap (T^*)^\lambda$, where $(M^\lambda)^{\triangleright\triangleleft}$ is closed wrt. the type theoretic context $\mathfrak{C}_T(L^\lambda)$.*

Proof. Let M be c-closed; every string context $(\vec{w}, \vec{v}) \in M^{\triangleright}$ corresponds to a function of the form $\lambda x.\mathbf{B}(\mathbf{B}/\vec{w}/x)/\vec{v}/$, which takes a term $/u/$ as argument, concatenating it with a $/\vec{w}/$ to its left and $/\vec{v}/$ to its right, resulting in a term $/wuv/$. Call the set of these functions $(M^\lambda)^{\blacktriangleright}$. We now take $(M^\lambda)^{\blacktriangleright\triangleleft}$. Obviously we have $M^\lambda \subseteq (M^\lambda)^{\blacktriangleright\triangleleft}$. We show that $M^\lambda \supseteq (M^\lambda)^{\blacktriangleright\triangleleft} \cap (T^*)^\lambda$: if we have, for $\vec{w} \in T^*$, $\vec{w} \notin M$, but $/\vec{w}/ \in (M^\lambda)^{\blacktriangleright\triangleleft} \cap (T^*)^\lambda$, then we have $i^{-1}(/\vec{w}/) \in M^{\triangleright\triangleleft}$, because each type context in $(M^\lambda)^{\blacktriangleright}$ corresponds to a string context in $M^{\triangleright}$. This is a contradiction, as M is closed under $[-]^{\triangleright\triangleleft}$.

So we have $M^\lambda = (M^\lambda)^{\blacktriangleright\triangleleft} \cap (T^*)^\lambda$, and $(M^\lambda)^{\blacktriangleright\triangleleft}$ is a closed set. Furthermore, as $(M^\lambda)^{\blacktriangleright} \subseteq (M^\lambda)^{\triangleright}$, we have (by the laws of Galois connections) $(M^\lambda)^{\blacktriangleright\triangleleft} \supseteq (M^\lambda)^{\triangleright\triangleleft}$. So we get $M^\lambda \supseteq (M^\lambda)^{\triangleright\triangleleft} \cap (T^*)^\lambda$. To see that $M^\lambda \subseteq (M^\lambda)^{\triangleright\triangleleft} \cap (T^*)^\lambda$, consider that as $M \subseteq T^*$, we have $M^\lambda \subseteq (T^*)^\lambda$; furthermore, $M^\lambda \subseteq (M^\lambda)^{\triangleright\triangleleft}$. Therefore, $M^\lambda \subseteq (M^\lambda)^{\triangleright\triangleleft} \cap (T^*)^\lambda$. This completes the proof. □

As expected, the converse does not hold, not even for terms which encode strings. In this sense t-concepts yield a proper generalization of c-concepts. This however does not obtain for the extension of the lattice with fusion and residuals: fusion in the t-concept lattice is completely incomparable to fusion in the c-concept lattice of a language.

4.13.4 Putting Things to Work

From our above argument on $sub(I^\lambda)$, it follows that for a finite language I, $SCL_T(I^\lambda)$ is in general an infinite lattice. Therefore, if we want to put our analogical maps to work, we have to restrict also the context and the concept lattice to $I^{\lambda k}$; but as we will see later on, this only makes sense for the extent, not for the intent! The context we use for a conceptual analogue of $\lambda kP1$ is thus $\mathsf{C}_T(I^{\lambda k})$. We now develop an inference scheme. Recall how we used concepts in the language theoretic setting; we will make a similar approach. We write $\mathcal{C}_t^k(L)$ for the concept of all terms $\mathtt{m} \in I^{\lambda k}$. Again, for $\mathtt{M}, \mathtt{N} \subseteq \mathtt{WTT}$, by $\mathtt{MN}$ we denote the set $\{\mathtt{mn} : \mathtt{m} \in \mathtt{M}, \mathtt{n} \in \mathtt{N}\}$. By $\mathtt{m}[\mathtt{N}]$ we denote the set $\{\mathtt{m}[\mathtt{n}] : \mathtt{n} \in \mathtt{N}\}$. We now define the inference rules denoted by $\mathfrak{g}^{\lambda\mathcal{C}}$.

$$\frac{\mathtt{M} \Leftarrow_I^P \mathtt{NM} \quad \mathtt{m}[\mathtt{M}] \subseteq \mathfrak{g}_P^{\lambda\mathcal{C}}(I)}{\mathtt{m}[\mathtt{NM}] \subseteq \mathfrak{g}_P^{\lambda\mathcal{C}}(I)} \tag{4.60}$$

Note that we slightly abuse notation, because I is here not a language, but a set of terms. In the end, we also need a "spell-out" scheme to get back from sets of terms to terms; we just mention this by way of example, and skip the dual inference rule to infer a set from its element.

$$\frac{\mathtt{m}[\mathtt{N}_1, ..., \mathtt{N}_\mathtt{i}] \subseteq \mathfrak{g}_P^{\lambda\mathcal{C}}(I) \quad \mathtt{n}_1 \in \mathtt{N}_1, ..., \mathtt{n}_\mathtt{i} \in \mathtt{N}_1}{\mathtt{m}[\mathtt{n}_1, .., \mathtt{n}_\mathtt{i}] \in \mathfrak{g}_P^{\lambda\mathcal{C}}(I)} \tag{4.61}$$

This seems satisfactory. What we still need is a "decent" analogical map. Again, we can simply look at string based concepts and transfer what we have worked out there. We can devise $\lambda\mathcal{C}P1$ as follows: We have $\mathtt{N} \approx_I^{\lambda k\mathcal{C}P1} \mathtt{MN}$, if and only if $\mathtt{m}[\mathtt{MN}] \subseteq I^{\lambda k} \Rightarrow \mathtt{m}[\mathtt{N}] \subseteq I^{\lambda k}$. Note that in these rules and with this map, we do talk about (term)-concepts in the narrow sense, but only about the extents, that is, sets which are closed under $[-]^{\triangleright\triangleleft}$. We have done this before, and we proceed with this for notational convenience.

Already the pre-theories on terms are quite abstract; for the pre-theories on concepts over terms it is almost impossible to have an intuition on how they work at this point. So we just present a small example; we go back to our language $I_e := \{a, aa, aaaa\}$. We have $i[I_1] = \{\lambda x.ax, \lambda x.aax, \lambda x.aaaax\}$. Next, we have to fix a k. Let us put $k := 2$, so that we consider the language $I_1^{\lambda 2}$. What does this language look like? We have

$$(I_1)^{\lambda 2} \supseteq i[I_1] \cup I', \text{ where } I' := \{\lambda x.x\mathtt{m} : \mathtt{m} \in i[I_1]\} \cup \{\lambda x.x\mathtt{m} : \mathtt{m} \in I'\}. \tag{4.62}$$

This is the subset of $(I_1)^{\lambda 2}$ which is quite uninteresting. But there is more to it; we also have

$$(I_1)^{\lambda 2} \supseteq \{\lambda x.\mathbf{B}xx(\lambda x.ax), \lambda x.\mathbf{B}xx(\lambda x.ax)\}. \tag{4.63}$$

So what is the concept $\mathcal{C}_T(\lambda x.\mathbf{B}xx)$ in $I_e^{\lambda 2}$? Here we encounter another problem: there is no term in $\mathtt{m}$ such that $\mathtt{m}\lambda x.\mathbf{B}xx \in I_1^{\lambda k}$! So we see in this example that we actually have to build concepts with respect to the entire set $(I_e)^\lambda$! So whereas it is necessary to restrict the intent, it is unreasonable to restrict the possible extents; the context we have to consider is $(\mathtt{WTT}, \mathtt{WTT}, \mathcal{I})$, where $(\mathtt{m}, \mathtt{n}) \in I$, if 1. $\mathtt{nm} \in I^\lambda$, and 2. $\mathtt{m} \in sub(I^{\lambda k})$; but there are no restrictions

on $\mathtt{n}$. It is clear that nonetheless $\mathcal{I}$ is infinite, because there are infinitely many $\mathtt{m}$ satisfying the criteria. Nonetheless, there are only finitely many distinct concepts, because there are only finitely many distinct *extents*, and of course concepts with the same extent are identical.

Let us return to the example. We have a concept of $\lambda x.\mathbf{B}xx$. This exists, because we have $\lambda y.((y(\lambda x.ax)\lambda x.ax)$, $\lambda y.((y(\lambda x.aax)\lambda x.aax) \in \{\lambda x.\mathbf{B}xx\}^{\triangleright}$. It turns out that – at least in the full calculus – this is not unique (up to $=_{\alpha\beta}$), because there are the terms $\lambda xy.x$, $\lambda xy.y$. So we have $[\lambda x.\mathbf{B}xx^{\triangleright\triangleleft}]_{\alpha\beta} = \{\lambda xy.x, \lambda xy.y, \lambda x.\mathbf{B}xx\}$. and there is (up to $\alpha\beta$-equivalence) no other $\mathtt{m}$ such that $\lambda y.((y(\lambda x.ax)\lambda x.ax)\mathtt{m}, \lambda y.((y(\lambda x.aax)\lambda x.aax)\mathtt{m} \in I^{\lambda}$. From this it follows that $\{\lambda x.ax, \lambda x.aax\}^{\triangleright\triangleleft} = [\{\lambda x.ax, \lambda x.aax\}]_{\alpha\beta}$; because any $\mathtt{m} \in \{\lambda x.ax, \lambda x.aax\}^{\triangleright\triangleleft}$ must be in I^{λ}, and $\lambda x.aaaax \notin \{\lambda x.ax, \lambda x.aax\}^{\triangleright\triangleleft}$, as can be easily concluded from our above considerations. Put $\mathtt{N}_1 := \mathcal{C}^k_T(\{\lambda x.ax, \lambda x.aax\})$, $\mathtt{N}_2 := \mathcal{C}^k_T(\{\lambda x.\mathbf{B}xx\})$.

Do we have $(\mathtt{N}_1, \mathtt{N}_2\mathtt{N}_1) \in \lambda 2P1(I_e)$? We can check this manually: assume we have $\mathtt{m}[\mathtt{N}_2\mathtt{N}_1] \in I^{\lambda 2}$. Then there are two possibilities; Let $\beta(\mathtt{m}[\mathtt{N}_2\mathtt{N}_1])$ be the set of β-normal forms of the terms.

1. Assume $\beta(\mathtt{N}_2\mathtt{N}_1) \subseteq sub(\beta(\mathtt{m}[\mathtt{N}_2\mathtt{N}_1]))$. In this case, $\lambda x.aaaax$ is a subterm of $\beta(\mathtt{m}[\mathtt{N}_2\mathtt{N}_1])$, and we must have $\mathtt{m}[\mathtt{N}_2\mathtt{N}_1] =_{\alpha\beta} \mathtt{N}_2\mathtt{N}_1$. Now we have $\mathtt{N}_2\mathtt{N}_1 \subseteq I^{\lambda 2}$, and so we also have $\mathtt{N}_1 \subseteq I^{\lambda 2}$, so the condition is satisfied.

2. Assume $\beta(\mathtt{N}_2\mathtt{N}_1) \not\subseteq sub(\beta(\mathtt{m}[\mathtt{N}_2\mathtt{N}_1]))$. In this case, the abstraction is vacuous and we can just put any term in there: so for any $\mathtt{N}$, we have $\beta(\mathtt{m}[\mathtt{N}_2\mathtt{N}_1]) =_{\alpha\beta} \beta(\mathtt{m}[\mathtt{N}])$. This means in particular: if $\mathtt{m}[\mathtt{N}_2\mathtt{N}_1] \in I^{\lambda}$, then $\mathtt{m}[\mathtt{N}_1] \in I^{\lambda}$. Moreover, as $\mathtt{m}[\mathtt{N}_1]$ has less reduction steps to the unique β-normal form in $i[I_e]$, it follows that $\mathtt{m}[\mathtt{N}_2\mathtt{N}_1]) \in I^{\lambda k} \Rightarrow \mathtt{m}[\mathtt{N}_1] \in I^{\lambda k}$ for all $k \in \mathbb{N}$.

So we actually now do have the analogy which we desired: we know at least $\{a^{2^n}\} \subseteq j \circ \mathfrak{g}^{\lambda}_{\lambda 2\mathcal{C}P1}([I]^{\lambda})$. It would be tedious to show that we also have the converse implication, so we leave it with that. But we see how complicated pre-theories of the kind we investigated here can get – even working on very simple languages. And this is the awkward thing about it.

4.14 Another Order on Pre-Theories

We have so far considered the inclusion order of classes of the form $\mathfrak{C}(\mathfrak{f}, P)$ or $\mathfrak{C}^{\infty}(\mathfrak{f}, P)$. We have seen that these orderings are not very meaningful, because the classes we consider are quite unnatural in the first place; most arguments can be brought back to finite languages because of alphabetical innocence, which in turn makes even results of the form $\mathfrak{C}^{\infty}(\mathfrak{f}, P) \subseteq \mathfrak{C}^{\infty}(\mathfrak{f}', P')$ quite meaningless. We now present a different order, which is maybe more informative and more meaningful. This order corresponds to the order of functions according to their graphs.

Definition 137 *Given two pre-theories $(\mathfrak{f}, P), (\mathfrak{f}', P')$, we say $(\mathfrak{f}, P') \ll (\mathfrak{f}', P')$ if for all finite languages I, we have $\mathfrak{f}_P(I) \subseteq \mathfrak{f}'_{P'}(I)$. We write $P \ll P'$, if for all finite languages I, $P(I) \subseteq P'(I)$, and we write $\mathfrak{f} \ll \mathfrak{f}'$, if for all analogical maps P, we have $\mathfrak{f}_P(I) \subseteq \mathfrak{f}'_P(I)$.*

It is easy to see that the relation $\ll$ is completely independent of the order $\subseteq$ or $\subseteq_{\infty}$ on $\mathfrak{C}(\mathfrak{f}, P), \mathfrak{C}(\mathfrak{f}', P')$. For example, there is an analogical map which is maximal according to $\ll$, namely P_{max}, where for all finite $I \subseteq \Sigma^*$, $P_{max}(I) = \Sigma^* \times \Sigma^*$. Of course, for $I \neq \emptyset$, we then have $\mathfrak{f}_{P_{max}}(I) = \Sigma^*$ for any reasonable

set of inference rules $\mathfrak{f}$. The order $\ll$ is transitive and reflexive; note however that $\ll$ is not antisymmetric: from $(\mathfrak{f}, P) \ll (\mathfrak{f}', P')$, $(\mathfrak{f}', P') \ll (\mathfrak{f}, P)$ it does not follow that $P = P'$, $f = f'$. In that case, we can however say that pre-theories are equivalent: we do not see a difference from the languages induced. We can thus transform this pre-order to a partial order by looking at pre-theories modulo equivalence. Note that $\ll$ for P, P' on $\Sigma^* \times \Sigma^*$ is in fact antisymmetric. But firstly, for us languages are more interesting than analogies. And secondly, the underlying objects of analogies are not always be comparable: how should we compare simple, powerset, or type-theoretic pre-theories. So what is more interesting is the order on pre-theories.

So $\ll$ does have a maximal element $(\mathfrak{f}, P_{max})$ for sets of rules $\mathfrak{f}$ for strings we considered, and which is unique up to equivalence. On the other side, for the relation $\subseteq$ and on $\mathfrak{C}(\mathfrak{f}, P)$ and $\mathfrak{C}^\infty(\mathfrak{f}, P)$, there is no maximal element, as we will see in the next section. The order $\ll$ is somewhat more easy to put to use on pre-theories. Note that if $P \ll P'$, then $(\mathfrak{f}, P) \ll (\mathfrak{f}, P')$. We can easily derive the following results:

Lemma 138 *1.* $Pr \ll P1$
2. $(\mathfrak{g}, P1) \ll (\overline{\mathfrak{g}}^{late}, \mathcal{P}P1) \ll (\overline{\mathfrak{g}}^{late}, FP1)$
3. $(\mathfrak{g}, Pr) \ll (\overline{\mathfrak{g}}^{late}, \mathcal{P}Pr) \ll (\overline{\mathfrak{g}}^{late}, FPr)$

Proof. 1. As the conditions for $P1$ are a subset of the Pr-conditions, for all finite I, we have $Pr(I) \subseteq P1(I)$.

2. $P1 \ll \mathcal{P}P1$ is quite trivial, as substitution is a polynomial function. Moreover, every polynomial function is computable. Moreover, in the scheme $\overline{ga}$ we can simulate string substitution with the introduction of terms.

3. See 2. □

So there is a number of immediate results we get. Another one we have already proved, but repeat for convenience is the following:

(**Theorem 101**) We have $(\mathfrak{g}, P1) \ll (\mathfrak{g}^{\mathcal{C}}, \mathcal{C}P1)$ and $(\mathfrak{g}, Pr) \ll (\mathfrak{g}^{\mathcal{C}}, \mathcal{C}Pr)$.

□

So we have here a formal foundation for the use of concepts in pre-theories: they serve to make pre-theories larger in the $\ll$-order. These results are somewhat partial and preliminary: further investigation in this direction seems to be promising, and we might obtain more general results without too much effort. This however presupposes much more abstract notions of pre-theories and their correspondence, as in the above example. We leave this open for further research.

4.15 A Kind of Completeness

Before we close this section on the classical metatheory of language, there are two important general results which I have to present. These do not concern particular pre-theories, but rather pre-theories in general, what they can do, and what they cannot do.

The first result is a sort of "completeness" result, which can be stated as follows: recall that we are looking for projections, that is, maps from finite to infinite languages; pre-theories were a particular "operationalization" of this concept, a way to approach these functions. The first result states that this interpretation comes without loss of generality: every (computable) projection

function can be described as a pre-theory. So we can equate projections and pre-theories. This also justifies our use of "implicit definitions" of pre-theories, where we simply defined them by the projection to which they give rise.

The second main theorem on the other hand is a sort of "incompleteness" result: technically, it shows that given an alphabet Σ, there is no computable, surjective function from the finite languages over Σ onto the recursive infinite languages over Σ. This means, rather philosophically speaking: for every decidable pre-theory/projection there is a "blind spot"; there are always languages which – being recursive and infinite – are philosophical candidates for "language", but which our pre-theories cannot induce, not given any finite language.

We will now introduce the *most general* inference rules. It is a scheme which we have not adopted before, as it does not seem to be particularly interesting from a linguistic point of view. It is however very interesting from a mathematical point of view, as it can – with a suitably adapted pre-theory – simulate any other set of inference rules. As we have said, inference rules always "pass down" some kind of property; we have illustrated this in the beginning by the function f. In this illustration, we can infer from $f(\vec{w}) \in L, \vec{w} \Leftarrow \vec{w}'$ that $f(\vec{w}') \in L$. Now, the most general inference scheme is exactly the case where f is simply the identity function. We thus assume simple, string based analogies as usual of the form $\vec{w} \Leftarrow \vec{w}'$, and inferences of the form:

$$\frac{\vec{w} \in I \quad \vec{w} \Leftarrow \vec{w}'}{\vec{w}' \in I}, \qquad \frac{\vec{w} \approx_I^P \vec{w}'}{\vec{w} \Leftarrow_I^P \vec{w}'} \tag{4.64}$$

Denote this scheme by $\mathfrak{mg}$ (so it is this tree, and the usual trees we need for any pre-theory). It would be tedious to show that for every pre-theory $(\mathfrak{f}, P)$ there is an extensionally equivalent pre-theory $(\mathfrak{mg}, P')$. As all pre-theories (are intended to) define projections, this however follows from the main theorem of this section.

Recall that a map $f : \wp(M) \to \wp(M)$ is *increasing*, if for all $X \subseteq M$, we have $f(X) \supseteq X$. We have said a projection $\pi : \wp(\Sigma^*) \to \wp(\Sigma^*)$ is a map such that $\pi(L) \supseteq L$, $\pi(L) = L$ if $|L| = \omega$, and there is at least one I such that $|I| < \omega$ and $|\pi(I)| = \omega$.

So projections are increasing maps on languages. Finally, we have also required that projections be computable. This is a trivial requirement if the image is a finite language. But if $\pi(I)$ is an infinite language, this is problematic, because we cannot just write out $\pi(I)$; what we rather need is an algorithm which either enumerates $\pi(I)$, or an algorithm which decides for any $\vec{w}$ whether $\vec{w} \in \pi(I)$. So we must read $\pi(I)$ in two senses: 1. $\pi(I)$ as an infinite language; we will denote this in critical cases by $\|\pi(I)\|$, or by the *extension of* $\pi(I)$. 2. $\pi(I)$ as a finite characterization of $\|\pi(I)\|$; we will call this the intension of $\|\pi(I)\|$. A very satisfying solution is to require that π maps I onto the code of a Turing machine recognizing $\|\pi(I)\|$. For simplicity, we omit the "code of" part and act as if π directly maps I onto a Turing machine. So intensionally speaking, a projection map is a map $f : \wp(\Sigma^*) \to \mathfrak{TM}$, where $\mathfrak{TM}$ is the class of all Turing machines (recognizing a language over the alphabet Σ), and where $f(I) = TM$ only if $L(TM) \supseteq I$.

All we need for our main theorem is that π be increasing and computable in the above sense, and for mathematical purposes, we formulate the result in the most general fashion; but of course, *a fortiori* the same also holds for projections in a

more restricted sense (satisfying requirements such as alphabetical conservativity, closure under isomorphism etc.).

Of course, it would be quite tedious to let a pre-theory construct a Turing machine. We therefore take another characterization of languages. Assume we have a pre-theory $(\mathfrak{f}, P)$ and a language I. We can associate with the extensional deductive closure $\mathfrak{f}_P(I)$ a finite, intensional characterization of the infinite language. This characterization is a triple $(\mathfrak{f}, I, R))$, where $\mathfrak{f}$ is a set of inference rules of the type we have used above, I is the input language and the set of premises we have for our inferences, and R is a recursive relation which equals $P(I)$. Importantly, a relation $R \subseteq M \times N$ has to be **recursive**; that is, for all $(m, n) \in M \times N$, we can decide whether $(m, n) \in R$ or $(m, n) \notin R$. R has to be recursive rather than recursively enumerable, because if R is only recursively enumerable, we cannot guarantee that the resulting deductive closure is recursively enumerable. So we will, given a TM, we construct a triple $(\mathfrak{f}, I, R)$ which characterizes the same language.

It is obvious that if I is finite, R is recursive, this procedure results in a recursively enumerable language. What our "completeness" shows is that also the converse obtains: we can simulate any TM with a triple $(\mathfrak{f}, I, P(I))$ satisfying the above requirements.

Theorem 139 *Let $f : \wp(\Sigma^*) \to \wp(\Sigma^*)$ be a computable, increasing map, such that $f(L) = L$ if L is infinite. Then there is a pre-theory $(\mathfrak{mg}, P)$ such that for every finite language I, $f(I) = \mathfrak{mg}_P(I)$.*

Proof. Assume $\|f(I)\| = L$, where $f(I) = TM$. Then L is recursively enumerable. Thus there is a computable bijection $\delta_{f(I)} : \mathbb{N} \to L$, for which there is a $k \in \mathbb{N}$ such that $(\delta_{f(I)})^{-1}[\{i : i \leq k\}] = I$. We now show how this bijection $\delta_{f(I)}$ can be obtained; the crucial point is to show that it is recursive rather than recursively enumerable.

Put $order_{TM}(\vec{w}) := |\vec{w}| + C(\vec{w})$, where $C(\vec{w})$ is the number of computation steps TM needs in in order to accept $\vec{w}$. Note that we assume that $order_{TM} : \Sigma^* \to \mathbb{N}$ only assigns finite order; so if we have $\vec{w} \notin L(TM)$, $order_{TM}(\vec{w})$ is undefined (or ω). Importantly, 1. it is decidable whether $order_{TM}(\vec{w}) = k$, 2. for every $k \in \mathbb{N}$, the set $\{\vec{w} : order_{TM}(\vec{w}) \leq k\}$ is recursive (rather than recursively enumerable). We can make this order antisymmetric by ordering strings of the same order by *rad*; this gives the linear order $linord_{TM} \subseteq \Sigma^* \times \Sigma^*$, which is defined by: $\vec{w}\ linord_{TM}\ \vec{v}$ iff and only if

1. $order_{TM}(\vec{w}) < order_{TM}(\vec{v})$, or
2. $order_{TM}(\vec{w}) = order_{TM}(\vec{v})$, and $\vec{w}\ rad\ \vec{v}$.

Note that by definition of *rad*, this order is irreflexive. Now we define $\delta_{f(I)} : \mathbb{N} \to \Sigma^*$ by $\delta_{f(I)}(n) = w$ iff $|\{\vec{v} : \vec{v}\ linord_T M \vec{w}\}| = n - 1$, mapping n on the nth element of the *linord*-order. This is a computable function.

Thus there is a computable function $j_{\delta_{f(I)}}$ such that for all $i \in \mathbb{N}$, $j_{\delta_{f(I)}}(\delta_{f(I)}(i)) = \delta_{f(I)}(i + 1)$. This map is also recursive, as long as it is restricted to strings in $\|f(I)\| = L$. So we have $|j_{\delta_{f(I)}}| \subseteq \Sigma^* \times \Sigma^*$. This is easily adapted such that there is a $k \in \mathbb{N}$ such that $(\delta_{f(I)})^{-1}[\{i : i \leq k\}] = I$.

As we have established this, we can simply define the analogical map P_f by

$$P_f : I \mapsto j_{\delta_{f(I)}}.$$

With the most general inference rules, we obtain then $\mathfrak{mg}_{P_f}(I) = L$, because every string $\vec{w} \in L$ is derivable by $(j_{\delta_{f(I)}})^n(\vec{v})$ for some $\vec{v} \in I$ and $n \in \mathbb{N}$. Moreover, every string in $\mathfrak{mg}_{P_f}(I)$ is in L, as we can show by an easy induction.

So we have $L(\mathfrak{mg}, I, P_f(I)) = L(TM)$. □

So actually, for every projection we can devise a pre-theory. This mans that our somewhat special treatment - due to linguistic intuitions - of projections comes with no loss of generality.

Actually, this proof has an interesting corollary, which we state because it will have some importance in the proof of the next main result:

Corollary 140 *Let L be recursively enumerable. Then 1. there is a recursive bijection $\delta_L : \mathbb{N} \to L$, and 2. a well-order $\leq \subseteq L \times L$, where for every $\vec{w} \in L$, we can effectively compute the immediate $\leq$-successor.*

Proof. Just take the map δ_L as bijection, and j_{δ_L} as well order. Both results follow from the last proof: if we know that $\vec{w} \in L$, we can also effectively compute $(\delta_L)^{-1}(\vec{w})$; and consequently, compute $\delta_L(((\delta_L)^{-1}(\vec{w})) + 1)$. □

So we can effectively enumerate strings in a given order, and consequently, there is a well-order $\leq \subseteq L \times L$, such that there is an algorithm which for any $\vec{w} \in L$, provided there is a $\vec{v} \in L$ with $\vec{w} \leq \vec{v}$, gives us a $\vec{v}' \in L$ with $\vec{w} \leq \vec{v}'$ after a finite number of steps.

4.16 A Kind of Incompleteness

So far, we have seen plenty of pre-theories, where for most of them we could make statements of the form: "for the pre-theory $(\mathfrak{f}, P)$, there is a class of languages C such that for no $L \in C$, there is a finite I such that $\mathfrak{f}_P(I) = L$." So we knew there are languages which cannot be induced in any way. From the point of view of linguistic theory, this is what makes a formalism interesting and relevant: because we know it can be falsified by some "languages" (at least in my view, though not in everyones view). If we use a framework for linguistic theory, we assume that "language" is given. If the framework is restrictive, the linguistic question is: is it adequate to model "language"? This is then supposed to be an empirical question, which can be answered in some way, though the answer will always remain preliminary, because we can never have all possible "languages". However, if it is not adequate, we have learned something about natural language, and if it seems impossible to falsify it, we have also learned something about language. If the formalism is on the other side not restrictive, none of these obtains.

For a linguistic metatheory, things are very different. A pre-theory cannot be falsified, and there is no empirical content to it, except for the checking against partial languages. So for a pre-theory, being unable to induce certain languages is not a merit, making empirical predictions, but is rather the formal counterpart to a methodological bias: we are blind to certain patterns. It entails that there are certain statements on "language" which cannot be falsified given any amount of any data, because of intrinsic properties of our pre-theory. So whereas from a linguistic point of view, there is good reason to be unable to describe certain languages; but from a metalinguistic point of view, there is very good reason to be able to induce *any* language.

In a word, we should be eager to devise a pre-theory which can induce any language which is recursive. On the other side, we should also devise it in a way such that it only induces recursive languages: otherwise, adequacy becomes an undecidable problem. We have already considered the "most general pre-theory", and seen that it can model any computable increasing map on languages, so *a fortiori* every projection. This tells us that we can safely just look at increasing maps/projections instead of pre-theories. For us, the question of completeness ("do we induce all mathematically possible languages") can actually be reduced to *infinite* languages, because regarding the images of our pre-theories, we are not interested in finite languages. So the question is:

Given an alphabet Σ, is there a computable increasing map $\pi : \Sigma^* \to \Sigma^*$, such that 1. for every finite $I \subseteq \Sigma^*$, $\pi(I)$ is recursive, and 2. for every infinite recursive language $L \subseteq \Sigma^*$, there is a finite language $I \subseteq \Sigma^*$, such that $\pi(I) = L$?

Note that in this statement, increasing map and projection can be interchanged arbitrarily. Our "incompleteness" result is the following: we will answer this question negatively. There is no pre-theory which induces all and only infinite recursive languages. We call this general result *incompleteness* with the following motivation. We might define a pre-theory $(\mathfrak{f}, P)$ as *complete*, if for every infinite recursive language L, there is a finite I such that $\mathfrak{f}_P(I) = L$, and every language L such that $\mathfrak{f}_P(I) = L$ for some finite L is recursive. Our result then says: there is no complete pre-theory. As this result is of some general relevance, our first formulation of it will be somewhat more general. We can also relativize this concept, saying: given a class of languages C, a map $f : \wp(\Sigma^*) \to \wp(\Sigma^*)$ is *complete wrt. C*, if for every language $L \in C$, there is a finite I such that $f(I) = L$, and for every finite I, $f(I) \in C$.

Theorem 141 *(Incompleteness) For every increasing, computable map $f : \wp(\Sigma^*) \to \wp(\Sigma^*)$ such that for all finite $I \in \wp(\Sigma^*)_{fin}$, $f(I)$ is recursive, there exists a recursive language L, such that there is no finite $I \in \wp(\Sigma^*)$ with $f(I) = L$.*

The proof is based on an idea of diagonalization, where in our case computability plays a crucial role.

Proof. Take an increasing, computable map $f : \wp(\Sigma^*) \to \wp(\Sigma^*)$ for a (non-empty) alphabet Σ. Fix a well-founded linear order on Σ^*, say, the radix order *rad*. Now take the singleton set $\{\vec{a}_0\} =: A_0$, where $\vec{a}_0 = min_{rad}\Sigma^+$ is minimal wrt. *rad*. Next, construct $f(\{\vec{a}_0\})$. Now take the immediate *rad*-successor of $\vec{a}_0$; call it $\vec{b}_0$. As $f(\{\vec{a}_0\}$ is recursive, we can decide whether $\vec{b}_0 \in f(\{\vec{a}_0\})$ or not. Now assume $\vec{b}_0 \in f(\{\vec{a}_0\})$; in this case we define $\vec{a}_1$ to be the immediate *rad*-successor of $\vec{b}_0$. Conversely, in case $\vec{b}_0 \notin f(\{\vec{a}_0\})$, we put $\vec{a}_1 := \vec{b}_0$.

Then, we put $A_1 = \{\vec{a}_0\} \cup \{\vec{a}_1\}$. This was in a sense an induction base. We now construct successor sets A_{n+1} for any set A_n which has been constructed in this fashion.

Assume we have a given set A_n. Now we define $\vec{a}_{n+1}$ to be the *rad*-smallest string in Σ^*, such that

1. for all $\vec{a} \in A_n$, $\vec{a}$ *rad* $\vec{a}_{n+1}$ (*rad* is not reflexive!)
2. For every $X \in \wp(A_n)$, we have either $\vec{a}_{n+1} \notin f(X)$, or there is an $\vec{a}'$, such that $\vec{a}' \in f(X)$, $max_{rad}(A_n) rad\ \vec{a}'$, and $\vec{a}'$ *rad* $\vec{a}_{n+1}$.

We have to show that for every finite set A_n, $\vec{a}_{n+1}$ exists; uniqueness follows from the linear well-order *rad*. As A_n is finite, $\wp(A_n)$ is finite. Put $max_n := max_{rad}(A_n)$. Consider the immediate *rad*-successor of max_n, say $\vec{x}$. Assume a) forall $X \in \wp(A_n)$, we have $\vec{x} \notin f(X)$. Then $\vec{a}_{n+1} = \vec{x}$. Conversely, assume b) there is $X \in \wp(A_n)$ with $\vec{x} \in f(X)$. Then we just move on to consider the immediate *rad*-successor $succ_{rad}(\vec{x})$; but now we have to check the condition a) only for $\wp(A_n) - X$. We iterate this, such that we always have to check for strictly smaller subsets, and as $\wp(A_n)$ is finite, at some point we will satisfy a).

Now as we know that for any finite set A_n, $\vec{a}_{n+1}$ uniquely exists, we simply put $A_{n+1} := A_n \cup \{\vec{a}_{n+1}\}$, which is again a finite set. Next, we define $A_\omega := \bigcup_{n\in\mathbb{N}} A_n$. A_ω is infinite, because we always have $A_n \subsetneq A_{n+1}$. We have to show two things to prove our theorem.

1. A_ω is recursive.

2. A_ω is not induced by any of its finite subsets under f.

1. A_ω is recursive. What we first have to show is that given any finite A_n, $\vec{a}_{n+1}$ can be computed in a finite number of steps. The procedure we have indicated above consists firstly in checking whether for a string $\vec{a}$, we have $a \in f(X)$. As $f(X)$ is recursive, we can decide this in a finite number of steps. Now as A_n is finite, we have to check this for $\vec{a}$ and a finite number of sets $X \in \wp(A_n)$. So for each candidate $\vec{a}$, the procedure is finitary. Moreover, from the above considerations it follows that we will find an $\vec{a}_{n+1}$ for A_n after checking a finite number of candidates.

So more than existent, for each A_n, $\vec{a}_{n+1}$ is effectively computable. It follows immediately that A_ω is recursively enumerable. But moreover, we have an enumeration which proceeds in line with the well-founded linear order *rad*: we enumerate $\vec{a}_0, \vec{a}_1, \vec{a}_2, ...$, and if $m < n$, then we know that $\vec{a}_m \; rad \; \vec{a}_n$. So in order to decide whether $\vec{w} \in A_\omega$, we just enumerate A_ω until we reach a first string $\vec{v}$ such that $\vec{v} \in A_\omega$ and $\vec{w} \; rad_{refl} \; \vec{v}$, rad_{refl} being the reflexive closure of *rad*; it is obvious that this condition can be checked. This can be done in a finite number of steps (well-foundedness of *rad*, effectiveness of computing $\vec{a}_{n+1}$ from A_n). At this point, we either have $\vec{w} = \vec{v}$ and so $\vec{w} \in A_\omega$, or $\vec{w} \neq \vec{v}$, and consequently, $\vec{w} \notin A_\omega$.

2. A_ω is not induced by any of its finite subsets under f.

Assume $A_\omega = f(I)$, where $I \subseteq A_\omega$ and $|I| \leq k : k \in \mathbb{N}$. Then there is a smallest integer $i \in \mathbb{N}$ such that $I \subseteq A_i$, and thus $I \in \wp(A_i)$. But then by construction, there is a string $\vec{a}_{i+1} \in A_{n+1}$ such that either (i) $\vec{a}_{n+1} \notin f(I)$, or (ii) there is a string $\vec{a}$ such that $max_{rad}(A_i) \; rad \; \vec{a} \; rad \; \vec{a}_{i+1}$, where $\vec{a} \in f(I)$ and $\vec{a} \notin A_{n+1}$.

Consider case (i): as $A_{n+1} \subseteq A_\omega$, $\vec{a}_{i+1} \in A_\omega$, while $\vec{a}_{i+1} \notin f(I)$ – contradiction. Consider case (ii): by construction, all strings $\vec{w}$ in $A_\omega - A_{n+1}$ satisfy: $max_{rad}(A_{n+1}) \; rad \; \vec{w}$. As $\vec{a} \; rad \; \vec{a}_{i+1}$, $\vec{a}_{i+1} \in A_{i+1}$, we have $\vec{a} \notin A_\omega - A_{i+1}$; but also, $\vec{a} \notin A_{n+1}$, so $\vec{a} \notin A_\omega$, while $\vec{a} \in f(I)$ – contradiction. □

This theorem is in my view not only quite interesting in general; it is also of fundamental importance for linguistic metatheory. We will therefore present an alternative formulation, which underlines its relevance for our purposes.

Corollary 142 *For every decidable pre-theory* $(\mathfrak{f}, P)$*, there is are infinitely many infinite recursive languages* $L_i : i \in I$ *such that for every* L_i *there is no finite language* I *such that* $\mathfrak{f}_P(I) = L_i$.

Proof. Basically, this restates the above theorem. We obtain the stronger claim that there are infinitely many unobtainable languages by a slight modification of the proof: instead of choosing $\vec{a}_{i+1}$ as smallest successor of A_i, we can choose it as $s(i)$th successor, for any computable sequence s; and there are infinitely many such sequences. □

So how should we interpret this result? In one interpretation, it is clear why we have given it the great name of "incompleteness", though it has technically little to do with the matter of completeness in logic. The reason is that conceptually, it is very similar to the importance of completeness in early metamathematics. For the Hilbert-programm, the completeness of a logic for a formal system (that of arithmetics) basically meant that there is a complete formalization of that system; every statement can be formally proved or disproved, because the logic came together with a proof theory. Gödel's incompleteness result then stated that this is not possible in general; we cannot entirely formalize mathematics (arithmetics) by means of a formal proof-theory. Our result can be interpreted in a similar fashion: there is no pre-theory which allows us to look at "language" without any methodological bias. There will always be patterns to which we are blind.

In one view, one could say that this is a very negative result, as there is no "master"-pre-theory; there will always be something unsatisfying about it. But note that Gödel's incompleteness theorem, though it was the fall of Hilbert's programm of the complete formalization of mathematics, has also a positive reading: under this reading, it says that the creativity of the mathematician is beyond any formal proof system. For us, there might be a similar reading: even though pre-theories are a formalization of the linguists reasoning on language - in the same way as mathematical logic is a formalization of mathematical reasoning - we cannot finally replace the linguist. There will always be some doubt about the correct pre-theory. Pre-theories will (hopefully) constitute a useful tool in linguistic reasoning, but they cannot ultimately defy the creativity and intuition of the linguist. The underlying, intuitive reason could be said to be: the nature of the data we observe influences what we consider to an adequate, meaningful pre-theory. In my view, this is a positive result.

Chapter 5

The Intensional Metatheory of Language

Summary of the Intensional Procedure

In the intensional procedure, we devise a set of extensive functions from finite languages to *finite* languages. We base these on pre-theories and the notion of atomic derivations. Then we gather some positive data and construct negative data. Next, based on whatever additional information we have, we choose a subset of the positive data, which we declare to be the immediate language (i-language). This i-language is then used to construct the intensional language. The intensional language is then tested for adequacy with respect to the entire collection of positive data and the negative data. If it is adequate, we are done; if not, there are several choices: either (i) we change the extensive functions, or (ii) we change the choice of i-language, or (iii) we collect more positive and/or construct less negative data, before we then choose a new i-language. Then, we repeat the procedure.

Again we illustrate the availability of information by using two persons: the metalinguist devises his extensive functions. The linguist first gathers positive data and constructs negative data. Then he chooses – based on whatever information – a subset of the positive data as i-language. He hands the latter to the metalinguist, who then constructs the intensional language and gives it back to the linguist. Then the linguist checks for adequacy wrt. his full positive and negative data. If the intensional language is adequate, they are done; if not, either (i) the linguist complains to the metalinguist to change his functions, or (ii) he changes the choice of i-language, or (iii) he collects more positive and/or constructs less negative data, before he chooses a new i-language and goes back to the metalinguist.

Note that we now have considerably more freedom of choice, because the linguist can choose his i-language freely from his observations, based on linguistic criteria. This also means: if the linguist has an adequate intensional language and makes a new observation which is already contained in his full intensional language, he does not need to worry: he can just claim that this observation would not have been in i-language anyway. But this freedom comes at a price: when testing adequacy, we have to make sure the resulting intensional language comprises *all* positive observations we have made. Therefore, we can exclude some positive data from projection, but have to make sure they figure in our final intensional language.

5.1 Problems of the Classical Conception

As we have said, linguistics in any modern sense is about *possible* utterances. So there is an intensional aspect to linguistics we cannot avoid. However, in the classical approach, once we have fixed "language", there is nothing intensional left: "language" is nothing but an infinite set. In the intensional paradigm, we want to have an intensional model of language in a proper sense. If we assume the cognitive (conceptual) perspective on language, the classical conception is a claim on the mind of the speaker: if we consider "language" as infinite set as being represented in the mind/brain of speakers of language, we find that we accept two important consequences, which are by no means innocent, and which in our view can only be maintained at a considerable price: for reasons of principle

1. for each string of words $\vec{w}$, every speaker "knows" at every time that either $\vec{w} \in L$ or that $\vec{w} \notin L$; and

2. for all strings $\vec{w}$, he *knows this in the same way.*

By "knows" we mean knowing in the same sense, in which we say that a speaker knows his language. We mark this explicitly, as it is by no means clear what kind of knowledge that is (as this is actually a critical point, Chomsky sometimes uses the term more neutral "cognizes". See [47] for a praise of Chomskyan wisdom).

The first point is the strong claim, that for any sentence, we know whether it is in our language or not; even if we have to sit down for three hours with paper and pencil to come to this decision – if we are able to come to a decision it at all, and if we want to count that what we did in the three hours as linguistic understanding. So the process of reasoning whether an utterance belongs to our "language" is a process of recognition, which deterministically yields either a positive or a negative answer. For example, consider the sentence:

(1) Who did the man the mouse the cat chased saw see?

Now assume that two years ago, I thought about this sentence, and came to the conclusion that it is English. In the meantime however, I have changed my mind – I only consider English what I can immediately understand, because I have been converted from a generative grammarian to a "usage-based" grammarian. In the classical paradigm, technically we would either have to admit that my "language" has changed, whereas most people would agree that it is only my attitude on language which has changed – or we would have to admit that my judgments are actually unrelated to "language" in the relevant sense. The latter is the standard move, as we will see, but for us, there is good reason to renounce to it.

Just for the sake of comparison, consider the following line of reasoning: assume we make the assumption, that we are able to recognize valid mathematical reasoning always and infallibly. We can make this assumption, and it is often made for example in constructive mathematics. However, from there it is a long way to saying that we "know" every theorem, simply because we recognize every proof. This is an assumption no one has ever seriously made, at least to my knowledge. So even if we assume that speakers reason infallibly, there is still a long way to go to claiming they "know" every result.

Usually, linguists reject the kind of argument we presented here in its entirety. It is very instructive to see on what grounds precisely they do so. The usual move is to say: linguistic knowledge is implicit, whereas mathematical knowledge is explicit, and the explicit reasoning leads us outside of the realm of linguistics. Our answer is: well, the matter of knowledge being implicit and intuitive only works for a small fragment of what we consider to be "language". The linguists answer would then probably be: fine, but we have to make the distinction between acceptability and grammaticality; we might go outside of acceptability, but not grammaticality. Interestingly, the reasoning of the speaker does not have any relevance for either of the two: it is irrelevant to acceptability, because it is not intuitive and immediate, and it is irrelevant to grammaticality, because the latter is a notion defined by the theoretical linguist, not by reasoning speakers. But our answer to the linguist will be the following: we see, but the move you made is not legitimate for us: we have the priority of epistemic concerns, and

your move consists in moving the proper subject – grammaticality – beyond what we can know for sure. We can now only *define* it, and have lost the empirical access to it. This hurts our fundamental assumption of the priority of epistemology.

The second point above is the strong claim that there is no way in which some utterances are more derived than others, and others are more fundamental. So whether I immediately understand an utterance, or I take two hours, again really makes no difference for the fact how this utterances belongs to "language"; it is a side issue which is considered to have no relevance with respect to linguistic theory. This point is strongly related to the former, and the reason why it is problematic similar: we have to make a strict separation between acceptability and grammaticality, as long as we do not assume that "language" coincides with o-language (which leads to the finitary meta-theory).

NB: the problem of defining "language" in a satisfying manner was *addressed* by the previous section; but is was *not* solved, and it was clear from the beginning that there *cannot* be a solution to our epistemic problems: because at best, we can *define* "language" in a satisfying way, but nonetheless, we do not have any empirical access to this notion.

Note that the problems we sketched here do not arise from explicit assumptions, but follow from the simple fact that we regard languages as infinite sets. The reason is that sets have the obvious properties: 1. for each set S and object o, we have $o \in S$ or $o \notin S$ (definiteness), 2. we have no other information on the relation between o and S; put differently, if for all objects o, $o \in S_1 \Leftrightarrow o \in S_2$, then $S_1 = S_2$ (extensionality). These two points are the reason why we can identify sets with their characteristic function, and everything we have said in this section follows from the simple assumption, that languages as sets (sets of strings, or, as many linguists prefer, sets of trees) are an adequate model of language.

Again, there is a rather old objection: "the set of utterances is the most uninteresting part of "language"; we want to have the intensional description". And again, we will answer: that move is not legitimate for us, because it moves the focus from an epistemologically remote object to an even more remote object; so from our perspective, it is like saying: "No problem the grapes are too high up, I wanted the clouds anyway". So we see that already the commitment to the priority of epistemic concerns over ontological concerns leads us to the intensional metatheory in a natural way, and prevents us from doing most of the standard moves.

5.2 The Intensional Conception: Philosophical Outline

So what is the position of the intensional metatheory? We make a very compact statement here, which we will explain in the sequel. Firstly, from the intensional point of view, the reasoning speaker is relevant for linguistics. That leads us to intensional languages: languages, in which the process of reasoning figures. Furthermore, as in principle, the speaker can reason as much as any linguist (in fact, the linguist is a speaker as well), there is no reasoning which *a priori* a speaker cannot perform, nothing which is impossible in principle. One main

question of intensional linguistics is however: what *do* speakers actually infer when they (mainly) speak, and what *do* they actually infer when they write – that is: which inferences do they actually perform, and which ones can be made "on the spot"? From this it follows that 1. it is not the linguists task to fix the speakers language in a very restrictive sense: intensional language are quite open, and provide more than speakers need to infer or ever do infer; they rather implement possible choices or more generally, possibilities of inferences. 2. The methods of classical metalinguistics become the methods of intensional linguistics. The main problem is the following: it is easy to criticize languages as sets; it is difficult to provide an alternative. So what we mainly undertake here is to lay out the philosophical and ontological foundations of the alternative conception.

In the classical approach, linguistic creativity is seen to be *inherent* in the knowledge of language. That is to say, the entire creativity is already *contained* in my knowledge, and once I have it, there is no way to transcend it, unless I change my language. Infinity makes sure there is no need to do so: once we learn a language, we already have an infinity of objects at our disposition, and this infinity contains any possible creativity we might observe. So the philosophical assumption that creativity is inherent in our knowledge of language coincides with the (methodological) assumption that languages are infinite sets. There is an alternative account of linguistic creativity. The alternative conception is that linguistic creativity is *transcendental*. That is to say: when we are creative in language, we transcend our basic knowledge, adding something genuinely new to it. In this view, language is *open*, in the sense that not all possible utterances are covered by the basic knowledge of language: speakers can go *beyond* it. That is also to say that there is a proper difference between just using language on the one hand, and being creative in language on the other: I can also use language without being creative at all.

So what should then "languages" look like? The underlying intuition for the structure of "language" is as follows: speakers have a (finite) amount of utterances, which they know in some immediate sense; for example, we could say that they know them *verbatim*, that is to say, we are literally acquainted with them (a similar idea has been pursued in logic by [39]). Call this language i-language; empirically, it is the set of strings, which we immediately understand and accept as grammatical (so i is for immediate). An alternative description would be: they are the utterances whose grammaticality cannot be reduced to any other utterances. Note that i-language is theoretically different from any other notion we have introduced so far: i-language can be equal to an observed language only by assumption; we can never be sure neither that we have observed all utterances we immediately know, nor that the utterances we have observed are part of i-language, as speakers *can* be creative. There is also a fundamental difference in cardinality: whereas observed languages are assumed to be *unbounded* – though finite – , that is, there is no upper bound to their size, it is unreasonable to assume that there is no upper bound to i-language; our immediate knowledge is limited by quite rigid constraints. So the upper bound to i-language is given by cognitive restrictions, as i-language *is* a cognitive notion; the bounds to observable language are given by practical restrictions to collect data. So we have to keep i-language and observed languages separate. Neither can we equate i-language with o-language, though in this case, there is a clear inclusion: every utterance in i-language us observable. The converse

cannot obtain, as o-language by our fundamental assumptions is infinite, whereas i-language has to be finite by assumption: having an infinitude of utterances we immediately know would make the concept pointless. For immediate knowledge, we have to presuppose acquaintance.

So we have a finite i-language we immediately know, but this is of course not sufficient. In addition to i-language, we have some devices, which allow us to derive new utterances from the ones we already know, either immediately or by derivation. These deductive mechanisms now account for linguistic creativity, and it is when we use these devices that we are being creative. What should these devices look like? What we need is a formalization of linguistic reasoning; and luckily, this is what the entire classical metatheory is about: So we can use exactly the pre-theories we have scrutinized in the last chapter. When we put this machinery to use again, we should be aware that these tools have changed their status: in the classical approach, they were just tools for the metalinguist. Now, they become a model of the creative *speaker*, and thereby belong to the subject of (intensional) linguistics. So the tools to define "language" in the classical metatheory become models of "language" itself in the intensional approach. Though this is quite a big change, we have been prepared for this: the main argument in favor of a certain pre-theory in the classical paradigm – apart from intrinsic mathematical properties – was that it formalizes linguistic reasoning as a working linguist would do it; and working linguist in turn would reason in a way he in turn thinks a speaker of a language would do.

This is all we need to construct intensional "language". But there are some more ontological differences. The first one is: in the classical approach, we had to take an observed language and project it to the infinite. In the intensional approach, of course we need an observed language to depart from; but there is an additional step in between: we have to decide which part of this observed language is actually i-language. So we have some freedom of decision here; but note that the choice of i-language, though arbitrary from a metalinguistic point of view, is an interesting empirical question from a linguistic point of view, because it makes a strong statement about the mind of the speaker. So this choice or freedom is not part of metalinguistics anymore, but rather of (intensional) linguistics proper. This is the reason we do not want it to become deterministic after all: we want to formalize the *metalinguistic* procedure, not linguistics itself. i-language is a cognitive notion, and to decide on it, linguists should consider all data which is relevant and available (such as reading times etc.).

The next important change is: rather than simply constructing the closure under the inferences, we keep track by which means inferences a certain set of strings has been derived. So rather than deriving a single infinite language, we aim at deriving a set of languages, which is structured by inclusion and the inference steps we used to get from one language to another.

Giving up the set conception solves many problems at once: gradience and acceptability becomes a matter of which analogy people can draw ad hoc; the difference of "language" and o-language is the difference between inferences linguists draw and inferences which speakers draw when they speak. This in turn is mostly the difference between "paper and pencil" inferences and inferences which can be made on the spot.

So given that we have so deep changes in our view on language, are there similarly deep changes in the underlying ontology of metalinguistics? Maybe surprisingly, there are only minor changes we have to make; the big changes

concern the part of "language" which we have to construct, not the one we are given.

The "canonical datum" of linguistics is still the judgment that $\vec{w} \in L$ or, in a considerably weaker form, $\vec{w} \notin L$. We have called this a linguistic judgment. As we said, for us a linguistic judgment need not be based on some immediate, implicit knowledge, but can also be derived. So we might want to distinguish between *immediate judgments* and *derived judgments*, which are derived from other linguistic judgments by means of analogy. The latter are thus no longer implicit and immediate, but require reasoning about language. This is however not visible to us; so it is our decision to classify them.[1]

So the positive language is given, and we do not have to make changes to that. The important difference is: we do not need to make the positive language the base of our analogies: we can also take a subset. We then only have to make sure that the rest of the positive language is derivable from the fragment we have used. Regarding the negative language, there are really no changes: for us, it was simply an instrument of control, and this it will stay. The content of the negative language was highly intensional in the first place, so there is nothing we need to change.

But for the object we construct, our ontology is much richer, we can distinguish the following languages:

1. i-language, which is immediately known
2. the language of sentences which can be derived ad-hoc ($\approx$ o-language) or "online" (in principle, there might be several of this)
3. the language of sentences which can be derived with arbitrary resources (in principle, there might be several of this)
4. a negative language, which should not intersect with any of the former.

This corresponds to a distinction of linguistic subjects. Regarding the first item, it does not seem that there has been research on i-language in our sense; but maybe it is worth the while: maybe the fact that this topic has not received attention is that there was no theory which pointed out its existence. The second point has gathered considerable attention, as it concerns the speaker as he is intuitively speaking ("natural data"). It can be thought of as the linguistic universe the *speaker* lives in, that is, the utterances he can make and understand. The third point corresponds to the linguistic universe of the linguist rather than the one of the speaker: it is the analogies we can make with paper and pencil. This roughly corresponds to the proper subject of linguistics in the classical conception.

So formal questions which arise most naturally are the following: 1. what are the features of the first, second and third? The first is an entirely new construct, so what does it look like? What are its formal/empirical characteristics? The

[1]Note that there are some terminology issues to be considered here, on the roles of explicit and implicit knowledge. In the terms of Hintikka ([26]), knowledge is implicit if it can be derived from what I know explicitly. In this sense, i-language is explicit, and what is beyond is implicit. On the other side, i-language can be said to be implicit, as I do not need to make any reasoning in order to arrive there, whereas what goes beyond can be said to be explicit on the grounds of (conscious) reasoning. We will therefore use the terms immediate - derivative in the latter sense.

next question is: 2. how can we characterize the second within the third? Questions regarding the third class roughly coincide with questions asked in the classical paradigm. The important difference is: we do no longer fix the "language" of the speaker, but rather "possible languages"; and which ones the speaker uses is an empirical matter.

On the downside, there is of course the danger that we solve all these problems and avoid all problems of the classical approach at the price of getting eve bigger ones. The main downside of our new approach is: the resulting intensional "languages" are very different from anything we are used to see as "language"; in particular they are much more complex. It is of course interesting to do linguistics with these structures, but surely not in the same fashion as in the classical conception, where all formalisms are, in one way or other, are characterizations of (infinite) sets. We will devote a small subsection to the question what intensional linguistics actually looks like. As we will see, this is not too obvious; but still, intensional linguistics seems to be interesting to pursue, and maybe might even put into practice what some linguists already implicitly consider to be a "better linguistics", without being too explicit about it.

5.3 The Thinking Speaker: Independent Evidence

5.3.1 Preliminaries

Going down this road, we challenge another fundamental assumption of linguistics. We already said that linguistics is usually supposed to be about the implicit, immediate knowledge of language. We have already said that this is very problematic and cannot be true from an epistemic point of view. So far, we have challenged it as a methodologic claim for theoretical linguistics, that cannot be sustained for the constellation of finite and infinite underlying linguistic theory. Now, we also challenge it as a programmatic guideline for linguistics: linguistics in the narrow sense is simply not interested in what happens when speakers *think* about their language. In this view, the thoughts and reasoning of speakers is essentially noise to the mythical, original competence, and has to be filtered out either by experimental methods (short presentations of stimuli), or by considering "natural data" as spontaneous speech, which is supposed to be immaculate by the reflections of the speakers.

The intensional view is quite different: the statement that knowledge of language is an implicit, unconscious knowledge for us is true *only* for i-language, which is a finite language without an interesting (syntactic, semantic) structure, because we assume it is characterized just by immediate acquaintance. This is *not* true for o-language, and much less for "language", the proper subject of linguistics. In intensional linguistics, the central notion is now the *extension* (rather than projection) of i-language, which is again effected via certain *inferences*. We think that we can use this term in more or less the same sense as we did in the classical metatheory, but on another level: it would be unjustified to assume that linguistic inferences are of any other kind as inferences in everyday reasoning, they are automatic and effortless up to a certain limited threshold, and beyond they become fallacious or even impossible. But now, they form part of knowledge of language.

What this is to say, however, is the following: "language" cannot be separated from the *reflecting* speaker, who thinks about his own language; and knowledge of language cannot be separated from reasoning about language. We have gone some way to find this conclusion acceptable (or maybe even necessary). Looking back at the traditional concepts of linguistics, we find it at odds with almost all standard approaches, be they cognitive in the Chomskyan or anti-Chomskyan sense, or not interested in cognition at all.

Many people will without hesitation adopt the classical metatheory; probably the same will hold for the finitist metatheory. In fact, we can say that our treatment of these metatheories consists in making explicit and mathematically concise what many scholars do anyway. For the intensional metatheory I cannot make this claim; it is surely the most non-standard, and it deviates a lot from what (to my knowledge) all linguists usually think and do. In particular, the claim that the speaker who is explicitly reasoning should be subject of linguistics proper will be hard to accept for most scholars. For this reason, I think it is the only one of the meta-theories presented here which needs an explicit, independent justification and motivation. Therefore, before we go into the formal foundations, we give a short overview of theories and observations, which provide an independent motivation for a notion as the reasoning speaker.

The most important point we want to make here is that in most "peripheral disciplines" of linguistics, the *thinking speaker* plays an important role. This mostly concerns sociolinguistics, historical linguistics, but also some non-standard views on theoretical linguistics. So all we want to do is to bring him from the periphery of linguistics to the core. We do not want to make a conclusive argument why the *thinking* speaker should be the proper subject of linguistics proper; neither do we try to give a complete picture of the role he plays in the disciplines we have mentioned here. Both enterprises would result in a book on their own. What I undertake is rather: I present some topics and literature where the thinking speaker plays a crucial role. I know the only thing I can achieve thereby is to show that it is a conception which should not be dismissed easily; I content myself with that.

Granted that the speaker reasoning about his language is quite well-established in many areas of linguistics and it seems quite hard to "explain him away" (even though probably it is not impossible), the intensional metatheory only to brings him from the periphery of linguistics to the very core.

5.3.2 Language Change

In historical linguistics, we can hardly overestimate the reasoning processes of speakers. Firstly, there are rather peculiar phenomena as popular etymology, which cause processes `capitolium` $\rightarrow$ `campidoglio` ('fields of oil'), which reinterpret a word which has become meaningless in a language into a meaningful one, even though there is no semantic motivation for it. It is clear that this kind of change presupposes a process of reasoning. Another example is the following: it has been observed that vowel systems in Australian languages are somewhat more narrow than elsewhere in the world, that is, they occupy only a subspace of the space of possible vowels. This contradicts an old and often repeated hypothesis which was first made by Martinet ([49]), which says that if a language has n vowels, then it will most probably have the n most distinct vowels, in terms of articular and acoustic vowel space. That means that these vowels will

be most probably found at the outer ends of the vowel space (except for the case where a language has only one vowel, which is not attested to my knowledge). For example, a language with three vowels will most probably have the vowels [a],[i],[u]; a language with five vowels will most probably have [a],[i],[u],[e],[o] etc.

This is a very natural hypothesis, which also has often been empirically confirmed (as a statistical universal, though). So it is puzzling that there is a geographical group of languages, the Australian aboriginal languages, which systematically deviates from this pattern. There is however a good explanation for this: in the area of Australian languages, there is a very common ear infection, which used to strike about half of the child population. This infection results in a loss of hearing mostly at the outer spectra of the human range, whereas the medium range rests quite intact. The Australian vowel systems are thus a direct adaptation to the needs of a large part of population with impaired hearing. The interesting thing is that this change cannot be triggered by language learners, as the disease strikes children which already master their language. Therefore, it has to be a (more or less) conscious change of the language by the speakers which has resulted in these vowel systems (see [68]).

Note that there are even more clear cases of language change: there seem to be changes which have been performed by conscious decision of a respected speaker or group of speaker. While the importance of these examples should not be underestimated, it should neither be overestimated: in the end of the day, these examples count as peculiarities. We will now consider one of the core processes of language change, namely analogy, which is (arguably) the most frequent and fundamental process to drive language change. We will see that analogy presupposes reasoning speakers, at least in the most plausible conception.

Let us take the case of sound change. Sound change often happens across the board, that is, a certain change equally affects all phones/phonemes in some phonetically/phonologically defined environment. Of course, there are exceptions to this: a sound change might affect only a particular item, say a very frequent one. Thinking about this, in fact, in fact it seems much more natural that sound change affects particular items rather than going across the board: surely sound change is triggered by speaking, and so it should not affect forms which are not affected by speech, such as rare forms, forms which belong to written language. So in the end, it is across the board sound change which need an explanation; and the most (only) natural explanation seems to be analogy. This of course presupposes that sound change is in fact triggered by capable speakers rather than language learners: otherwise, we would have to challenge the assumption that the same sounds are heard in the same way by humans (see also Lehmann, [44] p.209 for a similar criticism of the position that sound change only proceeds via language acquisition). So sound change must be triggered by what speakers do in their lifetime as capable speakers, and the fact that changes happen across the board must be due to their (more or less) conscious decisions on how to speak. This however seems to be impossible without analogy of phonological contexts. The same applies to morphologic change: we often see inflection paradigms change. Now if we make the (arguable) assumption that these paradigms do not exist as such in the mind of the speakers, these changes have to go by analogy. So in order to get morphological changes across the board, we need analogy. If we admit that these changes are performed by capable speakers (even though not necessarily adults), then we have to admit that speakers think about their

language, because *drawing analogies* (and inferences) is pretty much the essence of what we have treated as linguistic reasoning so far.

5.3.3 Sociolinguistic Typology: Trudgill

The "reasoning speaker" is also very well acknowledged in sociolinguistics. It is known since a long time that the spread of linguistic change crucially depends on social factors (see the famous work of Labov, [43]). There is a more recent and less widely accepted claim that not only the spread of linguistic innovations, but also the *type* of linguistic change depends on social factors (see [68]). For example, there is strong evidence for the claim that languages which are rarely learned as a second language and which moreover are spoken in stable societies tend to become rather complex, contrary to language which are frequently learned by L2 speakers and/or develop in socially instable environments.

In this perspective it is much more plausible that language change is triggered by adults as well as by children: because how would children know about the need to make themselves comprehensible to a wide community of different native speakers, or the lack of this need? Moreover for language change to happen the way it does, it is necessary that adults reason about their language, reason about comprehensibility etc., and make conscious decisions based on their social experiences. Of course, this sharply contrasts with the classical, generative stair model of language change being triggered only by L1-learners; but it also contrasts with the conceptions that language and linguistic knowledge is untouchable for the reasoning of speakers. For an extensive treatment, we refer to [68].

5.3.4 Roy Harrison: The Language Makers

The main point we want to make here has already been made very explicitly by Roy Harris (see [23]). He claims that languages are social constructions, made up by the attitudes and ideology of speakers towards it. So whereas all communities have some language, this object is strongly underdetermined from various points of view. The object speakers think to be their language is determined by certain social/cultural/intellectual background conceptions. This is a direct approach to the same question we are after, though it has a very different background motivation, and also very different goals. The observation is however the same: language as the collection of linguistic phenomena is a very incomplete object. In order to make it the object of scientific study, we have to "complete" it by adding some feature. The topic of this work is a case in point: in order to be a satisfying model, "language" has to be infinite, whereas the linguistic objects we observe lack this quality. So even before we can look for a satisfying theory, we first need a satisfying model of reality.

We can pick up one particular aspect of Harris' work. As Harris argues, our (modern) view of language is strongly determined by writing. Whereas he focusses on phonology, this is an important point also for syntax: can anyone be convinced that our standard projection and competence/performance distinction would be the same, if we did not consider written language at all? Not only as data, but just imagine also doing linguistics would be completely oral! Then it is easy to imagine that our usual conceptions rely to a huge extent on the fact that we write down sentences and ponder about them. Now assume we did

not have this possibility (or would not make use of it). I think many things which we claim to be "only performance restrictions" would actually be coded into competence, because we would not even see how to exceed them. This is actually a good point, as modern linguistics (at least linguistics proper) since de Saussure always underlined the priority of spoken language – but our usual linguistic conception of language is crucially based on writing. I do not think this in itself is a bad thing; but I think it is easy to agree that writing changed our conception of what is "language" – for the linguist as much as for the speaker; and this is exactly because language is shaped by speakers *thinking* about it.

5.3.5 Coseriu on Knowledge of Language

Another place where we find many traces of the ideas we lay out here is the work of Coseriu. Actually, this is not very surprising: as Coseriu does not have a strong cognitive commitment, he is quite open minded on the structure of language. In fact, I think it has been the Chomskyan strong focus on "language" as being something real in the mind/brain which has suffocated a lot of interesting discussions. We will therefore quickly review the work of Coseriu, mostly based on [11]. According to Coseriu, knowledge of language is quite fine-grained. This is firstly because he introduces the notion of the various *norms* of a language. He establishes a well-known three-valued distinction, as opposed to the classical dualities: there is firstly the *parole - habla - rede*, which corresponds roughly to the notion in Chomsky and Saussure. There is the *system*, which roughly corresponds to *langue* in Saussure. As a third and mediating object, he introduces the *norm*, which specifies how one should speak, that is, it constrains the use of the system. In a natural language with a normal history, he goes on, there are always many norms, according to the (diaphasic, diastratic, diamesic) variations of the language, and usually speakers are fluent in more than one norm. So the system is more liberal than the norm, it has less constraints. Extensionally speaking, the system is larger than the norm, but for Coseriu, the system does not have an extension and it cannot be instantiated without a norm.

Why is this interesting? The norm is a social thing and not part of the language system; nonetheless it guides the way in which speakers speak. But as by assumption it is not part of proper linguistic knowledge, it must be extra-linguistic knowledge of the speaker *which he puts to use while speaking*. Actually, this is a clear thing: nobody ever claimed speakers do not think while they speak. But the thing is that norm really concerns the structure of language, it is a necessary instance for the instantiation of the system. Elaborating on this thought – to be honest – it is difficult for me staying in line with Coseriu, maybe because he thinks in structuralist terms, so often it remains unclear to me when he talks about the speaker's mind or just about some abstract system of language.

However, there is another very interesting notion Coseriu introduces. In ([11],p.272,277), he introduces the notion of *Sprachtypus* (language type) as an additional concept, which is still more abstract and general than the system, which describes

> "die Gesamtheit der funktionellen Zusammenhänge zwischen Funktionen und Verfahren, die auf der Ebene des Systems als verschieden auftreten."

For Coseriu, type is rather functional than directly connected to language

structure; nonetheless, it describes something which is above the system, so in a sense, it is creative and transcends the language itself, while still being part of knowledge of language in the broadest sense.

We can change the norm, whereas the system remains the same; the converse is not supposed to happen, as changes propagate from the speech to norm to system. In the same way, the system might change, whereas the type remains unaltered. Coseriu is quite vague on this notion of language type. So with some interpretation on my behalf, we can describe the type as follows: it determines, in which way a language expands, changes and creates new possibilities. So *Sprachtypus* comes into play, where the system does no longer specify anything; and it remains constant even when the system changes. But note that the language type is not coded explicitly at any point, but is rather *implicit* in the knowledge of the linguistic system. In my interpretation, it arises from reasoning about the system. Language type could be said the *line of abstraction* of language; it determines the way in which we construe and construct new structures in the grammar, rather than in the utterances it describes.

In whatever way we want to make this concept precise, it is clear that there is something to it transcending the normal notion of a grammar. It is somehow encoded in the language, but surely it is not part of linguistic knowledge. Importantly, this is the same way that the intensional linguist thinks that certain structures (introduced by inferences) are encoded in the rules and i-language, though not explicitly represented. Phenomena we can typically connect with this *type* of a language might be ellipsis, which is always somewhat metagrammatical, in particular, the *direction* of ellipsis (left in German and Japanese, right in English, romance languages). Another topic is the one of *drift* (see [69]), that is, the fact that languages tend to change their word order consistently into one direction, even in unrelated structures. So the type specifies things which transcend the system.

We should also mention that these conceptions can already be found in Humboldts work, who is also quoted by Coseriu. Unfortunately, these interesting ideas have not yet found their way into formal or even canonical linguistics. Maybe this might change, in case the intensional metatheory gains some popularity.

5.4 The Mathematics of Intensional Linguistics

5.4.1 Languages as Structures

We now come to the formal treatment of intensional languages. We call a function f on a set of sets *extensive* if $I \subseteq f(I)$ for any I of the domain. Let J be an arbitrary index set. A constructive language is a structure $(I, \{\eta_i : i \in J\}, \{I_{\vec{j}} : \vec{j} \in J^*\})$, where I is a finite language, the η_i are extensive functions from (finite) languages to (finite) languages, and the set $\{I_{\vec{j}} : \vec{j} \in J^*\}$ is a set of languages, which is defined as follows: (1) $I = I_\epsilon \in \{I_{\vec{j}} : \vec{j} \in J^*\}$; and (2) if $I_{\vec{l}} \in \{I_{\vec{j}} : \vec{j} \in J^*\}$, $\eta_i \in \{\eta_j : j \in J\}$, then $I_{i\vec{l}} = \eta_i(I_{\vec{l}})$, and thus $I_{i\vec{l}} \in \{I_{\vec{j}} : \vec{j} \in J^*\}$. The η is now mnemonic for "extension" (rather than projection).

This is to say that each language carries as an index the inferences by which it has been derived from I, which is the immediate language. We leave it open

whether the set J is finite or infinite; for practical reasons it will remain finite in the sequel, but in principle, it needs to be only finitely specified. Note that even if I, J are non-empty, $\{I_{\vec{j}} : \vec{j} \in J^*\}$ modulo (extensional) equality of languages need not be infinite. In order to provide us with infinitely many distinct $I_{\vec{j}}$, the functions η_i and/or the language I need to satisfy additional requirements.

This is what we have already stated in the last chapter. We can now elaborate on this. We can define η to convert a pre-theory into an extension function rather than a projection. Let $(\mathfrak{f}, P)$ be a pre-theory. Unfortunately, we cannot define a general notion of an **atomic** $(\mathfrak{f}, P)$-derivation, just because different pre-theories are so diverse (some can derive new linguistic judgments in one step, others need several intermediate steps); so we just define the general conditions which atomic derivations of a pre-theory must satisfy, where this definition still leaves considerable options in most cases.

Definition 143 *Let $(\mathfrak{f}, P)$ be a pre-theory. We say $\mathfrak{X}$ is a **set of atomic** $(\mathfrak{f}, P)$-**derivations**, iff*

1. *every derivation $T \in \mathfrak{X}$ is an $(\mathfrak{f}, P)$-derivation,*
2. *for any finite language I, only a finite set of $\vec{w} \in \mathfrak{f}_P(I)$ can be derived by some derivation in $\mathfrak{X}$,*
3. *every $(\mathfrak{f}, P)$-derivation tree can be decomposed into subtrees in $\mathfrak{X}$, such that every node of the $(\mathfrak{f}, P)$-derivation tree, which belong to two $\mathfrak{X}$-trees, is labelled by a linguistic judgment.*

We say an $(\mathfrak{f}, P)$ derivation is atomic wrt. $\mathfrak{X}$ if it is in the set $\mathfrak{X}$ and $\mathfrak{X}$ is a set of atomic $(\mathfrak{f}, P)$-derivations.

Of course not every pre-theory does even admit a set atomic derivations; but we assume that all reasonable ones do so, and if a pre-theory does not, then it is not suitable for our purposes here. If we speak of a pre-theory and its atomic derivations in the sequel, we assume they have been defined in some way in accordance with definition 143. Let $\mathfrak{X}$ be a set of atomic $(\mathfrak{f}, P)$-derivation. We now define $\eta_{(\mathfrak{f},P,\mathfrak{X})}(I)$ as the set of all strings $\vec{w}$, such that either $\vec{w} \in I$, or there is an atomic $(\mathfrak{f}, P)$ derivation of $\vdash \vec{w} \in \mathfrak{f}_P(I)$. Coming back to our index set J, we now assume there is a bijection ϕ from J to a set of pre-theories each associated with a set of atomic derivations. For simplicity, we write η_i for $\eta_{\phi(i)}$. Furthermore, we define, for $\vec{j} \in J^*$, $i \in J$, $\eta_{\vec{j}i}(I) = \eta_i(\eta_{\vec{j}}(I))$, such that we have the general equality $\eta_{\vec{j}(I)} = I_{\vec{j}}$, in an appropriately defined intensional language.

This definition has one arguable feature, namely the following: for each extension, everything we have derived so far has the same status as the original, immediate language. Under this definition, we therefore do *not* get, for $\phi(i) = (\mathfrak{f}, P, \mathfrak{X})$,

$$\mathfrak{f}_P(I) = \bigcup_{\vec{j} \in i^*} \eta_{\vec{j}}(I) \tag{5.1}$$

because we always recompute new analogies for each new set we derive. From this results an incomparability to the classical pre-theories. But for example if $(\mathfrak{f}, P)$ is strongly upward normal and weakly monotonic, under these assumptions, the equation can be shown to hold, as follows from lemma 64. In general, this definition might be acceptable, as we always stick with finite languages; but

on the other side, we might want to treat immediate knowledge and derived knowledge differently.

We therefore also have an alternative definition. Let I_1, I_2 be (finite) languages, $(\mathfrak{f}, P)$ be a pre-theory, $\mathfrak{X}$ a set of atomic $(\mathfrak{f}, P)$-derivations. By $\eta^{I_1}_{(\mathfrak{f},P)}(I_2)$ we denote the set of all strings $\vec{w}$, such that either $\vec{w} \in I_2$, or there is an atomic $(\mathfrak{f}, P)$-derivation of $\vdash \vec{w} \in \mathfrak{f}_{P(I_1)}(I_2)$, that is, from premises $\vec{v} \in I_2$ and analogies in $P(I_1)$. The alternative construction of intensional languages is by the usual map ϕ connecting indices with pre-theories and atomic derivations, and by η^I: we define the language as $(I, \eta^I_i : i \in J, I_{\vec{j}} : \vec{j} \in J^*)$. So in this language, the extending maps always refer to I for their analogies. For an intensional language constructed in this fashion, equation (5.1) always holds, and so in the case we only use pre-theories being both upward normal and weakly monotonous, the two ways of constructing intensional languages should not differ essentially.

But of course, what is really interesting about intensional languages is that in the intensional paradigm, we have a *number of pre-theories*, which all yield possibly different languages, and we put all of them to use at the same time. For strings, define the relation *pref* as follows: for $\vec{x}, \vec{y} \in \Sigma^*$, we have $\vec{x}$ *pref* $\vec{y}$ iff there is $\vec{z} \in \Sigma^*$ such that $\vec{x}\vec{z} = \vec{y}$. We obviously have the following: in some intensional language $(I, \{\eta_i : i \in J\}, \{I_{\vec{j}} : \vec{j} \in J^*\})$, if $\vec{j}$ *pref* $\vec{j}'$, then $I_{\vec{j}} \subseteq I_{\vec{j}'}$. Still, though sets are always growing, it is important to keep in mind the difference between $(I, \{\eta_i : i \in J\}, \{I_{\vec{j}} : \vec{j} \in J^*\})$ and $\bigcup_{\vec{j} \in J^*} I_{\vec{j}}$. The reason is that in the former we do have some structure, whereas the latter is simply a set. For example, we can define a regular language $R \subseteq J^*$, and define $\bigcup_{\vec{j} \in R} I_{\vec{j}}$; this might actually be a set which cannot be defined by any pre-theory we have used!

So the big advantage for us is that though "language" in the intensional paradigm looks really different from "language" in the classical sense, we can transfer the classical methods, i.e., we can put our pre-theories to use. This also means: many formal questions can be answered by means of the classical answers. For example, assume we have a given intensional language $(I, \{\eta_i : i \in J\}, \{I_{\vec{j}} : \vec{j} \in J^*\})$, J representing a set of pre-theories. What is the language $\bigcup_{j \in J^*} \eta_j(I)$? There is no easy answer to this, but one might think that for extension functions of the form $\eta^I_i : i \in J$ there might be one. Given two pre-theories $(\mathfrak{f}, P), (\mathfrak{f}', P')$, define their least upper bound by $\mathfrak{f}_P \vee \mathfrak{f}'_{P'}(I) = \mathfrak{f}_P(I) \cup \mathfrak{f}'_{P'}(I)$. By our completeness result, this implicitly defines a pre-theory, and of course, we can extend this to arbitrary finite sets. Assume J is finite, and each $\eta^I_i : i \in J$ has the form: $\eta^I_{(\mathfrak{f}^i, P^i, \mathfrak{X}^i)}$, where $(\mathfrak{f}^i, P^i)$ is a pre-theory, $\mathfrak{X}$ its atomic derivations. We can easily show the following equation to be valid under the above assumptions:

$$\bigcup_{\vec{j} \in J^*} I_{\vec{j}} = (\bigvee_{i \in J} \mathfrak{f}^i_{P_i})(I) \tag{5.2}$$

The proof is immediate from our definitions. Note however that it is wrong if our extension functions do not have the form η rather than η^I!

5.4.2 Language Definability

There is now type of question which arises: given a set J of (representatives of) pre-theories, a language $R \subseteq J^*$, what is the class of languages: $\bigcup_{\vec{j} \in R} I_{\vec{j}}$ for some finite I? Or more generally, given a class of languages C, a set a set J of representatives of pre-theories, what is the class of languages of the form

$\bigcup_{\vec{j}\in R} I_{\vec{j}}$ for some finite I and some $R \in C$? In this case we say a language/class of languages R/C **defines** a language within $(I, \{\eta_i : i \in J\}, \{I_{\vec{j}} : \vec{j} \in J^*\})$; we call this notion **language definability** (or shortly, l-definability).

A notion like l-definability seems to be far off from current linguistics. However, we would like to explain *why* such a notion might be interesting. In a word, it might be a way to find a bridge between the classical and the finitist approach. The finitist approach sticks to what is strictly visible; the classical approach tries to project any *locally visible* pattern into the infinite, regardless of whether speakers in the end are able to actually understand, utter or judge such an utterance. As we have pointed out in the last chapter, there are inferences which seem to *preserve* acceptability, there are those which do not, and there are inferences which preserve acceptability on a certain restricted, local scale. Now the main advantage of the intensional view is that we do not fix "language" for the speaker; an intensional language is not a model of effective knowledge, but rather a model of all possible inferences. So this paradigm allows us to investigate: which is the largest substructure of an intensional language, such that it still preserves acceptability? That is, by means of language-definability we can use analogies which only locally preserve acceptability, but we can require them to be used in a restricted fashion; whereas analogies, which globally preserve acceptability, might be used in an unrestricted fashion.

This shows us that the additional structure of intensional languages is not just for the sake of itself: we can (try to) define the *acceptable* strings within the language in a way we cannot do in the classical paradigm. More generally speaking, we have freed ourselves of a fundamental worry. In the classical approach, once we have fixed our meta-theory, there were only two possible answers to the question whether $\vec{w}$ belongs to "language": either yes or no. Now we can give in principle infinitely many answers: it belongs (or does not belong to) the language defined by R etc. The linguistic challenge is to boil these infinitely many answers down to a reasonable number of answers, say 3: 1. yes, in the sense of the most general extensions; 2. yes, in the sense of acceptable extensions, 3. no. Or to get a still more fine-grained distinction, we can distinguish between 1. immediate knowledge, 2. "online" acceptability, 3. "offline" acceptability (or grammaticality), and 4. ungrammaticality.

The notion of l-definability seems quite appealing, but we can easily construct an example why it is unsatisfying from a linguistic point of view. Consider the pre-theories $(\mathfrak{g}, RPr)$, $(\mathfrak{g}, Pr)$; we put $J = \{1, 2\}$, and $\phi(1) = (\mathfrak{g}, RPr, \mathfrak{X}_1)$, $\phi(2) = (\mathfrak{g}, Pr, \mathfrak{X}_2)$. Recall that RPr is the regular restriction of Pr. We define their sets of atomic derivation $\mathfrak{X}_1, \mathfrak{X}_2$ in the most obvious way, as the smallest non-trivial derivations of linguistic judgments. It is easily checked that this defines atomics derivations for $(\mathfrak{g}, Pr), (\mathfrak{g}, RPr)$. Next we consider a language

$$I := \{wxv, wxvw'xv', wy_1xy_2v, wy_1xy_2v, wy_1xy_2vw'xv', wy_1xy_2vw'y_1xy_2v', wxvw'y_1xy_2v'\}. \tag{5.3}$$

We then get $RPr(I) = \{(wxv, wxvw'xv'), (wy_1xy_2v, wy_1xy_2vw'xv'), ...\}$; and $Pr(I) = RPr(I) \cup \{(x, y_1xy_2)\}$. Now we get the structure $(I, \{\eta_i : i \in \{1, 2\}\}, \{I_{\vec{j}} : \vec{j} \in \{1, 2\}^*\})$. We have

$$\bigcup_{\vec{j}\in\{1\}^*} I_{\vec{j}} = \{wxv, wy_1xy_2v\} \cdot (\{w'xv', w'y_1xy_2v'\})^*; \tag{5.4}$$

and we have

$$\bigcup_{\vec{j}\in\{1,2\}^*} I_{\vec{j}} = \bigcup_{\vec{j}\in\{2\}^*} I_{\vec{j}} = \{w(y_1)^n x((y_2)^n y : n \in \mathbb{N}_0\}\cdot(\{w'(y_1)^m x(y_2)^m v' : m \in \mathbb{N}_0\})^*. \tag{5.5}$$

It is actually not too difficult to construct languages of the form

$$\bigcup_{\vec{j}\in(1^*+(1*21^*))} I_{\vec{j}} \tag{5.6}$$

etc. As we said above, we might think that RPr-inferences preserve acceptability, whereas Pr inferences in general do so only on a very local level. For example, we can conjecture that strings in $\{w(y_1)^n x(y_2)^n v : n \in \mathbb{N}_0\} \cdot \{w'(y_1)^m x(y_2)^m v' : m \in \mathbb{N}_0\}^*$ are acceptable as long as $n, m \leq 3$. This small example allows us some criticism of our method so far: assume we use l-definability for an $R \subseteq \{1,2\}^*$. What we want to restrict is the number of 2s in words in R; for $\vec{w}$ a string, by $|\vec{w}|_a$ we denote the number of as in $\vec{w}$. So for example, $R := \{w \in \{1,2\}^* : |w|_2 \leq 6\}$. We use 6, because this 6=3+3, the maximum values for m, n. But here is the problem: the number 6 is too liberal, because it can be that $n = 6$, $m = 0$. But at the same time, it is to restrictive: we can have in many iterations of $w'y_1y_1xy_2y_2v'$, each time using a single inference in 2, and still preserve acceptability, because the inferences are not "nested" in the intuitive sense.

So l-definability does not seem to be a good way to go: it is too restrictive and too liberal at the same time. This is because languages do not give us any information on where exactly we can use the inferences; they only tell us about cardinality and order, but this underspecifies many things. In particular, one would think that we should in some way refer to the structures we induce – presupposing a sort of structured inference as in $\mathfrak{g}$. We could further investigate on definability in intensional languages. The reason we do not do so is the following: I think that this already leads us into the realm of **intensional linguistics** rather than metalinguistics; because the question whether an intensional sublanguage preserves acceptability is clearly a linguistic and empirical question. In virtue of this fact, this is no longer covered by the topic of this thesis, which is "strictly metalinguistics". Nonetheless I think these considerations have their place here, because they give us an idea of what the intensional *linguist* is doing.

5.4.3 Adequacy

As we have stated above, in the intensional approach we can, as in all linguistic metatheories, assume that there is a partial language (I_1, I_0), which we are given as primary data. What has become slightly more complicated is the procedure for adequacy. We have to test for three things: a partial language $(I, \eta_i : i \in J, I_{\vec{j}} : \vec{j} \in J^*)$ is adequate wrt. a partial language (I_1, I_0), if

1. $I \subseteq I_1$;
2. $I_1 \subseteq \bigcup_{\vec{j}\in J^*} I_{\vec{j}}$;
3. $I_0 \cap \bigcup_{\vec{j}\in J^*} I_{\vec{j}} = \emptyset$;
4. $\bigcup_{\vec{j}\in J^*} I_{\vec{j}}$ is infinite.

The last three conditions are clear: no extension must contain any of the negative data, and all positive data must be contained in some extension; there must be infinitely many strings in some extension. The first one needs explanation: I here corresponds to i-language, not the observed language, and as we have said, i-language and observed language need not be in either direction included in one another. This, however, holds only in principle. Once we put our metatheory to work, we cannot assume that i-language contains any string we have not observed: if we have not even made an observation, how can we assume we immediately know it? If this would be the case, we should be able to verify that it belongs to our positive data, and so we have to assume that we can add it to I_1: what is legitimate as a general possibility, is not a legitimate assumption for the construction of a concrete "language", and therefore we require $I \subseteq I_1$.

5.5 Some Notes on Intensional Linguistics

Before I close this section, I add some notes on what I think intensional linguistics looks like. This is because I have the impression that in this paradigm, the line between linguistics and metalinguistics is easily blurred. So what does intensional *linguistics* look like? Of course, its methods are very different from classical methods, as the latter mostly consist in characterizations of infinite sets. So what would an intensional metatheory mean for the working linguist?

Firstly, it is of course the meta-linguists task to provide a sufficiently rich set of pre-theories. The first task of the linguist is then to make observations. Then, assuming he has a dataset D of observations, the first important linguistic question is: which part of D is i-language? This is of course a cognitive and therefore an empirical question, so it is proper linguistics, and the linguist should use all his tools and gather all information he can to decide on this question. Such information can be reaction times (for grammaticality judgments), reading times, and might even involve the brain. This is a first major task.

Now assume the intensional linguist has fixed an i-language I. Then by the pre-theories, we already have an intensional language. We then have the usual question for adequacy, which is rather part of the metalinguistic work. Now assume we have an adequate intensional language. By our philosophical assumptions, this is by no means the "language" of the speaker; it rather is a structure of all linguistic inferences he can possibly make. We now come to the second main task of the linguist: he should try to find the fragment of the intensional language which the speakers actually use (in various senses). So here we come to the question of definability, which we have outlined above. Note that there are two main questions: the first one is empirical and asks in how far certain extensions preserve acceptability. The second one is: which tools can we develop in order to define certain intensional sublanguages? Our approach using l-definability turned out to be unsatisfying even for the most simple examples, so I guess there will be plenty of work to do developing appropriate tools.

Once we have defined a certain sublanguage of the intensional language, what else is to do? At the first glance, quite little, because the pre-theories already determine the structure of language, so most of the "classical" linguistic work is redundant. But there is one thing we should keep in mind: as pre-theories now are models of reasoning speakers, there might be different criteria speaking in favor or disfavor of one or another, even though they might be extensionally equivalent

(or indistinguishable): again, there might be additional data suggesting that speakers draw inferences in a certain way rather than in another, equivalent one. For an intensional linguist, the structure of an intensional language should – as much as this can be possible – contain some counterpart of the cognitive processes of speakers. And having said this, most linguists might agree they would not quickly run out of work in this paradigm.

So intensional linguistics is not as simple and straightforward as it might seem at the first glance. One should also bear in mind that it is not only the classical metatheory which does the work for the intensional linguist, but also vice versa: his considerations on "cognitively adequate inferences" should be quite relevant for the classical metalinguist, because he also is looking for "plausible" notions of analogy and inference.

Note also that we can always move back from the intensional to the classical paradigm: we simply take the union of all derivable strings or the union of the strings of a definable sublanguage; with the resulting sets, we can just do classical linguistics. I guess this is no longer in line with the strict "intensionalist philosophy", but it shows how the notion of intensional languages and definability might bring some new concepts and ideas even into the paradigm of classical linguistics.

Chapter 6

The Finitary Metatheory of Language

Summary of Finitary Conception

The finitary metatheory renounces to any explicit metalinguistic procedure: we gather positive data and construct negative data, and then directly devise our theories of language (in the sense of classes of possible grammars/languages). Our theories have to characterize infinite languages, whereas our data still consists of finite languages. Thus we necessarily have a mismatch between the two, and falsification becomes a non-trivial thing. Therefore, to make our theories meaningful, we have to make sure that they can be falsified in some way by some finite (partial) language we observe. Moreover, to ensure the possibility of falsification, linguists have to continue considering new data to try to falsify existing theories.

Note that strictly speaking, this is all (finitary) linguistics, rather than metalinguistics. If there is any explicit function for a finitary metalinguist, it is the one of a referee, who (i) controls that the theories which linguists use are in fact falsifiable by finite data in some way, (ii) controls that linguists in fact consider all existing data, and (iii) controls that they continue gathering new data.

So gathering new data, being problematic in the classical metatheory, acceptable in the intensional metatheory, becomes essential in the finitary metatheory.

6.1 The Finitist Position

We already have laid out the finitist philosophy above. We just quickly repeat the most important points to keep in mind: the finitist believes that "language" is o-language, in words: the proper subject of linguistics consists only in the observable utterances. From this it follows that "language" is regular by a simple argument: as the memory of all speakers is bounded by some constant, even hearing them speaking for an infinite amount of time, we would still not be able to observe a non-regular language.

As another consequence, the finitist renounces to project observed languages at all; he wants to stay only with observed data. There are two main reasons for this: firstly, he has little reason to do so: why should he project if the data he desires will come to him? But there is also a good reason not to project: using a projection, how can we know that all utterances of the projected language are really observable? Of course, there are inferences which seem to preserve acceptability, but can we really be sure? That inferences preserve acceptability already seems to be a strong assumption on the infinitary nature of "language". So we skip basically the entire metatheory and stick with what we have.

We stress however that this does not mean that the finitist does not believe that "language" is infinite: o-language is infinite, as there are no bounds to our observations; what is finite are *observed languages* (recall our discussion in chapter 2.8). So in finitist *linguistics*, we still write grammars for infinite languages; it is only the *metalinguist* which does not perform any form of projection. So whereas the intensionalist renounces to commit himself to a particular pre-theory, the finitist renounces to pre-theories entirely. But now of course there remains a gap between the finite and the infinite. In order to bridge this gap, he takes the approach of **falsificationism**: he devises theories, which he then falsifies by the data. This is not so much different from "normal",

classical linguistics; what is important however is that he must be able falsify his theories by means of finite languages, or at least by *partial languages*. As a consequence, he requires that his theories (classes of languages) C have a property as the finite language property (FLP, there exists a finite language $I \notin C$), or the partial language property (PLP), which we will define below.

As we said, finitism is in a sense the "simplest" solution to the problem of linguistic metatheory. This is surely the case from a philosophical point of view. From a mathematical point of view, the finitist *metatheory* does not need any formal tools. The only challenge consists in devising *linguistic theories* which are well-suited for finitism. There has been some work in this vein, but nearly nothing compared to the extensive work on the "classical paradigm" of linguistics. For this reason, we will here present the outlines of some formal methods to approach finitist linguistics. But note that strictly speaking, these already form part of finitist linguistics, not of metalinguistics.

There is also a downside to this property of being simple. As far as we can see from the approaches to finitist linguistics that exist in the literature, the generalizations and techniques are unfortunately not nearly as rich and thrilling as in "classical linguistics". We will therefore present some new techniques and ideas to make finitist linguistics more substantial, showing that there are interesting concepts and notions, which also have been put to work up to a certain extent. However, I do not think that changes the overall picture: still the techniques of finitary linguistics are much less developed than those of classical linguistics, and I guess few formal linguists would find it thrilling to work within this paradigm. That does however 1. not make them more or less true, and 2. is only a first impression, which might also be falsified by further research.

6.2 FLP, PLP and Subregular Languages

Let REG be the class of regular languages, FIN the class of finite languages. Linguistic theories in our general sense consist of classes of languages, or formalisms characterizing these classes. As we are committed to the priority of epistemic concerns, we just focus on the languages, no matter how they are characterized, because these are the empirically most accessible objects. As we have sketched above, we have two requirements for a class of languages which should qualify for finitary linguistics. Let C be a class of languages; in order to be adequate for finitary linguistics, we must have 1. $C \subseteq REG$, and 2. we must be able to falsify C by the data we have. There are two main ways to achieve this; the most simple one is: we require that $FIN \not\subseteq C$ (this is what we call FLP). What do classes look like which qualify in this sense? There are well-known classes of languages which satisfy these requirements; consider, for example, the local languages, co-finite languages, languages piecewise testable etc. There is quite some work on these subregular language classes; and as these are well-established, there is little need to expose them at this point (we refer the reader to [59]). There is however a problem with the "canonical" subregular language classes: they are not quite apt for our purposes. We would, for example, not think that local languages form a good model for the regularities of natural languages. We will therefore take a slightly different approach, which we hope will be considered more adequate for linguistic purposes.

One can correctly object that the FLP is not very useful in the end: also the

finitist acknowledges that "language" is infinite; he is only agnostic about how it looks like. Consequently, the finitist linguist also writes grammars for infinite languages; that he cannot write grammars for certain finite languages does not bother him: he would not write them anyway. What is rather important: his theory C prevents him to write a grammar for an infinite language which contains a dataset D he has observed. We have already mentioned this property: not every finite language has an infinite extension in C, that is: there exists a finite language I, such that there is no infinite $L \in C$ with $I \subseteq L$. This is however an extremely restrictive property; in particular, it would disallow us to have $\Sigma^* \in C$ for any alphabet Σ. None of the classes we presented above has this property. We do not really know how to define this property in a reasonable way. However, there is a way out: recall also for the finitist, data consists in *partial languages* rather than simply finite languages; this makes falsification much easier. Recall that L is a completion of (I_1, I_0) iff 1. L is infinite, 2. $I_1 \subseteq L$ and 3. $L \cap I_0 = \emptyset$.

Definition 144 *A class of languages C has the **partial language property** (PLP), if there exists a partial language (I_1, I_0) over Σ^*, such that there is no completion $L \subseteq \Sigma^*$ of (I_1, I_0) such that in $L \in C$.*

Note that this property of classes of languages is closely related to the property described by Angluin's theorem (see [1]). The PLP is a reasonable property for classes of languages, but consider that using PLP instead of FLP requires a stronger ontology: for FLP, we only need finite sets of positive observations; for PLP, we also need to make use of the negative data, which has a critical status. So though PLP is surely preferable on conceptual grounds, this comes at a price.

Before we proceed, let us characterize which classes of languages do not have the PLP. Let $Co\text{-}FIN$ denote the class of cofinite languages, that is, the class of languages L such that for Σ the smallest alphabet such that $L \subseteq \Sigma^*$, $\Sigma^* - L \in FIN$.

Lemma 145 *Any class of languages C such that $Co\text{-}FIN \subseteq C$ does not have PLP.*

Proof. For any partial language (I_1, I_0), where $I_1, I_0 \subseteq \Sigma^*$, we can form the completion $\Sigma^* - I_0$. As I_1, I_0 are finite by assumption, this is cofinite. □

In particular, assume $FIN \subseteq C$, and C is closed under complement. Then C does not have PLP. So the PLP is quite a strong restriction. Note however that the inverse of the above lemma is wrong: there are classes C such that $Co\text{-}FIN \not\subseteq C$, but which still do not have PLP; just consider the class of co-infinite languages, that is, the class of languages whose complement wrt. the smallest alphabet is infinite.

6.3 Derivatives of Languages

Let rad be the radix order over Σ^*, which is defined by an (irreflexive) linear order $<$ on Σ, and where $\vec{w}$ rad $\vec{v}$, if either $|\vec{w}| < |\vec{v}|$, or $|\vec{w}| = |\vec{v}|$, $\vec{w} = \vec{x}a\vec{y}$, $\vec{v} = \vec{x}b\vec{z}$, and $a < b$. By $min_{rad}(M)$ we denote the rad-minimal element of M. As rad is linear, this is a unique object in M. Let $L \subseteq \Sigma^*$ be a language. Then we define $suf_L(\vec{w}) := \{\vec{v} : \vec{w}\vec{v} \in L\}$. Next, we put $der(L) := \{\vec{v} : \vec{v} = min_{rad}(suf_L(\vec{w}))$ for some $\vec{w} \in \Sigma^*\}$. We thus have the set of rad-minimal sufficient suffixes; by

sufficient suffixes we mean: sufficient to complete any word in $pref(L)$ to a word in L, where $pref(L) := \{\vec{w} : \vec{w}\vec{v} \in L\}$. The rad-minimality is quite arbitrary to make suffixes unique, but makes sure the suffix we choose is unique and among the shortest. The following is easy to obtain:

Lemma 146 *Let $L \in REG$. Then $der(L)$ is a finite language.*

Proof. There is a deterministic finite automaton $\mathfrak{A}$ such that $L = L(\mathfrak{A})$. This means, that after reading $\vec{w}$, we are in one state q of a finite set of states. For each state, there is a rad-minimal word $\vec{v}$ such that for some $q' \in F$, we have $\delta(q, \vec{v}) = q'$. □

The converse does, of course, not obtain: there are non-regular languages such that $der(L)$ is finite. In fact, there are languages which are not even recursively enumerable which satisfy this condition: as is well-known, there are uncountably many infinite words over an alphabet Σ, provided that $|\Sigma| \geq 2$. In the sequel, we mark infinite words with an overline, as in $\overline{w}$. For each of these words $\overline{w} \in \Sigma^\omega$, the set of its finite prefixes, denoted by $pref(\{\overline{w}\})$, is a language such that $der(pref(\{\overline{w}\})) = \emptyset$. Now, as there are only countably many recursively enumerable languages over a given alphabet Σ, it follows that most (uncountably many) languages of the above form are not recursively enumerable. We now define the der_k-languages:

Definition 147 *L is a der_k language, iff $max\{|\vec{w}| : \vec{w} \in der(L))\} \leq k$.*

So a der_k-language L has a fixed upper bound for strings in $der(L)$. This is an interesting class:

Lemma 148 *For each $k \in \mathbb{N}$,*

1. *der_k has the finite language property, and*
2. *der_k has the partial language property.*

Proof. 1. Assume $k' > k$; then $\{a^{k'}\} \notin der_k$. 2. Assume $k' > k$; put $(I_1, I_0) = (\{a^{k'}\}, \{a^i : i < k'\})$. □

Of course, the first counterexample is uninteresting for our purposes, because also the finitist linguist writes grammars for infinite languages. Of course, we can easily construct an infinite language out if this example: just consider this finite language and take the union with an infinite language which is over a completely different alphabet. This technique can be applied in most cases which are to follow, but the resulting examples do not seem to be considerably more interesting.

What is more relevant is the following. If a language is der_k, this means that if someone you speak to begins any sentence (we now switch to linguistic terminology) and stops at an arbitrary point, you can finish this sentence with at most k words. Here we see an immediate relation to natural languages: the notion of der_k-languages is immediately related what we normally call "unresolved dependency" (see for example [20]) if we talk about (speaker's) language processing. From a language-theoretic point of view, these "dependencies" are just things which still have to be said to make the sentence complete, and der_k simply puts an upper bound to them. Actually, similar considerations are known in the literature and have been put to test to some extent, see [36]. Though

the approach is somewhat different, Kornai comes to the result that for natural languages, $k = 4$ is sufficient. This is a perfect example of a finitist approach to natural language.

So this is a promising approach, and we will try to elaborate it further. The approach using derivatives seems to be unsatisfying in one regard: in general, we assume that natural languages have "structure". We will not ponder a lot what this means in a purely language-theoretic sense, but one important property which seems to be implied by being "structured" is the closure under inversion: there is no fundamental asymmetry of left and right, the (global) linear order can be inverted, without affecting membership in the class. So if there is a language $L \in C$, so should be its inversion. Define $(a_1...a_n)^{-1} = a_n...a_1$, the inversion of a word. By $L^{-1} := \{\vec{w}^{-1} : w \in L\}$. A class of languages C is closed under inversion, if from $L \in C$ it follows that $L^{-1} \in C$. We now close der_k under inversion, by defining an inversion of der.

We define $pref_L(\vec{w}) := \{\vec{v} : \vec{v}\vec{w} \in L\}$ Put $der^{-1}(L) := \{\vec{v} : \vec{v} = min_{rad}(pref_L(\vec{w})$ for some $\vec{w} \in \Sigma^*\}$. Now we come to a new definition:

Definition 149 *L is an $ider_k$ language, if 1. $max\{|\vec{w}| : \vec{w} \in der(L)\} \leq k$, and 2. $max\{|\vec{w}| : \vec{w} \in der^{-1}(L)\} \leq k$,*

Note that $der_k \subseteq ider_k$; for $L \in der_k$, we have $L \in ider_k$ only if also $L^{-1} \in der_k$. From this it follows immediately that $ider_k$ has FLP; also the following is immediate:

Lemma 150 *For all $k \in \mathbb{N}$, $ider_k$ is closed under inversion.*

Proof. We have $der^{-1}(\vec{w}) = der(\vec{w}^{-1})$; $der^{-1}(\vec{w}^{-1}) = der(\vec{w})$. □

Corollary 151 *For all $k \in \mathbb{N}$,*

1. *$ider_k$ has the finite language property, and*
2. *$ider_k$ has the partial language property.*

What is the intuition behind these properties? Thinking in structures/trees, as many linguists do, one could say: der_k imposes a restriction on the depth of center-embedding; there is always a "way out" in at most k words. $ider_k$ in addition requires: to any kind of structure, there is a shortest "way in" of at most k words. For example, I cannot start a sentence with a relative clause, but I can reach a relative clause after k words. Or take a language such as $\{a^n b^m : n > m\}$. This language is an der_1, because I can always immediately finish a prefix of a word in this language. But it is not in $ider_k$ for any k: for a suffix of the form b^k, I require a prefix of length at least $k + 1$! What is thus interesting about the notions underlying $ider_k$ is that they have their counterpart in our intuition over linguistic descriptions in form of trees and dependencies.

These properties are quite interesting. As we have already said, we exclude, among others, languages of the form $\{a^{k'}\}$, where a is a letter, for some fixed k', where $k' > k$. $ider_k$ comes with additional constraints: consider a language $pref(a^{k'}b)$. This is in der_k; however, it is not in $ider_k$. This seems to be a good thing; in particular we seem to exclude something like: there is finite word of arbitrary length, only the prefixes of which are in L. We might think that for $ider_k$, if we make some observations beyond the bound k, we can make strong

conclusions on the infinitary nature of the pattern. However, this is not always true. To see this, just take factors of an arbitrary word $\vec{w}$ (or infinite word $\overline{w}$); we always have $fact(\vec{w}) \in ider_k$. This is what we will address next.

6.4 Infinitary Prefixes

We do not consider it plausible that languages just consist of prefixes and suffixes of a finite or infinite word. We rather think that if a patterns is visible up to a certain length, then it is also "infinitary", that is, it is visible up to an arbitrary length. We approach this problem as follows: we say a prefix $\vec{w} \in pref(L)$ is **infinitary** in L, if there are infinitely many $\vec{v} \in \Sigma^*$, such that $\vec{w}\vec{v} \in L$. We surely do not want to require that all prefixes in a language are infinitary: that would even exclude a language such as a^*b (as $b \in pref(a^*b)$. What we consider reasonable is that for every $\vec{w} \in pref(L)$, there is a $\vec{v} \in pref(w)$ such that 1. $\vec{v}$ is infinitary in L, and 2. $|\vec{w}| - |\vec{v}| \leq k$ for some fixed k. This leads to our next definition:

Definition 152 *A language L is **infinitarily k-prefix closed** (in k-IPC), if for each $\vec{w} \in pref(L)$, there is a $\vec{v} \in pref(\vec{w})$, such that 1. $\vec{v}$ is an infinitary prefix in L, and 2. $|\vec{w}| - |\vec{v}| \leq k$*

Lemma 153 *For all $k \in \mathbb{N}$,*

1. *k-IPC has the finite language property, and*

2. *k-IPC does not have the partial language property.*

Proof. 1. is trivial, as k-IPC requires languages to be infinite. For 2. just, just consider: given a partial language (I_1, I_0), we just complete it to $L := \{\vec{v}\vec{w} : \vec{v} \in I_1, \vec{w} \in \Sigma^*, \vec{v}\vec{w} \notin I_0\}$. This is a completion, and as I_0 is finite, it is firstly infinite, and secondly every prefix of it is infinitary. □

So k-IPC by itself is unsatisfying. To see another example, take the language $\{a^{k'}, b^n : n \in \mathbb{N}\}$. If $k' > k$, this language is not in k-IPC. But the requirement is still stronger: k-IPC actually requires that every prefix of the language has an at most k-shorter prefix, such that this prefix is infinitary. To put in a simplified fashion, if there is a "dead end" in our word which *forces* us to quite after a bounded number of steps, then there is an upper bound to the length k of this "dead end".

This is a very satisfying and reasonable restriction. But again, we should look for an extension of this property which is closed under inversion. The inverse property is quickly defined as follows: put $suf(L) = \{\vec{v} : \vec{w}\vec{v} \in L\}$. We say $\vec{w} \in suf(L)$ is an infinitary suffix, if there are infinitely many $\vec{v} \in \Sigma^*$ such that $\vec{v}\vec{w} \in L$.

Definition 154 *A language L is **infinitarily k-suffix closed** (in k-ISC), if for each $\vec{w} \in suf(L)$, there is a $\vec{v} \in suf(\vec{w})$, such that 1. $\vec{v}$ is (suffix-) infinitary in L, and 2. $|\vec{w}| - |\vec{v}| \leq k$.*

It is easy to see that this is exactly the dual notion (under inversion). Putting the two together, we get:

Definition 155 *A language is infinitarily k-closed (in k-IC), if it is 1. in k-IPC, and 2. in k-ISP.*

Lemma 156 *If $L \in k$-IC, then $L^{-1} \in k$-IC.*

Proof. Easy to see, because $L \in k$-IPC iff $L^{-1} \in k$-ISC. □

Note that, as above, we always have to work with fixed k, not with arbitrary finite k; the reason is that otherwise we lose the FLP! There are some simple results we obtain:

Lemma 157 *1. k-IC is closed under ϵ-homomorphism.*
2. k-IC is closed under union.

Proof. 1. By definition of a homomorphism, they preserve prefixes: if $\vec{v} \in pref(\vec{w})$, then $h(\vec{v}) \in pref(h(\vec{w}))$. Moreover, length is preserved or diminished: $|h(\vec{w})| \leq |\vec{w}|$.

2. For all $\vec{w} \in pref(L \cup L')$ we have either $\vec{w} \in pref(L)$ or $\vec{w} \in pref(L')$, so the claim follows; same for suffixes. □

Closure under intersection and complement can be easily shown to be false, because the intersection of two languages in k-IC can be finite, and no finite language can be in $k - IC$. We now put it all together:

Definition 158 *A language is k-structured (in k-S), if it is in k-IC and $ider_k$.*

Corollary 159 *For all $k \in \mathbb{N}$,*

1. *k-S has the finite language property, and*

2. *k-S has the partial language property.*

This follows *a fortiori*. Is there something else we can say about k-S? A first question to ask is: are there languages which are not regular/computable, which are in k-S for some $k \in \mathbb{N}$? This is slightly more complicated, but there is still a negative answer.

Lemma 160 *Let $\overline{w}$ be an infinite word. Then $fact(\overline{w}) \in 0$-S.*

Proof. We can check the condition: prefix and suffix-condition is fulfilled, and every word, consequently every factor, is infinitary. □

The problem is: this is not sufficient to show that there are uncountably many such languages over a given alphabet, because it is possible that $\overline{w} \neq \overline{v}$, yet $fact(\overline{w}) = fact(\overline{v})$ (see [46]). However, consider the following. Let $s : \mathbb{N} \to \mathbb{N}$ be sequence; we write s_n for $s(n)$. We define an infinite word $ba^{s_1}ba^{s_2}ba^{s_3}b...$. Call this word $\overline{w}(s)$ Obviously, for every distinct sequence we get a distinct word. But the following is more important: say a sequence s is strictly monotone increasing, if $n < m \Rightarrow s(n) < s(m)$ holds.

Lemma 161 *Let s, s' be two distinct, strictly monotone increasing sequences. Then $fact(\overline{w}(s)) \neq fact(w(s'))$.*

Proof. As $s \neq s'$, $\mathbb{N}$ is well-founded, there is a unique smallest $n \in \mathbb{N}$ such that $s(n) \neq s'(n)$. We then know that for $m < n$, we have $s(m) = s(m) < min(s(n), s'(n))$. Furthermore, for $m > n$, we have both $s(m), s'(m) > min(s(n), s'(n))$. Now assume w.l.o.g. that $s(n) < s'(n)$. It immediately follows that there is no $m \in \mathbb{N}$, such that $s'(m) = s(n)$. Then it follows that $ba^{s(n)}b \in fact(\overline{w}(s))$, but $ba^{s(n)}b \notin fact(w(s'))$. □

Now all we have to show is that there are uncountably many strictly monotone increasing sequences. This is easily done: as is well-known, we can represent a sequence as an infinite word over an infinite alphabet. Assume they are only countably many – then we can write them in a countable list. We can now apply Cantor's diagonalization argument – the only thing we need to take care of in addition is to always increase the numbers we change, so that the sequence resulting from diagonalization is still strictly monotonous increasing. This together proves the following:

Theorem 162 *There are languages in* 0*-S which are not computable.*

Proof. As there are uncountably many strictly monotone increasing sequences, there are uncountably many distinct languages of the form $fact(\overline{w}(s))$; and they are all in 0-S; consequently, most of them are not computable. □

This result shows that the k-S criterion alone is still much too liberal; but we can simply stick with our "philosophical" premise, and require that:

Claim: natural languages are regular and in k-S for some k.

The strive for generalizations on natural languages then would, for example, be the search for a smallest k. Note how different this is from the canonical approach in linguistics; from the usual machinery of modern syntax, there is almost nothing left.

6.5 A Note on Learnability

I only mention that despite appearances, finitist linguistics bears a strong relation to formal learning theory, in fact much stronger than the classical pre-theories. A concept which is very important for learnability is the notion of **finite elasticity**:

Definition 163 *A class of languages C has **infinite elasticity**, if for some infinite $L \in C$ there exists an infinite sequence $L_i : i \in \mathbb{N}$, such that for all $i \in \mathbb{N}$, L_i is finite, and we have $L_1 \subset L_2 \subset .... \subset L$.*

A class of languages has **finite elasticity**, if it does not have infinite elasticity. Finite elasticity is extremely important for learning in the limit in the sense of Gold, because provided a class C has finite elasticity, we can identify any (infinite) language in $L \in C$ *in the limit*; that is, given an infinite sequence of words $\vec{w}_i : i \in \mathbb{N}$, after reading a finite number thereof we can say: this characterizes a language $L \in C$ uniquely (this is a very intuitive and imprecise description; for formal definitions and more, consider [21],[10]).

So much for learning. Finite elasticity can however also be important for finitist linguistics. The reason is as follows: assume we have a class C with finite elasticity. In finitary linguistics, similarly as to Gold-learning, we have a

sequence of observations, which by the assumption on the infinitary nature of "language" (which the finitist shares!) is infinite. Recall that C is actually our linguistic theory. Now, as C has finite elasticity, at a certain point we know that there is only one $L \in C$ which is compatible with the data. And even before this point, there might be only one *infinite* $L \in C$ compatible with the data, or there might be a small set of languages compatible with the data. Once we have reached this point, we can easily falsify our theory C: the only thing we need is to find a $\vec{w}$ in our data, such that $\vec{w} \notin L$. So the concept underlying the classical learning theory is identification; and from (almost) unique identification, it is only a very small step to falsification.

So what we see is: learning in the sense of Gold is much more closely related to finitary linguistics than it is to classical linguistics – despite the contrary appearances!

6.6 Conclusion

We have seen that the strive for linguistic generalization in the finitist paradigm looks very different from the "classical approach". Work in this direction has been done, but most linguists would consider this work as research on "performance", which is more about memory etc. than about "language" itself; this is to say: as part of psycho-linguistics or corpus-linguistics rather than of linguistics proper. This is however not necessarily true: we have argued that in fact these approaches belong to linguistics proper, but under the assumption of a very different philosophy (or metatheory) of language.

So I hope to have achieved two things: firstly, to lay out a finitist metatheory of language. This metatheory might not only allow for substantial insights both on the empirical and theoretical side; it might also be a "philosophical backup" to some existing approaches to "language", which however are rather marginalized in theoretical linguistics. Secondly, I also hope that the formal methods presented here might might be of some help in this approach, and might serve as an inspiration for some linguist more empirically interested than I am. In particular, PLP seems to be very satisfying both from a mathematical as from a theoretical point of view; we must however take for granted that linguists gather both positive and negative data for their falsificationism. If we do disagree with this theoretical assumption, that is, deny the relevance of negative data, then things get hard, though not impossible: still the approach using classical learning theory might work for us.

Chapter 7

Conclusion and Outlook

7.1 Things that have been done

I will not try to summarize my work in this conclusion. Rather, I will first give a list of things I consider important and I think have been sufficiently addressed by this work, and then give a list of things I consider important that should be addressed in further research. In the first list, many of my maybe more "covert" motivations will figure.

My most urgent concern was maybe the following: the situation in current linguistics is such that strictly speaking, we probably cannot even speak of one field of linguistics. This is not only due to the fact that there are many facets of language, and many different phenomena, such as its social dimension, historical dimension, psychological dimension etc. As a matter of fact, there are many different approaches to the same "core phenomena" of language, which diverge both in their methods as in their goals in a tremendous way. What we completely lack nowadays is an underlying theory which ensures that people can at least talk to each other; we seem to have lost this "common denominator". One example is the following: scholars from one school of thought say it has been shown that "languages" are not context-free; scholars from another school of thought deny that this statement even has any meaning at all, as we cannot speak of "languages" as formal languages.

The first thing I hope to have achieved is exactly this: providing a common denominator for different approaches to "core linguistics". My approach was to take a step back away from the subject of linguistics, and look at how we construct it. It turned out that we can construct it in different ways, and each of these ways has a good justification and interesting consequences. So if scholars have totally different views on "language", one still can have reasonable arguments on "language", but on the metalinguistic level rather than on the linguistic level. So there is no need for the polemics which (to my impression) has become somewhat overwhelming in the communication between schools. So the first point is: communication between schools should always be possible, though on the level of metalinguistics rather than linguistics.

The second thing I wanted to achieve is: scholars making claims about the formal nature of "language" should be able to formally lay out the premises they make in deriving their consequences. Of course, this is well-known, and it should be well-known to any linguist that a statement as: X PROVES THAT NATURAL LANGUAGE ARE NOT REGULAR is way too bold: this claim only holds under certain assumptions (note on the contrary that a statement of the form: X PROVES THAT NATURAL LANGUAGE ARE NOT STRICTLY LOCAL *can* be made – finite language property). So it seems rather lack of awareness or simply laziness rather than ignorance which makes people forget to mention these assumptions. But this might to a large extent due to the fact that there are no theories about these assumptions. I hope to have provided such a theory and raised some awareness to these questions. I have the impression that formal linguists sometimes think that the projection of the language is the uninteresting part of their argument which is quickly done, while the rest of the formal argument has to be done very carefully. On the other side, when it comes to really deciding on a critical case of projection, linguists are often quite lost, as they do not have any hard criteria to guide their decisions. But neither of the two are inevitable: metalinguistics has its own interesting and complex mathematics; and assumptions we make can be formally included in any argument via the

notion of methodological universal and universal property modulo $(\mathfrak{f}, P)$.

The third thing I wanted to achieve is the following: even though there are many different views on language in their own right, this does *not* open the door to arbitrariness: each assumption and position comes with consequences, commitments and challenges. And this is an important thing: we cannot just conceive of "language" as we like it on different occasions: in order to do proper linguistics, we have to take one position and elaborate it consistently. For example: if we say we skip the whole projection, then we have to make sure our theories verify something like the finite language property or partial language property, and generalizations proceed via finitary falsificationism. Again, I think that formulating consistent positions and working within them consequently might lower the mistrust between different schools of thought. (But as a note: maybe also not; the Chomskyan paradigm is quite consistent, but still evokes most of the mistrust.)

A fourth thing I wanted to achieve is the following: the theory of linguistic formalisms has reached a very high level of abstraction and sophistication. But it sometimes seems that "real linguistics" is not catching up: linguists see too much of arbitrariness in formalizations and too many foundational problems in the languages we observe to follow into sophisticated formal arguments. Maybe the formalization of linguistic metatheory might help to lift linguistics to a more abstract level, making "real linguistics" more interesting to people with a major interest in formal methods. This of course would require that linguists (roughly) follow the arguments and methods I have laid out. And, of course, it requires that the methods of linguistic metatheory become much more elaborate than what I presented here: after all, I have only tried to lay out possible solutions, and I have not put them to the test against real datasets of languages.

A fifth point is the following: in theoretical linguistics, there is a long and ongoing debate on the ontological nature of its subject: is "language" in the mind, an abstract object, a social convention etc. For example, the entire consistency of the Chomskyan program seems to rely on the assumption that "language" is in the brain (more than in the mind) in a very strong sense, and this assumption is used to "kill" any epistemic concerns (see for example Ludlow, [47]). The main presupposition of this entire work is the priority of epistemological questions over ontological questions: it does not matter in the first place what "language" is, but rather what we can know about it; in particular, the discussions on what language *is* are meaningless if they go beyond what we can know. To make a stronger claim: any assumption on what "language" is which goes beyond what we can know is illegitimate. This view is very old in philosophy – I guess it is the essence of Kant's Critique of Pure Reason – but does not seem to have found access to linguistics yet. Sure, linguistics is a young discipline, and to promote a research program one needs strong assumptions. But in the light of the current situation, I think there is very good reason to switch to the epistemological position.

So this is my fifth main motivation: promote the priority of epistemology in linguistics. At a certain point I found it startling to find linguists speak of acceptability and its difference to grammaticality, that we only see acceptability but want grammaticality, without the slightest care about the fact that grammaticality is then no longer an empirical notion, and that consequently it is completely undefined what it means unless they explicitly define it in some way. Linguists just think they know "language" or grammaticality by the grammars

they write, and I have not found out what makes them be so sure.

One might ask: how can I promote the epistemological point of view, if I presuppose it? As far as I can see, there is no way to falsify the ontological point of view from the epistemical one or vice versa, because for the sake of any argument, we presuppose one of these positions. So the argument for one point of view is rather: taking this point of view, the questions we ask and the answers we get are richer and more meaningful than in the other. And this is the sense in which I wanted to push forward the epistemological point of view: by showing that it opens a whole new world of meaningful, fascinating questions, and that it allows to bridge the gaps between different schools of thought, which at least partly are due to different ontological assumptions and commitments.

7.2 Things that should be done

As there exists a critique of reason, can there exist a critique of linguistic reason? By a critique of linguistic reason I mean: a reasonably precise study on what we can know about "language" and what not. A part of what we have done already can be interpreted in this direction: there are many things on language we cannot know, and basically any statement which presupposes that language is infinite is a statement conditional on a pre-theory. But this is in a sense very coarse and obvious. What about a more fine grained distinction of statements which can be made and such that cannot be made? It turns out that in this view, my work is mostly concerned with the conditional statements we can make, premises and conclusions. Though it seems to be really interesting to scrutinize this line between admissible and inadmissible statements, I have not done this here, as it seems to me that it might end in a work twice as big as this one; so I leave it for further research.

A second problem I have not talked about is semantics. I think it would be very interesting and even necessary to extend the work I have done from languages to relations, thereby including semantics. However, I am doubtful that it can be achieved with the methods I have presented here: the combinatorics of relations is much more complicated than the one of languages (see [40]), and I guess we will have to rethink most of the problems in order to make them accessible for semantics. This is the reason I have not addressed this problem at all.

The third question I have not addressed at all is the following: how do we even get from the data we have to strings? I think this is an interesting question; the reason I have not addressed it here is twofold: firstly, I think this problem is very different in nature from the problem I have addressed. This means in particular that the methods which are needed are very different. The second point is: I do not think this question really interacts with the question I have addressed. Granted, to address the problems I have addressed, I presuppose there is a solution to the other problem. Still I think that the two problems can be separated quite neatly.

A fourth problem I have left partly open is the problem of what I have called linguistic intensionalism. Whereas I think I have given a satisfying treatment of the classical and finitary position, I dare not say the same about the intensional position. I have already said there is a lot more to say from a philosophical and linguistic point of view than what I have said. Yet, I also think the formalization

of intensional languages I have presented might be unsatisfying in many regards – the foremost being: it seems hard to do linguistics with it. This is a general problem of the intensional position: whereas in the classical and finitist approach, the philosophical position quite clearly determines the ontology, in the intensional approach the ontology (what is "language"?) is rather mysterious, and there are many possible answers. So it is another big challenge to work out this position to the point where I can say: whoever has this philosophical position wrt. "language" has to accept our mathematical conclusions. I hope to address this in further research.

There is a fifth problem I have not addressed at all, which concerns my work in particular, but also most of formal linguistics in general. I have not found it mentioned explicitly except for one place, and there only in connection with a formal argument (see Mohri [51]). I will call it the problem of **fragmentation**. A concrete instance is the one brought up by Mohri. Everyone reading this should be familiar with the proof of non-regularity of English; we construct a sublanguage $\{\mathsf{people}^n\ \mathsf{see}^n : n \in \mathbb{N}\}$ of English, intersect English with $\mathsf{people}^*\ \mathsf{see}^*$, and there we are. There is however a flaw in this argument: how do we know that if people^n see^m is English, then necessarily $n = m$? That is a point we have not made yet, which however is strictly necessary for the argument! And just think about

(1) People see, see, see

– syntactically, that should be as fine as

(2) People talk, talk, talk.

We will not argue about whether the argument for the particular non-regularity of English can be saved in some way or rather not. For us, the interesting point is rather: making statements of the form: "English is not a regular language" are not only made dependent on a certain projection; they also depend on the fact that we only look a certain *fragment* of English. This means: statements of this kind are not only dependent on a pre-theory, but also depend on the assumption that further data we have not yet considered will not spoil the argument. Note that neither monotonicity nor upward normality are of any use in preventing us from this problem: we either cannot help the projection of a pattern being blocked, or we cannot help it being "covered" by a "larger" pattern.

To put it simply: claims of the above type are always based on fragments. In our ontology, we never know whether we have observed all relevant data for a certain pattern. So what can we do to make them complete? There does not seem to be a solution, because we have nothing but fragments at any point (by finiteness). Could we improve the situation by making claims of the form: THERE IS A FRAGMENT OF ENGLISH SUCH THAT ...? But this is highly unsatisfying: we can also easily find a fragment of English where utterance length grows exponentially, if choose it appropriately. What we rather want to say is: the fragment is somehow *coherent*, we have not ignored any strings which naturally belong to it. That of course does not entail that there are no strings which are relevant, but of which we are not aware yet. And this seems to be the best we can do: we can say TO THE BEST OF OUR KNOWLEDGE, which means we have chosen the relevant data for projection without deliberately ignoring anything. So there remains still a further condition on which statements of the form: NATURAL

LANGUAGES ARE NOT REGULAR depend, and there is nothing we can do about it.

An issue which I should also mention is the following: in some discussions on this work,[1] we discussed the question whether linguistic metatheory can simply take over linguistics; that is: the classical tasks of linguistics (with the cognitive commitment) should be solved by linguistic metatheory, or at least with its methods. My position is the following: I think this goes too far. I have stressed for all possible positions that there is a proper difference between linguistic theory and linguistic metatheory.

In the classical approach, this difference is as follows: the linguist takes a certain dataset, and *chooses* to project it into the infinite. If we want to use this as a model how the speaker learns his language, we immediately encounter a problem: the speaker does not have this choice. He has to stay open; more concisely: he cannot just define his subject, as does the linguist – he has to respect all external constraints. As linguists, it is inevitable that we always work with fragments; for the speaker – in the classical perspective – this is impossible: a speaker by definition learns his language, so by definition he must have all relevant data and succeed on all possible (or plausible) presentations of the relevant data, at least in the usual idealized paradigm (see [10] for criticism and alternatives). The linguist on the other side is free in his decision, because he can *possibly be wrong*: he need not succeed in reconstructing the unique language a speaker knows, though this is what he strives for. So the (classical) speaker is in a much weaker position than the metalinguist; and if our methods are sufficient for metalinguistics, it is not said at all that they will also do for linguistics. That is, in short, my position: I will not say my methods are useless for linguistics, as say, for a theory of learning of the speaker. But I am not convinced they are sufficient. Just consider in how far our projections deviate from classical learning in the limit, which gives much weaker results, exactly because we cannot choose the point of projection.

In the finitist approach, there is an obvious difference between pre-theory and theory: this is because the meta-theory is basically inexistent – it is just a philosophical position, nothing more. What we have described in the above section was rather already theory.

What is maybe most delicate is the separation of linguistics and metalinguistics in the intensional meta-theory; for this reason we have discussed this issue separately. In the intensional approach, we can assume an arbitrary set of pre-theories as given from the metalinguist. From what we have seen, intensional linguistics mostly consists in in fixing i-language and defining acceptable sublanguages, as they correspond to inferences speakers can perform on the spot. In this sense, there is a very sharp boundary between meta-theory (defining the whole intensional language) and theory (e.g. defining sublanguage). The whole point of the intensional approach is: we do no longer fix "language" as what speakers actually know, but rather: as what they *can* know, if they reason. What they *do* reason in a sufficiently immediate and intuitive sense is an empirical question, and that is linguistics. So there is a sufficiently sharp distinction between linguistic theory and linguistic metatheory also in this approach.

There is also a principled concern I have regarding the idea of using the techniques of linguistic metatheory (say pre-theories) for linguistics within the

[1] Mostly with Udo Klein.

cognitive commitment. Metalinguistics as a discipline in its own right is only meaningful as long as we believe in the priority of the epistemological point of view. As soon as we convince ourselves that language is in the "mind" and we just have to see what it is like, there is no place for it. Now, if we do linguistics with the cognitive commitment, it is clear that we want to describe something in the mind of the speaker. But that in turn leads to the working hypothesis that linguistic metatheory can be neglected, if it has any meaning at all. If linguistic metatheory is irrelevant at best, that does not necessarily mean its techniques are. Still, I do not see why we should have pre-theories if we do not share the basic assumptions of linguistic metatheory. Just consider the strictly language-theoretic ontology of all metatheories. Adopting the cognitive commitment, one is quickly led to the (Chomskyan) conclusion that the part of language which we observe (that is, strings) is actually the most uninteresting of it. What is interesting from this perspective is rather the representation of language in the mind. It is immediately clear that this focus is diametrically opposed to any epistemological perspective: in the latter, we have a strong focus on the observable objects; in the former, we consider the observable objects negligible from a theoretical point of view. But why should we use pre-theories if we consider strings to be of minor interest?

So I see there is a dissonance between my approach and the commitment to a cognitive ontology, as it is prominent not only in the Chomskyan approach, but in most modern approaches to linguistics. I know that some researchers will see my major points very clearly, yet they will not easily be convinced to abandon their cognitive ontology. So as a very last point, let me stress that I do not think that the two cannot be unified: in fact, once we adopt a metatheory and construct "language", nothing prevents us from conceiving of that object as something "cognitively real". What is impossible to unify with our approach is the absolute priority of the "hard reality" of "language" in the brain over any epistemic concern.

Bibliography

[1] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[2] Henk Barendregt. *The Lambda Calculus. Its Syntax and Semantics*. Number 103 in Studies in Logic. Elsevier, Amsterdam, 2 edition, 1985.

[3] C. C. Chang and H. Jerome Keisler. *Model Theory*. North–Holland, Amsterdam, 3 edition, 1990.

[4] Noam Chomsky. *The Logical Structure of Linguistic Theory*. Plenum Press, New York, 1975.

[5] Noam Chomsky. *The Minimalist Program*. MIT Press, 1995.

[6] Alexander Clark. A learnable representation for syntax using residuated lattices. In Philippe de Groote, Markus Egg, and Laura Kallmeyer, editors, *Proceedings of the 14th Conference on Formal Grammar*, volume 5591 of *Lecture Notes in Computer Science*, pages 183–198. Springer, 2009.

[7] Alexander Clark. Learning context free grammars with the syntactic concept lattice. In José M. Sempere and Pedro García, editors, *10th International Colloquium on Grammatical Inference*, volume 6339 of *Lecture Notes in Computer Science*, pages 38–51. Springer, 2010.

[8] Alexander Clark. Logical grammars, logical theories. In Denis Béchet and Alexander Ja. Dikovsky, editors, *LACL*, volume 7351 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2012.

[9] Alexander Clark. Learning trees from strings: a strong learning algorithm for some context-free grammars. *Journal of Machine Learning Research*, to appear.

[10] Alexander Clark and Shalom Lappin. *Linguistic Nativism and the Poverty of the Stimulus*. Blackwell, 2011.

[11] Eugenio Coseriu. *Sprachkompetenz : Grundzge der Theorie des Sprechens*, volume 508 of *Tbinger Beitrge zur Linguistik ; 508*. Narr, Tbingen, 2007.

[12] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 2 edition, 1991.

[13] Adriaan de Groot. *Thought and Choice in Chess*. Mouton De Gruyter, 1978 (Reprint from 1966).

[14] Philippe de Groote. Towards Abstract Categorial Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, pages 148–155, Toulouse, 2001.

[15] Ferdinand de Saussure and Elisabeth Birk [Bearb.]. *Wissenschaft der Sprache. Neue Texte aus dem Nachlaß*. Suhrkamp-Taschenbuch Wissenschaft ; 1677. Suhrkamp, 2003.

[16] Michael Devitt. *Ignorance of Language*. Clarendon Press, Oxford, 2006.

[17] S. Eilenberg, C. C. Elgot, and J. C. Shepherdson. Sets recognized by n-tape automata. *Journal of Algebra*, 13:447–464, 1969.

[18] Roland Fraïssé. *Theory of Relations*. Studies in logic and the foundations of mathematics ; 118. North-Holland, 1986.

[19] Nikolaos Galatos, Peter Jipsen, Tomasz Kowalski, and Hiroakira Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Elsevier, 2007.

[20] Edward Gibson. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68:1–76, 1998.

[21] Mark E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[22] Hubert Haider. Grammatische Illusionen: Lokal wohlgeformt, global deviant. *Zeitschrift für Sprachwissenschaft*, 30:223–257, 2011.

[23] Roy Harris. *The Language-Makers*. Duckworth, London, 1980.

[24] Zellig S. Harris. *Structural Linguistics*. The University of Chicago Press, 1963.

[25] Roger Hindley. *Basic Simple Type Theory*. Number 42 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 2008.

[26] Jaakko Hintikka. *Knowledge and Belief. An Introduction into the logic of the two notions*. Cornell University Press, Ithaca, 1962.

[27] R. Huybregts. Overlapping Dependencies in Dutch. *Utrecht Working Papers in Linguistics*, 1:3–40, 1984.

[28] Aravind K. Joshi and K. Vijay-Shanker. Compositional semantics with Lexicalized Tree-Adjoining Grammar (LTAG): How much underspecification is necessary? In H.C. Bunt and E.G.C. Thijsse, editors, *Proc. IWCS-3*, pages 131–145, 1999.

[29] Makoto Kanazawa. Indentification in the limit of categorial grammars. *Journal of Logic, Language and Information*, 5(2):115–155, 1996.

[30] Makoto Kanazawa. Second-order Abstract Categorial Grammars as Hyperedge Replacement Grammars. *Journal of Logic, Language and Information*, 19(2):137–161, 2010.

[31] Jerrold J. Katz. *Language and Other Abstract Objects*. Basil Blackwell Publisher, Oxford, 1981.

[32] Jerrold J. Katz and Paul M. Postal. Realism vs. conceptualism in linguistics. *Linguistics and Philosophy*, 14:515–554, 1991.

[33] Kevin T. Kelly. Uncomputability: the problem of induction internalized. *Theor. Comput. Sci.*, 317(1-3):227–249, 2004.

[34] Steven C. Kleene. *Introduction to Metamathematics*. North–Holland, Amsterdam, 1964.

[35] Gregory M. Kobele. *Generating Copies: An investigation into structural identity in language and grammar*. PhD thesis, UCLA, 2006.

[36] András Kornai. Natural languages and the chomsky hierarchy. In *Proceedings of the 2nd European Conference of the ACL 1985*, pages 1–7, 1985.

[37] Marcus Kracht. Syntactic Codes and Grammar Refinement. *Journal of Logic, Language and Information*, pages 41–60, 1995.

[38] Marcus Kracht. *Mathematics of Language*. Mouton de Gruyter, Berlin, 2003.

[39] Marcus Kracht. Gnosis. *J. Philosophical Logic*, 40(3):397–420, 2011.

[40] Marcus Kracht. *Interpreted Languages and Compositionality*. Springer, 2011.

[41] Manfred Krifka. In defense of idealizations: A commentary on Stokhof and van Lambalgen. *Theoretical Linguistics*, 37,1-2:51–62, 2011.

[42] Saul A. Kripke. *Wittgenstein on Rules and Private Language. An Elementary Exposition*. Blackwell, Oxford, 1982.

[43] William Labov. *Principles of Language Change: Social Factors*, volume 29 of *Language in society*. 2001.

[44] Winfred P. Lehmann. *Historical linguistics : an introduction*. Routledge, London [u.a.], 3. ed. edition, 1992.

[45] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge (UK), 1995.

[46] M. Lothaire. *Algebraic Combinatorics on Words*. Encyclopedia of mathematics and its applications ; 90. Cambridge Univ. Press, 2003.

[47] Peter Ludlow. *Philosophy of Generative Linguistics*. Oxford University Press, Oxford, 2011.

[48] Alexis Manaster-Ramer. Copying in natural languages, context–freeness and queue grammars. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 85–89, 1986.

[49] André Martinet. *Économie des Changement Phonétiques*. Maisonneuve & Larose, Paris, 2005, 1st edition: 1955.

[50] Jens Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Universität Potsdam, 2001.

[51] Mehryar Mohri and Richard Sproat. On a common fallacy in computational linguistics. In Mickael et.al Suominen, editor, *A Man of Measure: Festschrift in Honour of Fred Karlsson on this 60th Birthday*, volume 19 of *SKY Journal of Linguistics*, pages 432–439. 2006.

[52] Richard Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven and London, 1974. edited by Richmond H. Thomason.

[53] Glyn Morrill, Oriol Valentín, and Mario Fadda. The displacement calculus. *Journal of Logic, Language and Information*, 20(1):1–48, 2011.

[54] Frederick J. Newmeyer. *Possible and Probable Languages: A Generative Perspective on Linguistic Typology*. OUP, Oxford, 2005.

[55] P. Reich. The finiteness of natural language. *Language*, 45:831 – 843, 1969.

[56] Luigi Rizzi. *Issues in Italian Syntax*. Foris, Dordrecht, 1982.

[57] James Rogers. *Studies in the Logic of Trees with Applications to Grammar Formalisms*. PhD thesis, Department of Computer and Information Sciences, University of Delaware, 1994.

[58] James Rogers. *A Descriptive Approach to Language-Theoretic Complexity*. Studies in Logic Language and Information. FoLLI, 1999.

[59] James Rogers. Cognitive and sub-regular complexity. In *Proceedings 17th Conference on Formal Grammars, 2012*, to appear.

[60] Sasha Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14(2):169–209, 2008.

[61] Ferdinand de Saussure. *Cours de Linguistique Générale*. Payot & Rivage, Paris, 5 edition, 2005.

[62] Thomas Schack. Building blocks and architecture of dance. In *Neurocognition of Dance*. Psychology Press, Dordrecht, 2009.

[63] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context–free grammars. *Theor. Comp. Sci.*, 88:191–229, 1991.

[64] A. Sestier. Contributions à une théorie ensembliste des classifications linguistiques. (Contributions to a set–theoretical theory of classifications). In *Actes du Ier Congrès de l'AFCAL*, pages 293–305, Grenoble, 1960.

[65] Edward P. Stabler. The finite connectivity of linguistic structure. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, pages 303–336. Lawrence Erlbaum, 1994.

[66] Edward P. Stabler. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics (LACL '96)*, number 1328 in Lecture Notes in Artificial Intelligence, pages 68–95, Heidelberg, 1997. Springer.

[67] Martin Stokhof and Michiel van Lambalgen. Abstraction and idealization. the construction of modern linguistics. *Theoretical Linguistics*, 37,1-2:1–26, 2011.

[68] Peter Trudgill. *Sociolinguistic Typology : Social Determinants of Linguistic Complexity*. Oxford linguistics. Oxford Univ. Press, Oxford, 2011.

[69] Theo Vennemann. An explanation of drift. In C. N. Li, editor, *Word Order and Word Order Change*, pages 269–305. University of Texas Press, Austin and London, 1975.

[70] Christian Wurm. Modularization of regular growth automata. In Matthieu Constant, Andreas Maletti, and Agata Savary, editors, *FSMNLP*, ACL Anthology, pages 3–11. Association for Computational Linguistics, 2011.

[71] Christian Wurm. Concepts and types - an application to formal language theory. In Laszlo Szathmary and Uta Priss, editors, *CLA*, volume 972 of *CEUR Workshop Proceedings*, pages 103–114. CEUR-WS.org, 2012.

[72] Christian Wurm. Completeness of Full Lambek calculus for syntactic concept lattices. In *Proceedings of the 17th Conference on Formal Grammar, Springer Lecture Notes in Computer Science*, in press.