A formal theory of frames with an application to plural quantification and cross-sentential anaphora

Ralf Naumann and Wiebke Petersen

Institut für Sprache und Information
Universität Düsseldorf
Germany

Abstract. In this article we present a formal theory of frames and apply it to two empirical phenomena: cross-sentential anaphora arising with plural indefinite NPs and resolution of pronouns based on bottom-up information. In the first part frames are defined as minimal submodels of possible worlds with the latter being first-order models based on an order-sorted domain with a relational structure that is similar to Peirce algebras. In the second part we adapt this definition to type logic using Van Eijck's Incremental Dynamics. In contrast to previous approaches like Van den Berg (1996) and Brasoveanu (2008) the notion of dependency arising in cross-sentential anaphora is defined purely semantically in terms of chains in (event) frames. Pronoun resolution based on bottom-up information is defined solely in terms of the sortal and relational structure of frames associated with possible antecedents and does therefore not rely on knowledge of the actual truth of sentences.[1]

1 Introduction

According to Barsalou (1999, 1992), frames constitute the general format of representations in the human cognitive system. As noted in Löbner (2014, p.23), this thesis combines in essence two separate hypotheses.

(1) H1: The human cognitive system operates with a single general format of representation.
H2: If the human cognitive system operates with one general format of representations, this format is essentially Barsalou frames.

Whereas the first thesis claims that there is a *single* format of cognitive representation, the second thesis singles out frames as this unique format. Following Löbner (2014), the combination of the two theses will be called the *Frame Hypothesis*. For Barsalou, frames are elements of Long Term Memory (LTM). They are construed from perceptual encounters with objects of a particular kind.

[1] The research was supported by the German Science Foundation (DFG) funding the Collaborative Research Center 991.

Barsalou (1999) illustrates this process with the example of a car. On a first encounter with a car particular components of it are extracted. This can be the car's overall shape, its tyres or its doors. The result is a first, approximate frame for a car. In this frame each extracted component is represented by its corresponding value. This can be represented in terms of pairs consisting of an attribute (or a label) and a value. It is important to note that the values need not be words (or their mental correlates) but are taken as 'subsets of perceptual states in sensory-motor systems. [...] As a result, the internal structure of these symbols is modal, and they are analogically related to the perceptual states that produced them', (Barsalou, 1999, p.578).[2] Additional encounters with cars will lead to two kinds of principle changes of the first approximate frame. First, further attributes are added, e.g. for the handles of the doors, the engine or its brand name. Second, the range of admissible values for an attribute is enlarged. For example, a car can have different shapes and its doors and their handles can be of a variety of different materials, sizes and colors.

The important point is that upon perceptually encountering a further instance of a car or upon reading or hearing the word 'car', not only the sortal information **car** is retrieved from Long Term Memory but also information about objects to which a car is normally related, e.g. its material parts, its brand or (the color of) its tyres. This use of a frame can also be given a procedural twist: retrieving information about an object or the concept expressed by it always allows a transition to other objects which are related to the first.[3] Cognitively, such transitions can be taken as corresponding to the retrieval of other units stored in Long Term Memory into Working Memory. Hence, information retrieval is not restricted to sortal information which may informally be expressed by saying that an object falls under a particular concept or belongs to a particular class or set. In Figure 1 a possible frame for the concept 'car' is given.

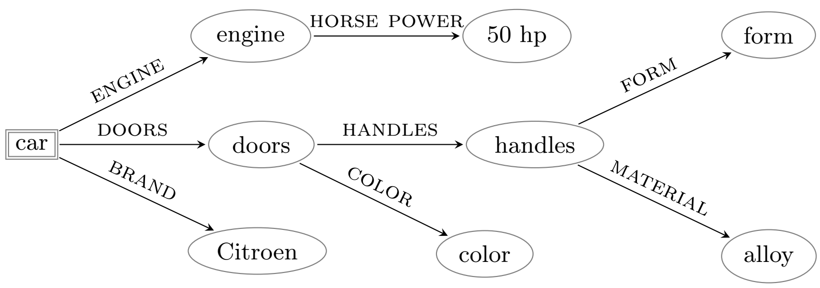

Fig. 1. Frame for a car

[2] A similar point is made in Löbner (2013, pp.302). He discusses a frame for the concept 'passport'. Such a frame will have an attribute FACE the value of which will be a photograph of the bearer of the passport.

[3] In addition, such transitions can either be strict (entailments) or non-strict (probabilistic).

This way of defining the meaning of a concept is in marked contrast to the way concepts expressed by common nouns and verbs are represented in model-theoretic semantics. For example, common nouns like 'orange' or 'paper' are basically analyzed as sets of objects. For example, 'orange' is first translated as the lambda-term $\lambda x.orange(x)$, which, in a second step, is interpreted as a subset of the domain, or, more precisely, as a function from this domain to the set of truth values (2-a). Similarly, using an event-based approach, verbs like 'run' are interpreted as sets of events or the corresponding characteristic function (2-b).

(2) a. $[\![orange]\!]^{\mathcal{M}} = (\lambda x {\in} D_{\langle e_{object}\rangle}\ f_{orange}(x){=}1)$
b. $[\![run]\!]^{\mathcal{M}} = (\lambda e {\in} D_{\langle e_{event}\rangle}\ f_{run}(e){=}1)$

These examples show that in model-theoretic semantics only sortal information is represented with respect to those concepts: x is an orange, y is a running event. Knowledge of such sortal information is inherently non-relational in the sense that (i) it is related to subsets of the universe $D_s \subseteq D$ where D_s is the set of objects of type s and D the global universe and, therefore, (ii) no information about other objects can be deduced from this information. Hence, the fact that objects are related to other objects and that they are therefore 'structured' is not accounted for. A first step in integrating relational information into lexical semantics are decompositional approaches. For example, in a neo-Davidsonian approach, 'run' is analyzed as (3).

(3)

$$[\![run]\!]^{\mathcal{M}} = (\lambda y {\in} D_{\langle e_{object}\rangle} \lambda x {\in} D_{\langle e_{object}\rangle} \lambda e {\in} D_{\langle e_{event}\rangle}\ f_{run}(e){=}1 \\ \wedge f_{goal}(e,y){=}1 \ \wedge\ f_{actor}(e,x){=}1)$$

However, in such approaches relational information is driven by (syntactic) argument structure so that relational information is restricted to arguments of an item. More importantly, such a decomposition does not capture possible dependencies between relations.[4] Furthermore, frames are neither part of the object (description) language nor are they a separate domain of the models used to interpret the object (description) language.

Quite independently of any overt cognitive underpinnings, this limitation of model-theoretic semantics has given rise to an interest in developing a formal frame theory and in integrating such a theory in model-theoretic semantics. For example, Cooper (2010, p.2) criticizes classical model-theoretic semantics because semantic objects are either atoms or unstructured sets and functions. Frames, by contrast, are structured objects. This criticism is echoed by Muskens (2013, p.175), according to whom frame theory offers the possible gain of replacing the rather crude semantics usually given to open class words by a much subtler one, giving rise to the hope that the richer structure thus obtained can be put to good use in explaining features of phrasal composition. This raises, of course, the question what a theory of frames has to offer compared to current

[4] See section 2.1 below for details.

formal semantic theories. Furthermore, as noted by Löbner (2014), the Frame Hypothesis is only attractive if it comes with a hypothetical concrete model of this general format. Such a model has to fulfill at least two requirements: (i) it must be sufficiently expressive to capture the diversity of representations which the human cognitive system is assumed to employ, and (ii) the model must be sufficiently precise and restrictive in order to be testable (Löbner, 2014, p.23). An additional question is whether the notion of a frame applies in effect to all kinds of concepts as the Frame Hypothesis requires. At least Muskens (2013, p.176) is quite explicit in rejecting the idea that all natural language meaning can be represented with the help of frames. For example, in contrast to Barsalou (1999) such concepts as quantification, disjunction and negation are not analyzed in terms of frames by him.

The rest of the article is organized as follows. In the next section we provide evidence for one use of frames. The focus is on the notion of dependency, i.e. relations between the elements of sets introduced by indefinite and quantified NPs, that plays a prominent role in cross-sentential anaphora. We critically discuss previous accounts of this notion and informally discuss how the use of frames can solve the theoretical and empirical problems encountered by these accounts. In section 3 we define frames as minimal submodels of possible worlds satisfying particular constraints. Possible worlds are order-sorted first-order models. Whereas possible worlds are based on a global signature, frames are based on specific signatures, so-called frame signatures, which can be taken as partial definitions of a concept. Frames of a particular sort, say 'person', form a hierarchy. Minimal elements of this hierarchy only provide sortal information, e.g. 'it is a person'. Extensions of a minimal frame provide in addition information about attributes, i.e. relational information, and the constraints on the target sort. Elements in the hierarchy therefore correspond to sets of (prefix-closed) chains of attributes which impose a restriction on the sort at the end of the chain. The structure of a frame is similar to a two-sorted Peirce algebra in which sets and relations not only co-exists but are in addition related by so called modes and projections which allow to go from a relation to a set or from a set to a relation.

This definition of a frame is adapted to Van Eijck's Incremental Dynamics in section 4. Following standard model-theoretic semantics, possible worlds are taken as a separate domain of (structureless) objects. Similarly, frames are (structureless) objects of a second domain of frames which are related to possible worlds in a particular way. The relation between a frame and its sortal and relational structure is captured in a way similar to two-sorted type theory. Sortal and relational formulas are interpreted not as subsets and binary relations on the domain D of individuals and events but as relations between objects and frames/possible worlds and relations between pairs of objects and a frame/possible world. At the discourse level elements of a context (stack in Incremental Dynamics) are sets of pairs $\langle o, f_o \rangle$ consisting of an object and an associated frame which contains the sortal and relational information about o got in the discourse.

In the two final sections we show how our frame theory can be used to account for dependent and independent readings arising in cross-sentential anaphora with plural (indefinite) NPs as well as to account for pronoun resolution based on bottom-up processing.

2 Evidence for frames: dependency information in cross-sentential anaphora

We start by critically reviewing the major approach for capturing dependency relations involving plural NPs due to Van den Berg (1996) and Brasoveanu (2008) in section 2.1 and 2.2. This will lay the ground for an informal discussion in section 2.3 of how frames can be used to solve the problems faced by this approach.

2.1 Applying frames: storing dependencies between sets of objects

In the context of plural quantification, cross-sentential anaphora often involve the retrieval of dependencies between sets that have already been introduced in the preceding context. Take, for example, the two sentences in (4-a).

(4) a. Three students wrote an article together. They sent it to L&P.
    b. Three students each wrote an article. They sent it to L&P.

The first sentence in (4-a) gets a collective reading ('together'). Hence, there is a single article jointly written by three students. The most prominent reading of the second sentence in (4-a) is the collective one. The three students sent their article together to the journal. On a distributive reading of this sentence the article was sent separately by each of the three students so that the editors received a total of three copies of the article. For both readings, the anaphoric relation between 'it' and 'an article' can be established without reference to the set of students introduced in the first sentence. Such anaphoric relations will be called 'independent' (Nouwen, 2003). The first sentence in (4-b) gets a distributive reading ('each') so that there is a total of three articles, one for each of the three students. A collective reading of the second sentence is not possible because the cardinality constraint imposed by 'it' is not satisfied since there are three articles and not only one. By contrast, on a distributive reading of the second sentence each student sent his (own) article to L&P. On this reading there is a distribution over the anaphorically retrieved set of three students as the antecedent of 'they'. Inside the distributive loop each student has to be correlated with his or her own article. The sentence is false if each student sent one of the other articles to L&P. Hence, the VP is interpreted w.r.t. each student separately so that the cardinality constraint imposed by 'it' is satisfied. The important point to note is the following: In this context not only the set of three students and the set of three articles have to be retrieved but also the correlation between the two sets. This use of a pronoun will be called 'dependent'. The upshot of this example is that it must be possible to store in an information state not only sets of objects but also dependencies between the elements of these sets.

The question that needs to be answered is: where in the semantic representation are such dependencies stored? In dynamic semantics information states are standardly taken to be (partial) assignments, i.e. (partial) functions from the set VAR of variables into the domain D of the model. In a plural setting, one can lift assignments in such a way that they take their values in the power set $\wp^+(D)$ in order to countenance plural objects. However, as first shown in van den Berg (1996), this move does not solve the problem of storing dependencies. To see this, consider the assignment in Table 1 which could be the result of processing sentence (4-b). In variable x the set of three students and in variable y the set of articles is stored.

x (*student*)	y (*article*)
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$

Table 1. Storing plural objects in variables

Though each plural object is stored, the dependencies between the objects are not. The solution to this problem, first proposed in van den Berg (1996) and taken up in Brasoveanu (2008), is to use sets of assignments instead of a single one. Each assignment now stores the dependency between a student and his or her article. Table 2 shows such a set of assignments $I = \{g_1, g_2, g_3\}$, the rows correspond to the assignments in the set and the columns to the variables which are in the domain of the assignments.[5] Each assignment stores a single student in x and a single article in y (local level). As a result, the dependency between student s_i and article a_i is stored in assignment g_i.

	x (*student*)	y (*article*)
g_1	s_1	a_1
g_2	s_2	a_2
g_3	s_3	a_3

Table 2. Storing dependencies in sets of assignments

Globally, the value stored in x is the set of objects stored in the single assignments of the information state. This global value of x is called the *projection* of I onto x and is defined in Definition 1.

Definition 1 (Projection.) *Given a set of assignments I and a variable x, the projection of I onto x is defined by*

$$I(x) := \{d \mid \exists g{\in}I\ g(x){=}d \wedge d{\in}D\}.$$

[5] It is assumed that one has: $\forall g_i, g_j{\in}I\ dom(g_i){=}dom(g_j)$ for I a set of assignments.

$I(x)$ is called the global or the cumulative value of x.

For instance, in the above example one has $I(x) = \{s_1, s_2, s_3\}$ and $I(y) = \{a_1, a_2, a_3\}$. Using plural information states, domain-level plurality is related to the columns. For each variable x, $I(x)$ is the collection stored in x relative to I. Discourse-level plurality or dependencies are stored 'distributively' in the rows of a matrix. Consider x and y in the matrix above. The relation stored in I relative to these two variables is given by $\{\langle g_1(x), g_1(y)\rangle, \langle g_2(x), g_2(y)\rangle, \langle g_3(x), g_3(y)\rangle\} = \{\langle s_1, a_1\rangle, \langle s_2, a_2\rangle, \langle s_3, a_3\rangle\}$. More generally, one has Definition 2.

Definition 2 (Projection onto a relation.) *The binary relation R_{xy} associated with the collections $I(x)$ and $I(y)$ stored in x and y is defined by*

$$R_{xy} := \{\langle d, d'\rangle \mid \exists g{\in}I\ g(x){=}d \wedge g(y){=}d' \wedge d, d'{\in}D\}.$$

The formal definition of the notion of dependency is based on two operations on plural information states: projection relative to a variable already defined above in Definition 1 and substate w.r.t. to a variable and a value.

Definition 3 (Substate) *The substate operation returns for a given variable x and value d all assignments in the plural information state that assign d to x.*

$$I|_{x=d} := \{g \in I \mid g(x){=}d\}.$$

The notion of dependency of one variable y on another variable x is defined as follows.

Definition 4 (Dependency) *In a plural information state I variable y is dependent on variable x iff*

$$\exists d, e{\in}I(x)\ I|_{x=d}(y) \neq I|_{x=e}(y).^{6}$$

According to Definition 4, variable y is dependent on variable x if there exists a pair of substates which assign different values to x s.t. the cumulative value assigned to y in relation to the substate $I|_{x=d}$ is different from the cumulative value assigned to y in relation to the substate $I|_{x=e}$. Consider the following example plural information state.
In this plural information state y is dependent on x. To see this, consider the two substates $I|_{x=s_1} = \{g_1\}$ and $I|_{x=s_2} = \{g_2\}$. Considering the cumulative value for the variable y in these two substates, one notes that they are not identical: $I|_{x=s_1}(y) = \{a_1\} \neq \{a_2\} = I|_{x=s_2}(y)$.

[6] Remember that $I|_{x=d}(y) = \{d' \mid g \in I \wedge g(x) = d \wedge g(y) = d'\}$.

	x	y
g_1	s_1	a_1
g_2	s_2	a_2
g_3	s_3	s_3

2.2 Merits and problems of the formal definition of discourse dependency

The central merit of using plural information states is of course that they make it possible to capture many instances of dependencies that arise at the level of discourse-level plurality. In this section we will discuss two problems which arise for the definition of plural information states as given above: *information downdate*[7] and *spurious dependencies*[8].

The first problem of information downdate is related to storing sets of objects 'vertically' in the columns of a plural information state. This way of storing plural objects leads to problems in distributive loops. Consider a variant of an example already used above.

(5) Three students each wrote an article. They each sent them to L&P.

The pronoun 'them' in the second sentence refers to the set of three articles, each of which was written by exactly one of the students. Since the second sentence is interpreted distributively, each student sent all three articles to L&P (three sending events; the editors received three copies of each article). However, in the distributive loop this set is not available. This is so because in the loop the VP 'sent to L&P' is interpreted w.r.t. the input state $I|_{x=s_i}$. But $I|_{x=s_i}(y) = \{a_i\}$ and not $\{a_1, a_2, a_3\}$. Hence, the downdate consists in the fact that the set $\{a_1, a_2, a_3\}$ is no longer available though it is an element of the whole plural information state. The conclusion one has to draw from this problem is that inside a distributive loop both the global and the dependent values must be available for anaphoric reference in order to avoid information downdate.[9]

The problem of spurious dependencies arises if to a given dependency another dependency is added. To see this, consider the example in (6) below and assume

[7] See Nouwen (2003), who first discussed this problem.
[8] See also Nouwen (2003) and Krifka (1996).
[9] The problem of information downdate inside a distributive loop also applies to Krifka's approach of parameterized sum individuals (Krifka, 1996). Parameterized sum individuals are plural individuals together with an assignment function which collects the objects dependent on the elements of the sum individual. As noted in Nouwen (2003, pp.99), the problem is similar to that for the approach in Van den Berg (1996): Once a variable is subordinated relative to another variable, its values can only be accessed through the latter variable. Inside a distributive loop, only one element of the plural object is accessed and as a result only its dependencies but not that of the whole plural individual to which the loop object belongs are available for anaphoric reference.

that s_1 and s_2 are students, s_1 read book b_1 and wrote article a_1 whereas s_2 read book b_2 and wrote article a_2.

(6) Two students each read a book and each wrote an article.

A possible information state is depicted below in Table 3.

	x	y	z
g_1	s_1	b_1	a_1
g_2	s_2	b_2	a_2

Table 3. Plural information state with spurious dependency

Now there is a dependency between the articles and the books: $I|_{y=b_1}(z) = \{a_1\} \neq \{a_2\} = I|_{y=b_2}(z)$. Note that this dependency has not been explicitly introduced (say, by a distribution operator). Let us call such dependencies *spurious dependencies*. The question one has to ask w.r.t. to this kind of dependencies is whether they can lead to wrong or implausible interpretations. The example (7) can be taken to show that the answer is positive.

(7) Two students$_1$ each collaborated with a fellow$_2$. They$_2$ each collaborated with a professor$_3$.

A possible information state is depicted below in Table 4.

	x	y	z
g_1	s_1	f_1	p_1
g_2	s_2	f_2	p_2

Table 4. Plural information state with spurious dependency 2

Now assume that the students happen to have collaborated each with that professor the fellow they collaborated with collaborated with, i.e. s_1 collaborated with p_1 and s_2 with p_2. Since there is a dependency in the model, the following sentence should be fine.

(8) They$_1$ each collaborated with him$_3$ as well.

However, this dependency has never been explicitly introduced in the discourse so that this sentence seems highly inappropriate in this context.[10]

[10] See also Nouwen (2003, pp.103) for a similar argument in a different context. We have built our last example on the one used by him.

Let us analyze the two problems in more detail. Plural objects are stored 'vertically' in an information state. They are the cumulative value that is got by taking the value of this variable in each element of the information state. An immediate consequence of this way of storing plural objects is that by going from an information state to one of its proper substates, this value is no longer available because only a proper subset is stored in that substate. Another way of stating this property is that plural objects are stored globally in the whole information state. This storage mechanism does not comply with the way dependencies are defined. In order to access a dependency one has to go to a proper substate because dependencies are defined between the elements of two sets and not between the sets themselves. Hence, a dependency can be said to be defined locally, in single assignments and hence in the elements of an information state. This problem leads to the first requirement on plural objects and dependencies.

(9) Plural objects and dependencies must be stored in such a way that they can be accessed independently of each other.

One way to comply with this requirement is to store sets of objects in a variable and to define dependency semantically. How this can be done is shown below in section 2.3.

Though a single dependency is stored 'horizontally' in a single assignment, relative to this assignment it is defined globally in the following sense. A dependency relation can, at least in principle, be defined for any pair $\langle x, y \rangle$ of variables (or n-tuple $\langle x_1, \ldots, x_n \rangle$) in the domain of an assignment as long as they pass the definition of dependency. As a consequence, arbitrary relations between variables (or discourse referents) can be considered, independently of whether they are triggered by a relation that was explicitly used in the discourse. This arbitrariness leads to the problem of spurious dependencies. One way of analyzing this problem is that dependency is defined purely structurally. Any pair of variables satisfying the definition exhibits a dependency. This completely neglects two important aspects of the notion of dependency. First, it is a semantic notion since it relates objects in a particular way. Second, it does not distinguish between such relations that trigger a dependency and such that have been used in a discourse. This yields the second requirement on plural objects and dependencies.

(10) Dependency information must not be defined in terms of arbitrary relations between variables in use. Rather, this information must be defined in terms of relations that have been explicitly used in the discourse.

One way of reading this requirement is to say that dependencies must satisfy a constraint similar to objects. Anaphoric reference to an object is possible only if it has been introduced into the discourse before (cf. Partee's famous example of the missing marble). Similarly, a dependency must have been explicitly introduced in order for it to be accessed later on. The structural definition of dependency fails to satisfy this constraint. Though a dependency is always defined relative to variables in use, this need not be the case for the relations on which the dependency is based. In the next section we will informally show how frames can be used to avoid the two problems and to comply with the two requirements.

2.3 Dependencies in a frame theory

Consider again the example of three students each writing an article. Recall that the dependency relation between the students and the articles is given by the relation $R_{xy} = \{\langle s_1, a_1\rangle, \langle s_2, a_2\rangle, \langle s_3, a_3\rangle\}$. Consider next the interpretation of 'write' in an event-based framework.

(11) $[\![write]\!]^{\mathcal{M}} = \lambda y \in D_{\langle e_{object}\rangle} \lambda x \in D_{\langle e_{object}\rangle} \lambda e \in D_{\langle e_{event}\rangle}\ f_{write}(e){=}1 \wedge f_{theme}(e, y){=}1 \wedge f_{actor}(e, x){=}1$

A minimal frame associated with this interpretation has two attributes: ACTOR and THEME.

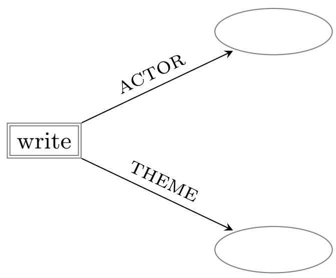

Fig. 2. Frame for 'write'

The frame in Figure 2 corresponds to one of the three writing events. Inspecting this frame, one notices that the article is not only related to the writing event by the THEME-attribute but the author is also related to his written article by the path $\text{ACTOR}^{-1} \bullet \text{THEME}$. Hence, the dependency between s_1 and a_1 is stored in this frame as the value of the path connecting the actor with the writing event and this event with the theme. Generalizing this example, one arrives at the following thesis.

(12) *Dependency Thesis:* Dependencies which arise in the context of intersentential anaphora with plural quantification are stored in event frames. They result from combining two or more attributes via attribute chaining or sequencing (corresponding to functional composition) $\bullet$ and the inverse operation $^{-1}$.

How are frames related to objects? Each object introduced in the current information state is paired with a frame. The elements of an information state are therefore not simply objects but pairs consisting of an object and a frame. The first component stores the objects one is talking about in the current discourse whereas the second component contains both relational and dependency information about these objects. In a plural setting one gets sets of pairs consisting of an object and a frame. Using frames, objects and dependency information are stored separately in the two components of the value of a variable. Furthermore, the relations on which dependencies are based are not defined structurally but arise from (chains of) attributes that are defined for frames of a particular sort and which have been introduced into a discourse. As a result, the two requirements are satisfied and the two problems which approaches using sets of assignments face are avoided.

Let us informally sketch how the two kinds of information are used in the interpretation of the distributive reading of (4-b). The indefinite 'three students' adds a set of three pairs $\langle s_i, f_i \rangle$ consisting of a student and a (minimal) student frame. On a distributive reading the VP 'wrote an article' is interpreted relative to each element in this set, i.e. relative to each of the three students separately. During this process a set of three writing events and a set of three articles is introduced, one for each student. As a result, there is a total of three writing events each of which stores the dependency between one of the students and the article (s)he wrote in the path $Actor^{-1} \bullet Theme$. Note that this path is explicitly stored in the frames associated with the writing events. This shows that dependencies can be based, and are usually based, on relations that are introduced using a decompositional analysis. Assuming that 'they' is anaphoric to 'three students',[11] there is again a distributive loop over the set of three students. Inside this loop the current information state is temporarily extended by the dependencies established for the object over which the loop runs.[12] In the case at hand this is the i-th article if the loop is over the i-th student. This extension has the effect that both the set of three articles and the article which is dependent on the student become available for an anaphoric reference to a pronoun inside the VP.

One may object that the dependency argument only shows that a more fine-grained decompositional analysis is needed and that an explicit reference to frames is not required. For example, a dependency relation between the actor and the theme of a writing event could be defined as given in (13-a) which can be generalized to (13-b).

(13) a. $\forall exy\ \textsc{write}(e) \wedge \textsc{theme}(e, y) \wedge \textsc{actor}(e, x) \rightarrow \textsc{author}(x, y)$
b. $\forall exy\ \textsc{write}(e) \wedge \textsc{theme}(e, y) \wedge \textsc{actor}(e, x) \rightarrow \textsc{dependent}(x, y)$

The important point to note, however, is that in the decompositional analysis in (13-b) this information is defined globally, i.e. in the model, and is therefore not directly available in an information state. Hence, a separate mechanism for accessing this global information in a discourse is needed. To put it differently: dependencies arise in discourse usually through the interpretation of verbs and

[11] See sections 4.7 and 4.10 below for details on how this link is established in our frame theory.

[12] The information state is also extended in the interpretation of the first sentence. But since the set of three students is newly introduced, there are no dependencies yet.

their arguments so that the author relation is not stored in an information state and such dependencies possibly need to be retrieved later on in a discourse without accessing the author relation. Using frames, dependency information is made available in an information state.

3 Formal frame theory

Consider the frame in Figure 3, which will be used as our running example in this section.

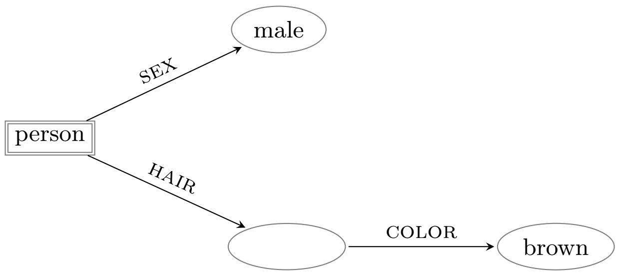

Fig. 3. Frame for 'male person with brown hair'

The frame in Figure 3 is a frame for 'male person with brown hair'. It is a frame for an entity typed as a person that is assigned values for three attributes: HAIR, COLOR and SEX. The SEX-attribute takes the value 'male', one out of two possible sex values. The value of the attribute HAIR takes the hair of the person as value.[13] This value is further specified by the attribute COLOR whose values are of type **color**.[14] In this case the particular value is 'brown'.

A frame is a description of a potential referent (Löbner, 2013, p.306). This referent is the value of a distinguished node, called the *referential node*. This node is marked with a double line circle. In Figure 3 the referent node is the node of type **person**.[15]

One well-known way of defining frames is as typed attribute-value structures. Let $\langle Sort, Attr \rangle$ be a signature s.t. *Sort* is a set of sort symbols and *Attr* is a set of attribute symbols or labels. Given Σ, a typed attribute-value structure is defined as follows.

[13] The hair of a person is an atom in the individual domain with the single hairs as material parts.

[14] In natural language an attribute and its target sort are often referred to by the same term. For discussion of the relation and the difference between attribute concepts and concepts for their values see Petersen (2007, 162ff.) and Löbner (2011, p.30-34).

[15] Not all frames have a root node and even if a frame has a root node it is not necessary the referential node. Frames in which the root node is the referential node are called *sortal frames*. Within the current paper we restrict ourselves to sortal frames (see Petersen, 2007, for details on non-sortal frames).

Definition 5 (Typed attribute-value structure) *A typed (or sorted) attribute-value structure of signature* $\langle Sort, Attr\rangle$ *is a triple* $\langle Q, \{S_s\}_{s\in Sort}, \{A_a\}_{a\in Attr}\rangle$ *where* Q *is a non-empty set of objects; for each* $a \in Attr$ A_a *is a binary relation on* Q *that is a partial function and for each* $s \in Sort$, S_s *is a unary relation on* Q*, i.e. a subset of* Q.

According to this definition, frames are relational structures, and hence first-order models, which are based on a language (signature) that is restricted to one-place and two-place relation symbols with the latter being required to be interpreted by functional binary relations. Additional constraints are imposed on the components of a model as well as on the relation between those components: (i) the domain is structured by imposing an order-sorted hierarchy on the set of unary relations, (ii) the binary relations get structured by imposing a set of operations on them and (iii) there is an interaction in form of operations between the two layers, i.e. the unary and binary relations are linked to each other in a particular way. In the next three subsections we will discuss and formally define these constraints in detail. We will begin by defining frame signatures.

3.1 Frame signatures

Definition 6 (Frame-signature of sort s**)** *A frame signature* Σ *is a quadruple* $\langle \mathcal{S}, \mathcal{P}, \mathcal{A}, s\rangle$ *s.t.*

(1) $\mathcal{S} = \langle Sort, \leq\rangle$ *is a sort hierarchy;*
(2) A $Sort^*$*-indexed family of sets of predicate symbols* $(\mathcal{P}_w)_{w\in Sort^*}$. *For* $p \in \mathcal{P}_w$ *one writes* $p : s_1 \ldots s_n$ *where* $w = s_1 \ldots s_n$.
(3) For every sort $s \in Sort$ *there is a unary predicate symbol* s *with argument sort* $\top$.
(4) A $Sort^* \times Sort$*-indexed family of sets of function symbols* $(\mathcal{A}_{w,s})_{w\in Sort^*, s\in Sort}$. *For* $\text{ATTR} \in \mathcal{A}_{w,s}$ *one writes* $\text{ATTR} : s_1 \ldots s_n \to s$ *where* $w = s_1 \ldots s_n$. w *is called the arity sort and* s *the coarity or target sort of* ATTR. *If* $w = s_1$, s_1 *is called the source sort of* ATTR.
(5) $s \in Sort$ *determines the sort of the signature.*

In Definition 6, $\mathcal{P}$ and $\mathcal{A}$ are pairwise disjoint. Sort predicates are assigned the argument sort $\top$ because one wants to be able to say for each object in a model whether it is of a sort s or not. $\top$ represents the most general sort to which every object in a model belongs. Therefore, it will be called 'object'.

Definition 6 of a frame signature is the most general one because it allows both for n-ary attributes and n-ary predicates. Both kinds have been used in frame theory (see Löbner, 2017; Kallmeyer and Osswald, 2013, for examples). The elements of $\mathcal{A}$ can also be taken as functional relations. We will use both perspectives interchangeably. In the sequel we only consider frame-signatures where $\mathcal{P}$ is restricted to sort predicates and all elements of $\mathcal{A}$ have arity 1, i.e. are unary function symbols. We will therefore identify *Sort* and $\mathcal{P}$ so that a frame signature is a triple $\langle \mathcal{S}, \mathcal{A}, s\rangle$.

3.2 The sort hierarchy

Attributes are typed (or sorted). They have both a source sort and a target sort. The most general sort, subsuming all other sorts and being true of all elements in the global model, is the sort **object** or $\top$. Possible subsorts include **individual**, **event** and **time** (point). These sorts, in turn, allow for finer distinctions by having subsorts as well. For example, the sort **event** has a subsort **moving events** with subsorts like **going**, **running** and **swimming**. The sort **individual** has a subsort **physical object** with subsorts like **person** and **animal**. Hence, the domain is not only sorted by a set of (possibly disjoint) sorts (flat ordering) but there is a genuine form of subtyping, yielding an order-sorted domain. Let us next make this idea formally precise. The following minimal constraints are imposed on the set *SORT* of sorts.

(i) *SORT* is partially ordered by $\leq$, i.e. $\leq$ is a reflexive, transitive and antisymmetric ordering. If $s \leq s'$ the sort s is said to be a subsort of s'.
(ii) There is both a bottom element, denoted by $\bot$, and a top element, denoted by $\top$, in *SORT*: $\{\top, \bot\} \subseteq SORT$; one has $\bot \leq s \leq \top$ for all $s \in SORT$.[16]
(iii) $\leq$ induces a meet operation $\sqcap$: $s \sqcap s' := \sup\{s'' \| s'' \leq s \textit{ and } s'' \leq s'\}$. $s \sqcap s'$ is the greatest common subsort of s and s'.

From the above three constraints it follows that the sort hierarchy is at least a meet-semilattice with a bottom and a top element. It is possible to close *SORT* under $\sqcup$ (the least common supersort) and ^{c} (complement), turning *SORT* into a lattice. We leave the exact structure of *SORT* open because this is an empirical rather than a purely theoretical or logical question.[17]

Two useful notions are those of the down-set and the up-set of a sort.

(14) a. $\textit{down-set}(s) := \{s' \mid s' \leq s\}$.
b. $\textit{up-set}(s) := \{s' \mid s \leq s'\}$.
c. $\textit{up-set}^*(s) := \{s' \mid s \leq s'\} \cup \{\bot\}$.

The notions of *down-set* and *up-set* are extended to sets of sorts in the expected way.

3.3 The attribute hierarchy

Attributes can be combined via sequencing, yielding chains of attributes. An example is HAIR and COLOR, which gives HAIR$\bullet$COLOR. This operation is partial. It is defined only if the target sort of the first attribute is a subsort of the source sort of the second attribute. As will be shown below in section 3.6, this is the only

[16] The requirement that $\bot$ be an element of *Sort* only holds for the global signature on which possible worlds are based.

[17] Though the sort hierarchy need only be a meet-semilattice, the set of terms of the language will always form a boolean algebra, i.e. if t_s and $t_{s'}$ are terms, then $t_s \sqcap t_{s'}$, $t_s \sqcup t_{s'}$ and $\neg t_s$ are terms too so that the set of terms is closed under conjunction, disjunction and negation (see e.g. Smolka, 1988, for details).

operation used in constructing the frame hierarchy. However, in the application of frames in semantic processing additional operations on attributes are needed as well. Dependency relations are defined in terms of chains in event frames connecting two objects participating in an event. Besides the use of sequencing, these chains are defined using the inverse operation $\otimes$. Finally, in talking about frames we will make use of boolean combinations of (chains of) attributes. Hence, the operations $\cap$ (intersection), $\cup$ (union) and $^{-}$ (complement) will be included too.

3.4 Operations between sorts and attributes: modes and projections

There are two types of operations which link the level of sortal information with the level of relational information, depending on the direction which the operation takes. First, we need to be able to express the fact that an object of a particular sort is in the domain or the range of an attribute or a chain of attributes. In a dynamic setting, such operations mapping static sortal information to dynamic relational information are called *modes* (see e.g. van Benthem, 1996, for details). Modes are directly linked to the relation between an object and a frame associated with this object. Recall that according to the frame hypothesis information about chains of attributes in a frame of sort s, say **person**, becomes available as soon as a comprehender gets the information during semantic processing that an object introduced is of sort s. For example, an object of sort **person** is in the domain of the chain of attributes HAIR • COLOR and SEX. Going in the other direction, one has *projections* (see again van Benthem, 1996, for details). These operations map (chains of) attributes to sorts. Suppose a comprehender has got the information that an object is of sort **person**. Using a mode, he knows which chains of attributes are associated with a frame of this sort. Applying a projection operation to one of this chains he gets its target sort, i.e. the target sort of the last attribute in this chain. For example, in a frame of sort **person** applying the projection function to the chain HAIR • COLOR yields the sort **color**.

When the constraints on the set of attributes and on the interaction between sorts and attributes are taken together, the structure associated with a frame is similar to that of a Peirce algebra.

3.5 Peirce algebras

Peirce algebras are two-sorted algebras with both set terms and relation terms so that sets and relations co-exist (see De Rijke, 1995, for an extensive discussion). In addition to the usual operations on sets and relations, there are operations which model the interaction between the two domains (for more details see De Rijke, 1995, p.228). We follow De Rijke (1995) and use two-sorted Peirce frames over a set D.[18]

[18] We use a slightly different definition than the one given in de Rijke (1994). Note that the notion of the term 'frame' used in 'Peirce frame' is *not* related to the notion of a frame used in our frame theory.

Definition 7 (The language $\mathcal{L}$ of a two-sorted Peirce frame) *Let $\mathcal{S}$ and $\mathcal{A}$ be as above. The two-sorted language $\mathcal{L}(\Delta, \otimes, \bullet, do, \cup, \cap, \uparrow, \downarrow, \mathcal{S}, \mathcal{A})$ is generated by the following rules:*

$$\phi ::= s \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid do(\pi)$$
$$\pi ::= 0 \mid 1 \mid \text{ATTR} \mid \pi \mid \pi_1 \cap \pi_2 \mid \pi_1 \cup \pi_2 \mid \otimes \pi \mid \pi_1 \bullet \pi_2 \mid \uparrow \phi \mid \downarrow \phi$$

The first kind of formulas are set formulas which are interpreted as sets, whereas the second kind are relation formulas which are interpreted as relations. Next, two-sorted Peirce frames based on a set D are defined. $D = \{D_s\}_{s \in Sort}$ is the union of finite domains D_s based on the sort hierarchy $\mathcal{S}$. In particular, one has: (i) $D = (D_s)_{s \in Sort}$ s.t. $(D_s)_{s \in Sort}$ is a *Sort*-indexed family of sets; (ii) $D_\top = D$ and $D_\bot = \emptyset$ and (iii) $D_s \subseteq D_{s'}$ if $s \leq s'$.

Definition 8 (Two-sorted Peirce frame based on a set D) *Let D be a set as defined above and $Re = D \times D$. A two-sorted Peirce frame over D is a tuple $\mathfrak{F} = \langle D, Re, I, R, C, F, F', P\rangle$ s.t. $I \subseteq Re$, $R \subseteq Re^2$, $C \subseteq Re^3$, $F \subseteq Re \times D$, $F' \subseteq Re \times D$, $P \subseteq D \times Re$ and*

$$I = \{\langle d, d'\rangle \in Re \mid d = d'\}.$$
$$R = \{(\langle d_1, d_2\rangle, \langle e_1, e_2\rangle) \in Re^2 \mid d_1 = e_2 \wedge d_2 = e_1\}.$$
$$C = \{(\langle d_1, e_1\rangle, \langle d_2, e_2\rangle, \langle d_3, e_3\rangle) \in Re^3 \mid d_1 = d_2 \wedge e_1 = e_3, e_2 = d_3\}.$$
$$F = \{(\langle d_1, e_1\rangle, d_2) \in Re \times D \mid d_1 = d_2\}.$$
$$F' = \{(\langle d_1, e_1\rangle, e_2) \in Re \times D \mid e_1 = e_2\}.$$
$$P = \{(d_1, \langle d_2, e_2\rangle) \in D \times Re \mid d_1 = d_2\}.$$

A model based on two-sorted Peirce frames is a structure $\mathcal{M} = \langle \mathfrak{F}, \{F_{\text{attr}}\}_{\text{ATTR} \in \mathcal{A}}, \{S_s\}_{s \in Sort}\rangle$ s.t. $\mathfrak{F}$ is a two-sorted Peirce frame. $\{F_{\text{attr}}\}_{\text{ATTR} \in \mathcal{A}}$ is a set of unary functions: $\{F_{\text{ATTR}} : D_{s_1} \to D_s \mid \text{ATTR} : s_1 \to s \in \mathcal{A}\}$. $\{S_s\}_{s \in Sort}$ is a set of unary relations $S_s = D_s \subseteq D$, i.e. sort predicates s are interpreted by the corresponding carrier set. Below the interesting satisfaction clauses are given. x_s is an element $d \in D$ and x_r is of the form $(d, d') \in Re$.

(15)
a. $\mathcal{M}, x_r \models \Delta$ iff $x_r \in I$.
b. $\mathcal{M}, x_r \models \otimes\pi$ iff $\exists y_r(Rx_r y_r \wedge y_r \models \pi)$.
c. $\mathcal{M}, x_r \models \pi \bullet \pi'$ iff $\exists y_r \exists z_r(Cx_r y_r z_r \wedge y_r \models \pi \wedge z_r \models \pi')$.
d. $\mathcal{M}, x_s \models do(\pi)$ iff $\exists y_r(Px_s y_r \wedge y_r \models \pi)$.
e. $\mathcal{M}, x_r \models\downarrow \phi$ iff $\exists y_s(Fx_r y_s \wedge y_s \models \phi)$.
f. $\mathcal{M}, x_r \models\uparrow \phi$ iff $\exists y_s(F'x_r y_s \wedge y_s \models \phi)$.

3.6 Constructing frames: the frame hierarchy

We begin by informally discussing a frame hierarchy. Each frame is of a particular sort which is given by the sort at its referential node. Frames of a particular sort form an ordered frame hierarchy. Minimal frames only express sortal information. They therefore are based on a language (signature) only of sort symbols and no attributes. If the sort of the frame is given by s, the signature Σ consists of s together with all sort symbols s' s.t. s' is a supersort of s: $s \leq s'$. The introduction of these additional sort symbols is necessary because whenever an object is of sort s it is also of sort s' for s' any supersort of s. Non-minimal frames of the same sort are construed from a minimal frame by successively adding attributes together with the required (upward-closed) sets of sort symbols to the language. Such an extension is possible only if the sort of the minimal frame is a subsort of the domain of the attributes which are added to the language. Consider again a frame of sort **person**. In a first step one adds the attributes HAIR, SEX or both. The result are frames in which maximal chains have length 1. At the next level one gets frames with maximal chains of length 2. An example is the person-frame in Figure 3 above. Iterating this process, one gets frames with maximal chains of an arbitrary finite length. Minimal frames will be said to have depth 0. Frames with maximal chains of length 1 have depth 1 and, in the general case, frames whose maximal chains have length n are of depth n. On this view a frame consists of a set of chains which are closed under prefixes s.t. all maximal chains have a common 'starting point': the root of the frame. Let us make these ideas formally precise. We begin by defining the signature for a minimal frame of a particular sort, i.e. a frame of depth 0 of that sort.

Definition 9 (Signature for a frame of depth 0.) *A signature for a frame of sort s having depth 0 is a triple $\Sigma_0 = \langle \mathcal{S}, \emptyset, s \rangle$ with $\mathcal{S} = \langle \textit{up-set}(s), \leq \rangle$.*

According to Definition 9, a signature for a minimal non-relational frame consists of a sort together with its supersorts but no attributes. As an example let's take our running example of a person frame. For the signature, one has: $\Sigma = \langle \mathcal{S}, \emptyset, \textbf{person} \rangle$ with $\mathcal{S} = \langle \textit{up-set}(\textbf{person}), \leq \rangle$.

The extension of a signature Σ_n of depth n by an attribute ATTR is defined as follows.

Definition 10 (Extension of a signature Σ_n by attribute ATTR)
Let a signature $\Sigma_n = \langle \mathcal{S}, \mathcal{A}, s \rangle$ be given. Let S_n be the set of target sorts of the attributes in $\mathcal{A}$ that were introduced at level n and let source-sort(ATTR) $\in$ up-set(S_n) and target-sort(ATTR) $= s'$. The ATTR-extension of Σ_n, $\Sigma_n +$ ATTR, is constructed as follows.

(i) $\textit{Sort}_{n+1} = \textit{Sort}_n \cup \textit{up-set}(s')$.
(ii) $\mathcal{A}_{n+1} = \mathcal{A}_n \cup \{\text{ATTR}\}$.
(iii) $\leq_{n+1}$ *extends* $\leq_n$.

$\Sigma_n +$ ATTR is defined only if the source sort of ATTR is the supersort of the target sort of an attribute defined at the previous level, i.e. at depth n. Consider the HAIR-extension of a minimal person-frame as an example. For the signature, one has: $\Sigma = \langle \mathcal{S}, \{\text{HAIR}\}, \textbf{person} \rangle$ with $\mathcal{S} = \langle \textit{up-set}(\textbf{person}) \cup \textit{up-set}(\textbf{hair}), \leq \rangle$.

The set of extensions Σ^ω of a signature Σ of sort s is inductively defined as follows.

(16) a. $\Sigma_0 \in \Sigma^\omega$.
b. If $\Sigma_n \in \Sigma^\omega$ and $\Sigma_n + \textsc{attr}$ is defined, then $\Sigma_n + \textsc{attr} \in \Sigma^\omega$.
c. Nothing else is in Σ^ω.

Finally, the notion of attribute extension is generalized to that of an extension as follows.

Definition 11 (Extension of a signature.) *Σ' is an extension of Σ iff Σ' and Σ are of the same sort and if there is a sequence of attribute extensions $\Sigma_1 \ldots \Sigma_n$ s.t. $\Sigma = \Sigma_1$ and $\Sigma' = \Sigma_n$.*

To each frame signature Σ_n there corresponds a particular set of axioms AX_{Σ_n}. For example, for a person-frame of depth 0, only axiom (17-a) applies, which has existential import. For a transition to a frame of this sort having depth 1, one has in addition axioms like that in (17-b). (17-c) is an example of an axiom for a person-frame of depth 2.[19]

(17) a. $\exists x.person(x)$.
b. $\forall x(person(x) \rightarrow \exists y(\textsc{hair}(x) = y \wedge hair(y)))$.
c. $\forall x(person(x) \rightarrow \exists y(\textsc{color}[\textsc{hair}(x)] = y \wedge color(y)))$.

We are now ready to define the notion of a frame.

Definition 12 (Frame definition) *Let $\Sigma_n = \langle\langle Sort, \leq\rangle, \mathcal{A}, s\rangle$ be a frame signature with corresponding axiom set AX_{Σ_n}. A frame based on Σ_n and AX_{Σ_n} is a minimal model $\mathcal{M} = \langle \mathfrak{F}, \{F_{\textsc{attr}}\}_{\textsc{attr} \in \mathcal{A}}, \{S_s\}_{s \in Sort}\rangle$ satisfying all axioms in AX_{Σ_n}.*

For example, a minimal frame of sort **person** has a universe with a single object which is a person. For the extension $\Sigma_0 + \textsc{hair}$, a minimal model (frame) has a universe with a person and his or her hair as its only elements.

Axioms like those in (17) embody two different kinds of information. Information about chains of attributes and information about relations between sorts. For the person-frame in Figure 3, one gets the set of chains in (18).

(18) $\{\textsc{hair}, \textsc{hair} \bullet \textsc{color}, \textsc{sex}\}$.

Abstracting from the relational component, the above axioms induce relations between sorts.

(19) a. $x : person$.
b. $x : person \rightarrow y : hair$.
c. $x : person \rightarrow y : hair \wedge z : sex$.

(19-c), for example, says that an object of sort **person** is related to objects of sort **hair** and **sex**. As noted by Fernando (2016, p.26), such relations between

[19] The axioms are formulated in first-order logic due to the familiarity of the latter. Formulated in terms of sort and relation formulas one gets for (17-a) and (17-b): (i) *person* and (ii) $person \rightarrow do(\textsc{hair} \cap \uparrow hair)$.

sorts can be taken as partial definitions of complex sorts from more basic ones. The interpretations of the constants in a Peirce frame are based on such relations. For example, if d is of sort **person**, d' of sort **hair** and d'' of sort **color**, one has $Cx_ry_rz_r$ with $x_r = (d, d'')$, $y_r = (d, d')$ and $z_r = (d', d'')$.

The above definition of frames has the advantage that frames can easily be related to possible worlds if the latter are taken as first-order models. A frame simply is a particular minimal submodel of a possible world. Possible worlds are based on a global signature $\Sigma_g = \langle\langle Sort_g, \leq\rangle, \mathcal{A}_g\rangle$.[20] It is not required that Σ_g is a frame signature though this is the case for those subsignatures of Σ_g that are used in the definition of frames. In the context of model theory the relation between frames and possible worlds can be made formally precise in the following way. We start by defining the necessary notions from model theory: (i) subsignature Σ' of a signature Σ, (ii) reduct of a model and (iii) submodel of a model.

Definition 13 (Subsignature) *Let $\Sigma = \langle\langle Sort, \leq\rangle, \mathcal{A}\rangle$ be a signature. A signature $\Sigma' = \langle\langle Sort', \leq'\rangle, \mathcal{A}'\rangle$ is a subsignature of Σ iff $Sort' \subseteq Sort$, $\mathcal{A}' \subseteq \mathcal{A}$, the arity function of Σ' is the restriction of the arity function of Σ to the domain of Σ' and $\leq'$ is the restriction of $\leq$ to $Sort'$.*

Definition 14 (Reduct of a model) *Let Σ' and Σ be two signatures s.t. $\Sigma' \subseteq \Sigma = \langle\langle Sort, \leq\rangle, \mathcal{A}\rangle$. Consider a model $\mathcal{M} = \langle D, \{F_a\}_{a\in\mathcal{A}}, \{S_s\}_{s\in Sort}\rangle$ with $D = (D_s)_{s\in Sort}$ based on Σ. The reduct $\mathcal{M} \upharpoonright \Sigma' := \langle D, \{F_a\}_{a\in\mathcal{A}_{\Sigma'}}, \{S_s\}_{s\in Sort_{\Sigma'}}\rangle$ is the reduct of $\mathcal{M}$ on Σ'. $F_{a_{\Sigma'}}$ and $S_{s_{\Sigma'}}$ are the restrictions of the interpretations of the F_a and S_s to Σ'.*

The reduct of a model $\mathcal{M}$ is the transition between signatures. In this transition the domain D of $\mathcal{M}$ remains constant, i.e. no elements of D are removed. Next, the notion of a submodel is defined.

Definition 15 (Submodel of a model) *Let $\mathcal{M} = \langle D, \{F_a\}_{a\in\mathcal{A}}, \{S_s\}_{s\in Sort}\rangle$ be a model. A submodel $\mathcal{N}$ of $\mathcal{M}$ is a model s.t. (i) the domain $D' = (D'_s)_{s\in Sort}$ of $\mathcal{N}$ is a subset of D: $D' \subseteq D$, (ii) for each $s \in Sort : D'_s = D_s \cap D'$, i.e. D'_s is the restriction to D' of the corresponding carrier set D_s and (iii) each n-ary F'_a is the restriction to D' of the corresponding n-ary F_a of $\mathcal{M}$, i.e. $F'_a = F_a \upharpoonright (D')^n$.*

Given a possible world $\mathcal{M} = \langle D, \{F_a\}_{a\in\mathcal{A}_g}, \{S_s\}_{s\in Sort_g}\rangle$, a frame based on the frame signature $\Sigma_s = \langle S, A, s\rangle$ and a set of axioms AX_{Σ_s} is constructed in the following way. In a first step one forms the reduct $\mathcal{N}$ of $\mathcal{M}$ to $\Sigma = \langle\mathcal{S}, \mathcal{A}\rangle$:

(20) $\quad \mathcal{M} \upharpoonright \Sigma = \langle D, \{F_a\}_{a\in\mathcal{A}_\Sigma}, \{S_s\}_{s\in Sort_\Sigma}\rangle.$

In this step, the universe of $\mathcal{M}$ remains unchanged. Let $\underline{\mathcal{N}}$ be the set of submodels of $\mathcal{M} \upharpoonright \Sigma$. The domain of the elements of this set is, by definition, a subset of

[20] For the sake of simplicity it is assumed that in this global signature the set of n-ary relations $\mathcal{P}$ coincides with $Sort_g$.

the domain of $\mathcal{M} \restriction \Sigma$. Frames of sort s based on Σ_s are particular elements of $\underline{\mathcal{N}}$.

Definition 16 (Frame of sort s based on Σ_s) *A frame of sort s based on the signature Σ_s is a minimal model $\mathcal{N} = \langle D_{\mathcal{N}}, \{F_a\}_{a \in \mathcal{A}_{g_\Sigma}}, \{S_s\}_{s \in Sort_{g_\Sigma}} \rangle$ in the set $\underline{\mathcal{N}}$ which (i) is based on a Peirce-frame and (ii) satisfies the axioms in AX_{Σ_s}.*

According to Definition 16, a frame is a minimal submodel of a possible world which is based on a Peirce-frame and satisfies the axioms given for the signature on which it is based.

4 Applying frames

4.1 Frames in type theory

We will use type theory to show how frames can be applied in semantic processing. In particular, we will use a variant of Van Eijck's type-logical Incremental Dynamics. Incremental Dynamics is based on the notion of a stack, which intuitively can be taken as linear sequences of objects. The use of stacks makes it possible to dispense with variables as the tool for modelling reference to objects in discourse. Besides the fact that Incremental Dynamics does not face the problem of destructive assignments (see van Eijck, 2001b,a; Nouwen, 2003, for details) which plagues DPL and its cognates, this formalism was chosen because it incorporates a two-layered architecture which corresponds to the distinction between the phrasal and the lexical level. At the phrasal level n-place predicate expressions are interpreted relative to stack positions and not relative to objects (type e). For example, determiners are relations between unary relations on stack position (or indices). By contrast, lexical items like common nouns and n-place verbs are basically interpreted in the same way as in ordinary type-logic, i.e. as the dynamic variants of expressions of type $\langle e, t \rangle$ and $\langle e^n, t \rangle$, respectively. When they are lifted to the phrasal level, they express (dynamic variants of) relations between stack positions though the basic predicates get evaluated at the values of those stack positions. A further advantage of this separation is that the phrasal layer can remain intact independently of which objects are stored on the stack. At the phrasal level one still deals with stack positions only. The difference only shows up when it comes to the interface between the two layers at the level of the interpretation of common nouns and verbs as well as domain-extension operations. If frames are integrated into this formalism, the two-layered approach makes it possible to restrict the use of frames in the lexicon to the interpretation of common nouns and verbs. Lexical elements which directly operate at the phrasal level like determiners and items expressing boolean operations receive interpretations that do not involve frames.

One problem that the incorporation of frames as defined in the first part into type theory faces is that the domain of possible worlds is standardly taken to be a set whose elements are atomic without any structure and not as first-order models. The relation between a world and information about this world is defined in a different way. For example, in two-sorted type theory this relation is modeled by shifting n-ary predicate expressions to n+1-ary expressions with the additional argument being a world. Consider the following example. In extensional type theory 'walk' is an expression of type $\langle e, t\rangle$. In two-sorted type theory it is instead of type $\langle s, \langle e, t\rangle\rangle$ where s is the type of possible worlds. For w a variable of type s, 'walk(w)' is of type $\langle e, t\rangle$, just as in the extensional variant.

We will follow this way of modelling the type s of possible worlds. More precisely, worlds and frames are of types s and f, respectively, with separate domains D_s and D_f whose elements are structureless objects. The types s and f are subtypes of the type η. One has $s \sqsubseteq \eta$, $f \sqsubseteq \eta$ and $s \sqcap f = \bot$ ($\sqsubseteq$ is the subtype relation).[21] For the domains, one has $D_\eta = D_s \cup D_f$. This way of incorporating frames in type theory is possible because the structure associated with frames is defined on the unary and binary relations. Hence, it is necessary to capture this structure at the atomic level. Following two-sorted type theory, common nouns and verbs get an additional argument. However, this argument is of type η and not of type s. The relation between a possible world and a frame is defined using a non-logical constant IN of type $\langle f, s\rangle$ which maps a frame to the world to which it belongs.

4.2 Incremental dynamics

In Incremental Dynamics, defined in van Eijck (2001b,a) and extended to a plural setting in Nouwen (2003), a context is defined as an element of D^*, the set of all finite sequences of elements taken from the domain D of an underlying model. Formally, such sequences are defined as stacks, i.e. as functions from an initial segment $0, \ldots, n-1$ of the natural numbers to D. Hence, stacks are finite sets of pairs $\langle i, d_i\rangle$. Positions are counted from left to right, beginning with 0 and referred to by $c[i]$. If $|c|$ is the length of stack c, c[i] is defined only if $i < |c|$ and undefined if $i \geq |c|$. If defined, $c[i]$ returns the object stored at the i-th position of the stack. Hence, Incremental Dynamic can be seen as a dynamic semantic theory operating on stacks of objects.

The concatenation operation $^\frown$ on stacks is defined in (21). An example is given in (22).

(21) $c^\frown c' := c \cup \{\langle i + |c|, d\rangle \mid \langle i, d\rangle \in c'\}$.

(22)

(22) shows that the position (or label) associated with an object is dependent on context. For example, in the second stack d'_0 is at position 0 whereas in the

[21] Frames and possible worlds are disjoint types because the latter is interpreted relative to 'maximal' entities whereas the former is interpreted as 'non-maximal' or 'partial' entities relative to those 'maximal' entities. This distinction at the level of types is meant to capture the definition of frames as minimal submodels based on a proper signature of the global signature on which (maximal) models are based.

0	1	2	...	n
d_0	d_1	d_2	...	d_n

$\sqcap$

0	1	2	...	m
d'_0	d'_1	d'_2	...	d'_m

$=$

0	1	2	...	n	$n+1$	$n+2$	$n+3$	...	$n+1+m$
d_0	d_1	d_2	...	d_n	d'_0	d'_1	d'_2	...	d'_m

resulting stack it is at position $n+1$. If c' is a one-element stack $\langle 0, d\rangle$, $c^\sqcap c'$ amounts to pushing d to the top of c. In this case $c^\sqcap c'$ will be abbreviated to $c^\sqcap d$.

Let $[e]$ be the type of contexts and ι be the type of (context) indices.[22] We use $a :: \alpha$ for 'a is of type α'. T abbreviates $[e] \to ([e] \to t)$ the type of context transitions, or, more precisely, the type of characteristic functions of binary relations on the type of contexts, with variables ϕ, ψ. P and Q are variables of type $\iota \to T$, the type of indexed context transition. Below the definitions of the main operations are given.

(23) a. $\exists^0 := \lambda c.\lambda c'.\exists d(c' = c^\sqcap d) :: T$.
b. $(\phi; \psi)^0 := \lambda c.\lambda c'.\exists c''(c'' \in \phi(c) \wedge c' \in \psi(c'')) :: T \to (T \to T)$.
c. $\neg(\phi)^0 := \lambda c.\lambda c'.(c = c' \wedge \neg\exists c'' \in \phi(c)) :: T \to T$.
d. $a^0 := \lambda P.\lambda Q.\lambda c.(\exists; P|c|; Q|c|)c :: (\iota \to T) \to ((\iota \to T) \to T))$.

$\exists$ defines the context extension operation. It takes a context and extends this context by an arbitrary element from the domain D. The operation ; is the composition of context transitions. It takes two context transitions and produces another context transition. Context negation $\neg$ requires the input context to be equal to the output context and imposes the additional requirement that it is not possible to execute ϕ in c. The definition of the determiner 'a' takes two indexed context transitions and maps them to a context transition.

Inspecting the above definitions, one notes that Incremental Dynamics is based on the following type lift compared to (extensional) Montague semantics.

(24) a. $t^* = T$
b. $e^* = \iota$
c. $(\alpha \to \beta) = \alpha^* \to \beta^*$

This type lift is also used in the interpretation of common nouns and verbs in the lexicon. Basically common nouns and intransitive verbs are of type $e \to t$

[22] The type system underlying Incremental Dynamic as well as its polymorphic character are defined in the appendix.

whereas transitive verbs are of type $e \to (e \to t)$. Applying the type lift in (24) one gets the following types at the dynamic level.

(25) a. $CN, ITV :: (e \to t)^* = e^* \to t^* = \iota \to T$
b. $TV :: (e \to (e \to t))^* = e^* \to (e \to t)^* = \iota \to (e^* \to t^*) = \iota \to (\iota \to T)$
c. $DTV :: (e \to (e \to (e \to t)))^* = (\iota \to (\iota \to (\iota \to T)))$

The lifts based on the above type assignments are given in (26).

(26) a. CN,ITV: $A^0 := \lambda j \lambda c \lambda c'.[c = c' \wedge A(c[j])]$
b. TV: $B^0 := \lambda j \lambda j' \lambda c \lambda c'.[c = c' \wedge B(c[j])(c[j'])]$
c. DTV: $C^0 := \lambda j \lambda j' \lambda j'' \lambda c \lambda c'.[c = c' \wedge C(c[j])(c[j'])(c[j''])]$

As already mentioned above, Incremental Dynamics can be seen as a two-layered architecture which is given by the distinction between stack positions, modeled by the type of context indices, and the values assigned to those stack positions, which are of type e in Van Eijck's original version. This two-layered architecture is visible in the type shift that underlies this dynamic system. The first layer is used at the phrasal level. For example, in the definition of 'a' the subject and the VP are interpreted as indexed context transitions of type $\iota \to T$. This corresponds to the lifted version of expressions of type $e \to t$. There are two operations which functions as a kind of interface between the two layers. First the context extension operation $\exists$ not only extends the input context by one position but it also assigns this position an element from D, i.e. an object of type e. This can be seen by looking at the type of $\{^\frown\}$ which is $[e] \to (e \to [e])$, i.e. it maps a context and an object to another context. Hence, the first interface operation is related to pushing an object on the stack. The second interface operation is given by the retrieval of an object from the stack. This operation is implicitly defined in the interpretation of verbs and common nouns. In their interpretation the basic forms of type $e^n \to t$ are used so that they are evaluated at the values of the stack positions which are used in the lifted versions at the phrasal level.

For our purpose, Incremental Dynamics must be modified in two respects: (i) frames must be introduced and (ii) plurality must be accounted for. This will be achieved by modifying the following three components: (a) information states, (b) the type of objects stored at a particular position in a stack and (iii) the interpretation of common nouns and verbs in the lexicon. These three changes will be discussed in the following three sections.

4.3 Intensionalizing Incremental Dynamics

Introducing frames has a distinctive epistemic flavour. Frames relate an object to other objects by chains of attributes. However, in many, if not most, cases a comprehender does not know the values at the end of those chains. Processing a discourse is one way of getting such information. Hence, processing a sentence should allow for a genuine update of an information state by eliminating some possibilities which are not compatible with the information conveyed by the sentence but which were taken to be possible by the comprehender before getting this information.[23] However, in an extensional framework, this is not possible since predicates are assigned fixed extensions. Therefore, a sentence is either true or false in an information state, provided it is defined in it. As an effect, either all possibilities survive if the sentence is true, or they are all rejected, if the sentence is false, yielding an empty, or absurd, information state. Consider e.g. 'Amanda amazed Brittany'. If a comprehender does not know whether Amanda amazed Brittany, there are possibilities in which this sentence is true and possibilities in which it is false, modelling his uncertainty about the truth of this sentence. Processing this sentence in an information state will therefore eliminate all those possibilities in which Amanda did not amaze Brittany.

One way of getting around this problem is to use possible worlds. On this perspective, a possibility consists of an assignment and a possible world, or, in the context of Incremental Dynamics, a stack and a possible world. This is the strategy that we will use below.

4.4 Information states in a frame theory

In this section we will define the notion of an information state in our frame theory. In section 2 we have already argued for one use of frames: they can store dependencies between sets of objects. In order for these dependencies to be used during semantic processing frames must be elements of information states. We will tackle the question of how this can be achieved from a broader perspective which analyzes what kinds of information have to be represented in such states.

Having both a stack (or variable assignment) and a world component is usually taken to reflect the distinction between discourse (stacks, variable assignments) and factual information.[24] Discourse information is mostly concerned with the question of what objects have been introduced into the discourse whereas factual information is information about the world which a comprehender has independently of any discourse or text he is processing. Where in this distinction does dependency information belong? According to the strategy discussed above in section 2.1, it belongs to discourse information because it is stored in sets of assignments. However, as also shown in section 2.1, dependency information is also factual information because it represents relations holding between objects in a particular possible world. What makes it discourse information is the fact that this information must be related to objects that have been introduced into the discourse, i.e. these objects must be elements of the first component of a possibility.

The above argument can be taken as showing that instead of only distinguishing between discourse and factual information, an information state should be built on two different parameters: (i) local (discourse) vs. global (world and situational knowledge), and (ii) (sortal) information about objects vs relational

[23] For the following, see also Dekker (1993, pp.197).

[24] See also Groenendijk et al. (1997) and Dekker (1993) for further discussion.

information about objects. In possible worlds the second distinction is reflected in distinguishing between a domain D (objects) and, say, a set of functions $\mathcal{F}$ and a set of relations $\mathcal{P}$. Stacks and partial variable assignments are the result of moving from the (global) domain D_w of a possible world w to the (local) domain D_g of a stack or a variable assignment : $D_g \subseteq D_w$ with D_g the set of objects one is talking about. The different combinations are listed in Table 5.

	objects	*relations between objects*
global	D_w	$\mathcal{P}, \mathcal{F}$
local	*domain of c or g* (D_g)	??

Table 5. Local and global information

This table leaves open the combination of *relations between objects* and *local*. In the context of a frame theory the thesis is that frames do the same as the domains of variable assignments in the combination *objects* and *local*. They move from the global elements given by $\mathcal{P}$ and $\mathcal{F}$ to their local variants relative to the objects in D_g. To be precise, locality at the relational level is achieved by using frames because they are submodels of possible worlds based on a subsignature of the global signature. Partiality of information at this level is encoded by associating with each object $d \in D_g$ a frame (of the appropriate sort) which only contains the information available about the object in the context of a particular given discourse. The completed tableau is shown below.

	objects	*relations between objects*
global	D_w	$\mathcal{P}, \mathcal{F}$
local	*domain of c or g* $(D_g \subseteq D_w)$	*frames*

Table 6. Local and global information with frames

How can the above distinction be incorporated into Incremental Dynamics? The global level gets represented by the second component, i.e. a possible world. For the local component, there are at least the following two options. The value assigned to a stack position is either a pair $\langle o, f \rangle$ consisting of an object o and an associated frame f, i.e. one has $root(f) = o$ or it is a frame. The second option is possible because given a frame f its corresponding object can always be retrieved by $root(f)$. Since discourse is standardly taken to be about objects and not about frames, we will choose the first option. However, in the applications below in section 4 we will sometimes use the second option because there is then no need to use projection functions. Note also that the second option results in a nice symmetry between local and global information: stack positions are frames (local) whereas the global component is represented by a possible world of which all the frames stored on the stack are a part (or a submodel, according to the way frames have been defined in the first part above).

As will be shown below in section 4.8, we will define dependency relations in terms of chains of attributes in frames. Hence, frames must be part of an information state since dependency relations arising during semantic processing belong to the local level. However, defining dependency in terms of frames, there is no need to use sets of stacks, as this is done in Nouwen (2003)'s analysis in Incremental Dynamics. Rather, it is sufficient to use the strategy for pluralization which is based on the shift from D to $\wp^+(D)$. On this strategy, a stack position is assigned a set of objects, i.e. an element of $\wp^+(D)$. In our frame-based setting this means that a stack position is assigned a set of pairs $\alpha = \langle o, f\rangle$. Such sets will be referred to by A and are called *discourse objects*. Equivalently, a stack position can be assigned a pair $\langle D', F'\rangle$ s.t. $D' \in \wp^+(D)$ and $F' \in \wp^+(F)$. What is the relation between D' and F'? For each $d \in D'$ there is a unique $f \in F'$ and for each $f \in F'$ there is a unique $d \in D'$. Hence, there is a bijection γ between D' and F'. γ can be defined using the function *root*: $\gamma(d) = f$ iff $root(f) = d$. Hence, each $d \in D'$ has a corresponding element in F' which is one of its corresponding frames. If γ is taken as a set of pairs $\langle o, f\rangle$, one gets the first representation. Another way of looking at the latter representation is as a variant of the second option where a stack position is basically assigned a frame. In a plural setting one gets a set of frames F'. Applying the *root* function to the elements of this set yields the set D'. In the sequel we will use both representations interchangeably.

We are now ready to define possibilities and information states in our frame theory.

Definition 17 (Possibility in a frame theory) *A possibility in a frame theory is a pair $\langle c, w\rangle$ s.t. c is a stack whose values are sets of pairs $\langle o, f\rangle$ with o an object and f a frame and $root(f) = o$ holds. w is a possible world and one has $IN(f) = w$.*

Definition 18 (Information state in a frame theory) *An information state in a frame theory is a set of possibilities.*

4.5 Capturing the sortal and relational structure of frames

The integration of frames makes it necessary to change the interpretation of common nouns and verbs. Recall that for frames there are two kinds of information expressed by set formulas and relation formulas, respectively. Hence, the non-decomposed common nouns and verbs in other approaches are replaced by combinations of elements of *Sort* and $\mathcal{A}$. Elements of *Sort* are of type $\langle o, \langle \eta, t\rangle\rangle$ and elements of $\mathcal{A}$ are of type $\langle o^2, \langle \eta, t\rangle\rangle$. We use the type η because sortal and relational information can be true both relative to a frame and a world.[25] Note

[25] See below (28) how this relation is captured.

that this extension with an additional argument is similar to the shift from an extensional type theory to an (intensional) two-sorted type theory. Compared to the frame theory in section 3, one has binary relations on $D \times D_\eta$ instead of unary relations P_s on D.[26] Similarly, the functional relations F_{ATTR} are replaced by relations on $D \times D \times D_\eta$ which are functional in their second argument[27], i.e. one has $\forall d, d', d'', f(F_{\text{ATTR}}(d)(d')(f) \wedge F_{\text{ATTR}}(d)(d'')(f) \rightarrow d' = d'')$.

Below we given the satisfaction clauses for sort and relation formulas.

(27)
1. $[\![s]\!](d)(f) = 1$ iff $\langle d, f\rangle \in P_s$.
2. $[\![\text{ATTR}]\!](d)(d')(f) = 1$ iff $\langle d, d', f\rangle \in F_{\text{ATTR}}$.
3. $[\![\pi_1 \bullet \pi_2]\!](d)(d')(f) = 1$ iff $\exists d'' : [\![\pi_1]\!](d)(d'')(f) = 1 \wedge [\![\pi_2]\!](d'')(d')(f) = 1$.
4. $[\![\pi_1 \cap \pi_2]\!](d)(d')(f) = 1$ iff $[\![\pi_1]\!](d)(d')(f) = 1 \wedge [\![\pi_2]\!](d)(d')(f) = 1$.
5. $[\![\pi_1 \cup \pi_2]\!](d)(d')(f) = 1$ iff $[\![\pi_1]\!](d)(d')(f) = 1 \vee [\![\pi_2]\!](d)(d')(f) = 1$.
6. $[\![\otimes\pi]\!](d)(d')(f) = 1$ iff $[\![\pi]\!](d')(d)(f) = 1$.
7. $[\![\uparrow \phi]\!](d)(d')(f) = 1$ iff $[\![\phi]\!](d')(f) = 1$.
8. $[\![\downarrow \phi]\!](d)(d')(f) = 1$ iff $[\![\phi]\!](d)(f) = 1$.
9. $[\![do(\pi)]\!](d)(f) = 1$ iff $\exists d'[\![\pi]\!](d)(d')(f) = 1$.
10. $[\![\Delta]\!](d)(d')(f) = 1$ iff $d = d'$.

The clauses for boolean connectives are standard. In the context of a chain of attributes sortal information is imposed using $\downarrow$ and $\uparrow$. For example, $\downarrow s \cap \pi \cap \uparrow s'$ holds of a triple $\langle d, d', f\rangle$ only if d is of sort s, i.e. $\langle d, f\rangle \in P_s$ and d' is of sort s', i.e. $\langle d', f\rangle \in P_{s'}$. To access sortal information at the root of a frame without reference to a chain of attributes, one uses $\Delta \cap \downarrow s$.

The relation between set and relation formulas at a frame and that at a world to which the frame belongs is captured by (28).

(28)
a. $\forall f \forall w[[\![\pi]\!](d)(d')(f) \wedge IN(f) = w \rightarrow [\![\pi]\!](d)(d')(w)]$.
b. $\forall f \forall w[[\![\phi]\!](d)(f) \wedge IN(f) = w \rightarrow [\![\phi]\!](d)(w)]$.

According to (28), truth of relational and sortal information relative to a frame implies the truth relative to the world to which the frame belongs.

The function *root* maps a frame to its root. The relation between a frame and the interpretation of the set of chains of attributes which are defined for it is captured by a function θ which maps a frame f to its corresponding set of relations.[28] Some examples are $[\![\Delta \cap \downarrow s]\!]$, $[\![\text{HAIR}]\!]$ and $[\![\text{HAIR} \bullet \text{COLOR}]\!]$. For our person frame in Figure 3, one gets $\theta(f) = \{[\![\Delta \cap \downarrow \textbf{person}]\!], [\![\text{HAIR}]\!], [\![\text{HAIR} \bullet \text{COLOR}]\!], [\![\text{SEX}]\!]\}$. $\theta(f)$ only models the chain structure of a frame f plus its sort at the root. Besides $[\![\Delta \cap \downarrow s]\!]$, $\theta(f)$ only contains relations based on relation formulas of the form $\text{ATTR}_1 \bullet \cdots \bullet \text{ATTR}_n$. Such formulas will be called ‘chain formulas’. What is missing is the sortal information, i.e. the target sort, at the end

[26] We assume that D is the same for all possible worlds, i.e. there is a unique domain of objects common to all possible worlds.

[27] We continue using P_s and F_{ATTR}. Furthermore, we always use a frame argument.

[28] To be precise, the value of θ corresponds to the axioms used in the first part. See e.g. (17) for our running person-frame.

of a chain formula. To this end, $\theta(f)$ is closed in the following way. For π a chain formula, $[\![\pi\cap\uparrow s]\!]$, for s the target sort of the last attribute in π, is also in $\theta(f)$. This extended set is referred to by θ_{sort}. For our person frame, one has $\theta_{sort}(f) = \theta(f) \cup \{[\![\textsc{sex}\cap\uparrow \textbf{male}]\!], [\![\textsc{hair}\cap\uparrow \textbf{hair}]\!], [\![\textsc{hair}\bullet\textsc{color}\cap\uparrow \textbf{brown}]\!]\}$. θ and θ_{sort} are closed under supersorts, that is if $[\![\Delta\cap\downarrow s]\!] \in \theta(f)$ or $[\![\pi\cap\uparrow s]\!] \in \theta_{sort}(f)$ for a frame f, then $[\![\Delta\cap\downarrow s']\!] \in \theta(f)$ and $[\![\pi\cap\uparrow s']\!] \in \theta_{sort}(f)$ for each $s' \in \textit{up-set}(s)$.[29] For event frames f, $\theta(f)$ is extended in the following way. If $[\![\textsc{attr}]\!]$ and $[\![\textsc{attr}']\!]$ are in $\theta(f)$, then $[\![\otimes\textsc{attr}\bullet\textsc{attr}']\!]$ and $[\![\otimes\textsc{attr}'\bullet\textsc{attr}]\!]$ are in $\theta(f)$ too. This extended set will be referred to by $\theta^*(f)$. $\theta^*(f)$ is required in the definition of event-related dependency since we need chains like $\otimes\textsc{actor}\bullet\textsc{theme}$.

Hence, the relational structure is not directly given by a frame f because frames are atomic objects. Rather the link between f and its associated relational structure is given by θ (θ_{sort}, θ^*). In section 4.8 below we will show how this relationship is used to define event-related dependencies in a discourse. Given an object d in the object component of a discourse object, its dependencies d' are accessed as follows. There must be an event e in the object component of a discourse object s.t. d and d are related in the associated frame f_e by a chain π with $R_\pi \in \theta^*(f_e)$. This line of reasoning is possible because the functions θ^* (θ_{sort}, θ) are always available since they are part of the global level, establishing the link between frames and sets of relations that are part of the frame hierarchy to which these frames belong. It is therefore not necessary to store $\theta^*(f_e)$ in a possibility together with f_e.

A further constraint between f and $\theta(f)$ is that if a relation R is in $\theta(f)$, then there are objects o and o' which satisfy R in f. Furthermore, if $[\![\pi]\!] \notin \theta(f)$, π cannot be true in f for any pairs of objects o and o'.

(29) a. $[\![\pi]\!] \in \theta(f)$ then $\exists o.\exists o'.[\![\pi]\!](o)(o')(f) = 1$.
b. $[\![\pi]\!] \notin \theta(f)$ then $[\![\pi]\!](o)(o')(f) = 0$ for any o and o'.

This constraint corresponds to the definition of frames as minimal submodels which are based on a particular signature. An alternative is to let such formulas be undefined relative to f if there is a frame f' extending f for which $[\![\pi]\!] \in \theta(f')$ holds and $[\![\pi]\!](o)(o')(f')$ for some objects o and o'. This move to a three-valued logic would be even more in line with the cognitive underpinnings of a frame theory. The value 'undefined' would then express lack of information in a particular information state.[30]

[29] θ should also be closed under 'superattributes'. For example, each shooting dead is a killing. Since we didn't incorporate this aspect of the set of attributes, it is not accounted for in the definition of θ.

[30] More generally, a frame theory has to address the question of partiality of information. Let us illustrate this for the case of an atomic attribute ATTR. At least the following cases have to be distinguished: (i) $[\![\textsc{attr}]\!] \in \theta(f)$, (ii) $[\![\textsc{attr}]\!] \notin \theta(f)$ but there is an extension f' of f with $\textsc{attr}]\!] \in \theta(f')$ (resolvable undefinedness, partiality) and (iii) $[\![\textsc{attr}]\!] \notin \theta(f)$ and there is no extension f' of f with $\textsc{attr}]\!] \in \theta(f')$ (unresolvable undefinedness). In order to account for these cases, a four-valued logic may be used. Using $\perp$ for resolvable undefinedness and $\surd$ for the unresolvable

The relation subframe $\sqsubseteq_F$ is defined as follows.

(30) $f \sqsubseteq_F f'$ iff $IN(f) = IN(f') \wedge root(f) = root(f') \wedge \theta(f) \subseteq \theta(f') \wedge \forall o.\forall o'.\forall R(R \in \theta(f) \wedge R(o)(o')(f) \rightarrow R(o)(o')(f'))$.

(30) requires that the frames belong to the same world, have the same root and that the chain-set of f be a subset of the corresponding set for f'. Finally, whenever a sort or relation formula is true in f it is also true in f'. In this definition we used the fact that sortal information can be expressed as relational information.

4.6 Domain extension operations

The domain extension update operation for indefinites in (31-a) is similar to the one in Incremental Dynamics. For each possibility in the input, a set of pairs $\alpha = \langle o, f\rangle$ consisting of an object and a frame is pushed on the stack. Let $A_w = \{\langle o, f_o\rangle \mid o \in D \wedge root(f) = o \wedge IN(f) = w\}$ for $w \in D_s$.

(31) a. $\exists^* = \lambda s.\lambda s'.\exists A(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge c' = c^\sqcap A \wedge A \in A_w)$

b. $\exists^*_{pro} = \lambda s.\lambda s'.\exists A.\exists i.\exists D'.\exists F_{D'}.\exists F'_{D'}(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge A = \langle D', F_{D'}\rangle \wedge c' = c^\sqcap A \wedge A \in A_w \wedge c[i] = A \wedge 0 \leq i < |c| \wedge \forall k(0 \leq k < |c| \wedge c[k] = \langle D', F'_{D'}\rangle \rightarrow k \leq i))$

For the interpretation of pronouns, a variant of (31-a) is used which requires A to be an element that is already on the stack. The condition $\forall k(0 \leq k < |c| \wedge c[k] = \langle D', F'_{D'}\rangle \rightarrow k \leq i)$ ensures that the frame information associated with the re-introduced object component D' of the discourse object A is current. This condition is necessary because re-introducing a discourse object always means adding additional information which gets stored in the frame component. Hence, though one still talks about D', there is now more information about its elements which leads to an update of the associated frame components. This will be discussed in more detail below.

So far we left open the question of which frame is introduced together with an object. The frame component is supposed to reflect the sortal and relational information a comprehender got about the object from semantically processing a sentence or a text. When an object is introduced a comprehender only gets sortal information about it. This sortal information is of the most general kind. In the present context this means that it is of sort 'object'. Hence, only a minimal frame providing most general sortal information is introduced. The only change that is

case, one gets (i) $[\![\text{ATTR}]\!](o)(o')(f) = 1$ if $[\![\text{ATTR}]\!] \in \theta(f) \wedge \langle o, o', f\rangle \in F_{\text{ATTR}}$; (ii) $[\![\text{ATTR}]\!](o)(o')(f) = 0$ if $[\![\text{ATTR}]\!] \in \theta(f) \wedge \langle o, o', f\rangle \notin F_{\text{ATTR}}$; (iii) $[\![\text{ATTR}]\!](o)(o')(f) = \perp$ if $[\![\text{ATTR}]\!] \notin \theta(f) \wedge \langle o, o', f\rangle \notin F_{\text{ATTR}} \wedge \exists f' : f \sqsubseteq_F f' \wedge [\![\text{ATTR}]\!] \in \theta(f)$ and (iv) $[\![\text{ATTR}]\!](o)(o')(f) = \surd$ if $[\![\text{ATTR}]\!] \notin \theta(f) \wedge \langle o, o', f\rangle \notin F_{\text{ATTR}} \wedge \nexists f' : f \sqsubseteq_F f' \wedge [\![\text{ATTR}]\!] \in \theta(f)$. This move would, of course, also make it necessary to rethink the relation between epistemic uncertainty and sets of possible worlds. We must leave this topic for another occasion. See also Muskens (2013) for a proposal of how to account for partiality in a frame theory.

required is related to the sets A_w: the condition $\theta(f) = \{\Delta \cap \downarrow o\}$ is added which requires the set of relations associated with with f to be both minimal and the most general one. $A_w = \{\langle d, f_d \rangle \mid d \in D \wedge \mathit{root}(f) = d \wedge IN(f) = w \wedge \theta(f) = \{\Delta \cap \downarrow o\}\}$ for $w \in D_s$.

This way of modeling the frame component may seem to be at odds with the motivation for introducing frames given above in section 2 where we claimed that common nouns and verbs introduce not only sortal information but also relational information associated with frames of the sort to which the object belongs. This objection does not take into account the following important distinction. The relational information associated with an object of a particular sort is about the chains of attributes and the source and target sorts of those attributes. This knowledge is given by the function θ which assigns to a frame a set of relations. This knowledge is independent of any particular discourse and belongs to the global level. In addition, this knowledge does not imply that a comprehender knows, for a particular object, the values at the end of those chains. Information about these values is in general local and provided in a discourse. The distinction between the frame component f based on bottom-up information and the information given by $\theta(f)$ becomes important in the definition of discourse dependency and pronoun resolution. Discourse dependency has to be defined in terms of information a comprehender has got from processing a text up to a given point since dependencies must have explicitly been introduced into a discourse in order to be available in cross-sentential anaphora. By contrast, the strategy of a comprehender in resolving a pronoun uses information contained in $\theta(f)$ and possible extensions of this set, i.e. it can be based on global information, see below sections 4.10 and 4.11 for details.

4.7 Adapting the lexicon to frames

We start with the interpretation of common nouns and verbs. Recall that a domain extension operation only introduces a most general frame of sort **object**. Additional information about this object is got by further bottom-up processing of atomic formulas, either by sortal information: it is a writing, it is a woman; or by relational information: Actor, theme or other binary relations.

The interpretation of verbs is based on a decompositional analysis in the tradition of event semantics. The decomposition predicates are either state or relation formulas which are interpreted relative to a frame and a single object (state formula) or a pair of objects (relation formula). Similar to DRT, an event is introduced which is not an argument of the verb. The interpretation of verbs is illustrated with 'write'. To ease readability, we use θ instead of θ_{sort}. The same holds for the interpretation of common nouns below.

(32) a. $\lambda j.\lambda i.\lambda s.\lambda s'.\exists c.\exists w.\exists c'.\exists e.\exists f.\exists d.\exists f_d.\exists d'.\exists f_{d'}(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge c' = c^\frown\{\langle e, f\rangle\} \wedge \mathit{root}(f) = e \wedge IN(f) = w \wedge c[i] = \{\langle d, f_d\rangle\} \wedge c[j] = \{d', f_{d'}\} \wedge [\![\mathit{write}]\!](e)(f) \wedge [\![\textsc{actor}]\!](e)(d)(f) \wedge [\![\textsc{theme}]\!](e)(d')(f) \wedge \theta(f) = \{[\![\Delta \cap \downarrow \mathit{write}]\!], [\![\textsc{actor}]\!], [\![\textsc{theme}]\!]\} \cup \mathit{up\text{-}set}_\Delta(\mathit{write}))$

b. $\mathit{up\text{-}set}_\Delta(s) := \{[\![\Delta \cap \downarrow s']\!] \mid s' \in \mathit{upset}(s)\}$

An event e and an associated frame f are introduced (since e is the root of f). It is tested whether e and f satisfy the sort and relation formulas relative to the arguments of the verb. $\theta(f) = \{[\![\Delta\cap \downarrow write]\!], [\![\text{ACTOR}]\!], [\![\text{THEME}]\!]\}$ ensures that f is minimal with respect to the information contained in the interpretation of 'write'.[31] The decomposition predicates are evaluated only w.r.t. the event frame, i.e. relative to the event component, and not relative to the corresponding event-related frames.[32] The reason is the following. First, truth of the relation formula relative to the event frame implies the truth of the corresponding relation formula in the event-related frame of the object bearing the attribute to the event. In particular, one has: $[\![\text{ATTR}]\!](e)(d)(f_e) \Rightarrow [\![\otimes\text{ATTR}]\!](d)(e)(f_{d_{\text{ATTR}}})$. Second, since discourse dependencies are defined at the level of event frames, there is no need to store the associated event-related frames.[33] The interpretation in (32) assumes that the arguments of the verb are singletons, i.e. of the form $\{\langle d, f_d\rangle\}$. For the subject position, this is warranted by our assumption to model collective readings in terms of groups, which are atoms. The VP is fed either with a group in subject position (collective reading) or the distributivity operator δ^* applies to it,[34] having the effect that the VP is interpreted relative to each element of the discourse object denoted by the subject. For the direct object, see below section 4.10.

The interpretation for other transitive verbs is strictly similar and the interpretation of intransitive and ditransitive verbs proceeds according to the same pattern.

The interpretation of a common noun of sort s is given in (33).

(33) $\lambda i.\lambda s.\lambda s'.\exists c.\exists w.\exists c'(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge |c'| = |c| \wedge c'[j] = c[j]\ for\ (0 \leq j < |c| \wedge j \neq i) \wedge \forall\langle d, f\rangle \in c[i].\exists f'([\![s]\!](d)(f') \wedge f \sqsubseteq_F f' \wedge \theta(f') = \theta(f) \cup up\text{-}set_\Delta(s)) \wedge c'[i] = \{\langle d, f'\rangle \mid \exists f : \langle d, f\rangle \in c[i] \wedge f \sqsubseteq_F f' \wedge \theta(f') = \theta(f) \cup up\text{-}set_\Delta(s) \wedge [\![s]\!](d)(f')\})$

The interpretation acts both as a test, i.e. it is eliminative, and as an update operation. The test is given by the conjunct $\forall\langle d, f\rangle \in c[i].\exists f'([\![s]\!](d)(f') \wedge f \sqsubseteq_F f' \wedge \theta(f') = \theta(f) \cup up\text{-}set_\Delta(s))$. For each pair $\langle d, f\rangle$ stored at position i there must be a frame f' extending f which satisfies the sortal information s provided by the common noun. The requirement that the value of θ for f' is the union

[31] In order to get all information associated with a writing event e in a discourse $\theta(f_e)$ should not be restricted to the information about thematic relations. Rather, this set should in addition include the extended chains for those relations got from information about the values of these relations. For example, for the theme one adds $R_{[\![\text{THEME}\bullet\pi]\!]}$, for π an element of $\theta(f_{d'})$, $\pi \neq \Delta\cap \downarrow s$ and d' the theme of e. We must leave to another occasion of how to build a comprehensive event frame based on the frames of (discourse) objects participating in it. This task consists in integrating a non-local update operation.

[32] For the notion of an event-related frame, see section 4.14

[33] Below in section 4.11 it will be shown that for the resolution of pronouns, event-related frames should be stored as well. Since this leads to a more complex interpretation of verbs, we postpone this more involved interpretation to that section.

[34] See below section 4.9 for the definition of δ^*.

of the value for f and the up-set of the newly introduced sort has the effect that f' is a minimal extension of f satisfying the constraints on the sortal and relational structure.[35] The expansive character is given by the fact that the frame component stored at the i-th position is not f but the minimal extension f'. Hence, if the input context stores at the i-th position only frames whose sole information is that it is of sort **object**, as this is the case after the domain update operation associated with 'a' or a bare numeral has been executed, the common noun 'student' extends each frame stored at position i with the sortal information that it is in addition a student. Since no new discourse objects are introduced but only a position is updated, $|c| = |c'|$ ensures that no new objects are introduced in the output stack whereas $c'[j] = c[j]$ *for* $(0 \leq j < |c| \wedge j \neq i)$ ensures that only the value at position i is updated.[36]

The way the meanings of common nouns are defined shows a more general aspect of our frame theory. The frame component f associated with an object is built incrementally. It is not fixed. Rather, it is updated if additional information about the object becomes available through bottom-up processing. The new component f' contains all the information contained in the old component plus the information provided by the common noun. Formally, this is achieved by requiring that the value of $\theta(f')$ be the union of $\theta(f)$ plus the set of relations associated with the common noun. It is not necessary to store $\theta(f')$ in addition to f' in the frame component of a discourse object since it is functional dependent on f' and it is part of the frame hierarchy and therefore part of the general knowledge of a comprehender.

This strategy can be generalized as follows. Bottom-up processing yields both sortal and relational information. To each set of such information there corresponds a particular frame f and a particular set of relations $\theta(f)$ $(\theta_{sort}(f)$, $\theta^*(f))$. The information got about an object are instantiations of the elements of $\theta(f)$ $(\theta_{sort}(f), \theta^*(f))$ relative to f and its root as first argument. Hence, f can be seen as a description of a partial instantiation of the concept modelled by the frame hierarchy to which f or $\theta(f)$ $(\theta_{sort}(f), \theta^*(f))$ belong. Since $\theta(f)$ $(\theta_{sort}(f)$, $\theta^*(f))$ is functional dependent on f, it is sufficient to store f together with its root. To get the object back to which the root is related in f by some element R of $\theta(f)$, one simply reverses the chain and calculates $\iota d.R(root(f)(d)(f)$. The overall approach is therefore modular. Knowledge about the frame hierarchy is general knowledge that is independent of discourse information. What has to be stored, therefore is the frame f' that corresponds to a particular set of relations in the frame hierarchy of the sort to which f belongs.

---

[35] Recall that chain formulas not in $\theta(f)$ for a frame f are always false in f.

[36] The interpretation of common nouns shows that a context is not a stack in the strict sense because in our theory an element that is not the top (last) element can be updated. The update operation is necessary because it reflects the fact that during processing a text additional information about objects can become available.

Before we turn to the interpretation of pronouns, the interpretation of bare numerals must be given. To this end we have to define the number of atoms in the value of the stack component in a possibility. Let $n \in \mathbb{N}$.[37]

(34) $n^* := \lambda i.\lambda s.\lambda s' \exists c.\exists w.s = s' \wedge s = \langle c, w\rangle \wedge |c[i]| = n.$

According to this definition, n is the number of pairs $\langle d, f_d\rangle$ stored at position i. n^* is defined only if $i < |c|$, i.e. if the position i is an element of the stack c of the possibility s. In the applications below this definedness condition is always satisfied because it is the length of the global input possibility of the transition function in which it occurs. For plural pronouns, we in addition need (35).

(35) $> n^* := \lambda i.\lambda s.\lambda s' \exists c.\exists w.s = s' \wedge s = \langle c, w\rangle \wedge |c[i]| > n.$

The interpretation of the bare numeral n is given in (36).

(36) $n := \lambda P.\lambda Q.\lambda s.(\exists^* \cdot n^*(|\pi^1(s)|) \cdot P(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$

We follow Nouwen (2003) and assume that the determiner ‘a’ has a built-in cardinality constraint.

(37) $a := \lambda P.\lambda Q.\lambda s(\exists \cdot 1(|\pi^1(s)|) \cdot P(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s.$

Finally, we turn to the interpretation of pronouns. We start with the interpretation of ‘it’.[38]

(38) $\lambda P.\lambda s(\exists_{pro} \cdot 1(|\pi^1(s)|) \cdot Q_{it}(|\pi^1(s)|) \cdot P(|\pi^1(s)|))s.$

Q_{it} is defined as follows.

(39) $Q_{it} = \lambda i.\lambda s.\lambda s' \exists c.\exists c'.\exists w.\exists d.\exists f_d.\exists f'(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge |c| = |c'| \wedge c[i] = \{\langle d, f_d\rangle\} \wedge f_d \sqsubseteq_F f' \wedge \theta(f') = \theta(f) \cup \{\text{ SEX }\} \wedge [\![\text{SEX}]\!](d)(\mathit{neutral})(f') \wedge c'[j] = c[j]\, \mathit{for}\, (0 \leq j < |c| \wedge j \neq i)).$

Similar to the interpretation of a common noun, ‘it’ functions both as a test and as an update operation. First, a new discourse object is pushed on the stack that is required to already have been pushed on the stack before. Next it is tested whether the cardinality of this discourse object is 1, i.e. a single pair $\langle d, f_d\rangle$. In the Q_{it} part it is tested whether there is a minimal extension f of f_d for which the value of the SEX-attribute is ‘neutral’. If this test succeeds, the frame component is updated with f, i.e. f_d is replaced by f. The value ‘neutral’ is meant only as a rough approximation. The exact specification of this value is

[37] Using an existential quantifier in the interpretation of bare numerals yields ‘at least n’ readings. One way to force an ‘exactly’ reading is to use a (local) maximality condition (see e.g. Van den Berg, 1996). However, this results in readings that often are too strong, witness ‘You pass the exam if you get 50 points’. An alternative is to model an ‘exactly’-reading in terms of a defeasible implicature. We leave this aspect of the interpretation of bare numerals to another occasion.

[38] π^1 and π^2 are projection functions. $\pi^1(s)$ is the stack (context) component of the possibility s whereas $\pi^2(s)$ is the world component.

an empirical question. Since Q_{it} does not introduce a new discourse object but only updates a position, $|c| = |c'|$ ensures that no new objects are introduced in the output stack. The assumption that at position i only a single pair is stored is warranted due to the previous cardinality test.

In contrast to 'it' 'they' and 'them' only impose a constraint on the cardinality of the discourse object. There is no condition on the frame component.

(40) $\lambda P.\lambda s.(\exists_{pro} \cdot > 1(|\pi^1(s)|) \cdot P(|\pi^1(s)|))s.$

Having defined a basic lexicon, we are ready to define in the next section the notion of dependency which is based on the information contained in the frame components of discourse objects.

4.8 Event-related dependency

The definition of dependency in Definition 4 above in section 2.1 is based on the notion of a plural information state. As a result, it is completely discourse-based. In contrast to this way of defining dependency, the basic definition of dependency in our frame theory is independent of notions that are directly related to discourse. Rather, discourse-dependency is a derived notion.

Consider again the example of the three students who each wrote an article. The structure corresponding to this sentence is given by the relation in (41), assuming that the students are stored in x and the articles in y.

(41) $R_{xy} = \{\langle s_1, a_1\rangle, \langle s_2, a_2\rangle, \langle s_3, a_3\rangle\}.$

This relation is closely related to the event of writing: there is a collection of events of sort 'write' which contains three atomic subevents of the same sort. Each of these atomic events is related to an element of (41) whereas the collection itself is related to the whole set of students and to the whole set of articles. The relation between the students and the articles is given by a chain in frames associated with the writing events: $\otimes$ACTOR$\bullet$THEME. Generalizing this example, we arrive at the following thesis.

(42) Discourse-dependencies are in effect dependencies involving an event.[39] Two objects are dependent on each other if they participate in a common event. Let us call this kind of dependency *event-related dependency.*

We begin by defining event-related dependency relative to an event and a chain in a frame associated with this event. Since frames are always related to a particular possible world, the definition must also be relativized to a world.

Definition 19 (Event-related dependency relative to an event and a chain) *The notion of two objects d and d' being dependent on each other relative to an event e, an associated frame f, a world w and a chain formula π, denoted by $dep_\pi(d, d', e, f, w)$, is defined as follows:*

[39] The notion of 'event' is understood in a broad sense including states which are denoted by stative verbs like 'own' and 'think' for example.

$$dep_\pi(d, d', e, f, w) \textit{ iff } root(f) = e \wedge IN(f) = w \wedge [\![\pi]\!](d)(d')(f).$$

According to this definition, two objects are dependent on each other relative to an event and a frame of this event in a world if they are connected by a chain in this frame. For example, in the case of (41) one has: $\pi = \otimes\text{ACTOR} \bullet \text{THEME}$. Let the three writing events be e_1, e_2 and e_3. One then has for example $dep_{\otimes\text{ACTOR}\bullet\text{THEME}}(s_i, a_i, e_i, f_{e_i}, w)$ because there is a frame f_{e_i} associated with e_i s.t. $f_{e_i}, s_i, a_i \models \otimes\text{ACTOR} \bullet \text{THEME}$.

Definition 19 can be generalized in two ways.

Definition 20 (Event-related dependency relative to an event) *$dep(d, d', e, f, w)$, i.e. d and d' are dependent on each other relative to e and f in w, is defined by*

$$dep(d, d', e, f, w) \textit{ iff } \exists R : R \in \theta^*(f) \wedge R = [\![\pi]\!] \wedge dep_\pi(d, d', e, f, w).$$

Definition 21 (Event-related dependency) *$dep(d, d', w)$, i.e. d and d' are dependent on each other in w, is defined by*

$$dep(d, d', w) \textit{ iff } \exists e.\exists f.dep(d, d', e, f, w).$$

Definition 20 requires for two objects d and d' to be dependent on each other that there be a chain in the frame f of e that connects the two objects. According to Definition 21, two objects are dependent on each other if there is an event and an associated frame in which they are related by some chain (always relative to a possible world).

The definition of event-related dependency and its generalizations given above are purely semantic in the sense that they do not refer to the information which is available to a comprehender in a particular discourse. Hence, they belong to the global level. However, such a link is required because a comprehender can only use dependencies between objects that have been introduced into a discourse. This was the problem of spurious dependencies discussed above in section 2.2. Consider also the following example taken from Heim (1990).

(43) a. Every man who has a wife sits next to her.
b. *Every married man sits next to her.

Only if the wife has explicitly been introduced in the information state is it felicitous to anaphorically refer to her, as in (43-a). By contrast, since in (43-b) the wife is not introduced into the discourse, anaphoric reference is not possible although her existence can be inferred from the fact that the man is married and despite the fact that this information is part of the frame associated with the man. These examples show that the objects related in a dependency must have been introduced into a discourse in order for the dependency to be felicitously accessed afterwards. It is not sufficient that the relation is defined in the model, i.e. at the global level. This is exactly similar to cases of anaphoric relations to objects. It is therefore necessary to define a notion of dependency which is related to objects that have been introduced into the discourse.

At the discourse level the dependency relation between two sets need not be based on a bijection between the two sets. Consider again one of our running examples 'Three students each wrote two articles'. After processing this sentence, there is a discourse object consisting of six articles plus the associated frames. Each of the three students is related to exactly two of these articles. This fact must be reflected in the definition of discourse event-related dependency because it is not possible to refer to one of the articles written by a student by a singular pronoun like 'it': 'Three students each wrote two articles. * They sent it to L&P.' Recall that a discourse object A is an element of a context c if (44) holds.

(44) $A \in c$ *iff* $\exists i : 0 \leq i < |c| \wedge c[i] = A$.

Definition 22 (Discourse event-related dependency)
dep$(\langle d, f\rangle, \langle D', F_{D'}\rangle, \langle E', F_{E'}\rangle, w, c)$ *is defined as follows:*

$$dep(\langle d, f\rangle, \langle D', F_{D'}\rangle, \langle E', F_{E'}\rangle, w, c) \text{ iff}$$

1. $\exists A, A', A'' \in c$;
2. $\langle d, f\rangle \in A, \langle D', F_{D'}\rangle \subseteq A', \langle E', F_{E'}\rangle \subseteq A''$;[40]
3. $IN(f) = w$, $\forall f_{d'} \in F_{D'} : IN(f_{d'}) = w$, $\forall f_{e'} \in F_{E'} : IN(f_{e'}) = w$;
4. $\exists n |E'| = n = |D'| \wedge D' = \{d'_1, \ldots, d'_n\} \wedge E' = \{e'_1, \ldots, e'_n\}$.
5. $dep(d, d'_i, e'_i, f_{e'_i}, w)$ *for* $1 \leq i \leq n$.
6. $\langle D', F_{D'}\rangle$ *and* $\langle E', F_{E'}\rangle$ *are maximal w.r.t. the above conditions: there are no* $\langle D'', F_{D''}\rangle \subseteq A'$ *and* $\langle E'', F_{E''}\rangle \subseteq A''$ *s.t.* $\exists m : m = |D''| = |E''| \wedge n < m \wedge D'' = \{d''_1, \ldots, d''_m\} \wedge E'' = \{e''_1, \ldots, e''_m\} \wedge dep(d, d''_i, e''_i, f_{e''_i}, w)$ *for* $1 \leq i \leq m$.

According to this definition, an object d with associated frame f bears a dependency to a set of objects D' (with associated frames $F_{D'}$) relative to the set of events E' (with associated frames $F_{E'}$) if all objects belong to the context c and d is event-related dependent to each $d' \in D'$ in an event-frame associated with an element in E' (modulo a maximality condition). The notion of discourse event-related dependency will be used in the definition of the distributivity operator below in 4.9. In the loop triggered by this operation the input stack is extended by a pair $\langle d, f\rangle$ over which the loop is executed and all pairs $\langle D', F_{D'}\rangle$ which are discourse dependent on it. This set is defined as follows.

(45) $dep(d, f, c, w) :=$
$\{\langle D, F_{D'}\rangle \mid \exists E'.\exists F_{E'}.dep(\langle d, f\rangle, \langle D', F_{D'}\rangle, \langle E', F_{E'}\rangle, w, c)\}$.

The definition of discourse dependency ultimately always relates atoms, i.e. elements of D. One may therefore object that it fails to capture dependencies involving proper sets as for example in collective readings like 'Three students wrote two articles together'. In this case one has two dependencies relating the set of three students to each of the two articles, respectively. We apply another

[40] Recall that we use $\langle D', F_{D'}\rangle$ interchangeably for a set of pairs $\langle d, f_d\rangle$. Hence, we use $\langle D', F_{D'}\rangle \subseteq A'$ though $\langle D', F_{D'}\rangle$ is strictly speaking a pair and not a set.

strategy. We follow Landman (2000) and assume that collective readings are modelled in terms of groups. For each non-empty subset D' of D there is a corresponding group $\uparrow D'$. For singletons one sets $\{d\} = \uparrow \{d\}$. For example, to the set $\{s_1, s_2, s_3\}$ of three students corresponds the group $\uparrow \{s_1, s_2, s_3\}$. Groups are atoms. Hence, the domain D of individuals is closed under group formation. The cardinality of a group is the cardinality of its underlying set. On a collective reading of 'Three students wrote two articles'. there are two events of writing s.t. in each associated event frame the event is related to the group of three students by the ACTOR-attribute and to the article by the THEME-attribute. Using groups for collective readings therefore has the effect that dependencies always relate atoms.

Let us point out some immediate consequences of our definition of dependency, in particular similarities and differences to the notion of discourse dependency in Definition 4 above in section 2.1. In contrast to discourse-dependency, event-related dependency applies to cases where the relation between two sets is given by the cartesian product.

(46) a. Every boy likes every girl in his class.
b. Three students together wrote an article.
c. John likes Mary.

In (46-a) each boy is related to each girl and vice versa. Hence, each boy is related to the whole set of boys so that there is no restriction which triggers a difference among the boys in relation to the girls. To see this, assume that the context set for 'boy' is the set of boys in some class. Assume furthermore that it is a very small class with just three boys and three girls. One then has: $b_i \rightarrow \{g_1, g_2, g_3\}$. Hence, there is a dependency between the boys and the girls. But according to Definition 4 above in 2.1, there is no dependency because $I|_{x=b_1}(y) = I|_{x=b_2}(y) = I|_{x=b_3}(y) = \{g_1, g_2, g_3\}$. Similarly, in the case of (46-b) the students as a whole are related to a single article (or its singleton set) so that there either is no difference w.r.t. to the set of articles assigned to a single student or the plural information state consists of a single assignment so that there can be no difference at all. A similar argument applies to (46-c). In this case too the plural information state consists of only one assignment so that there can be no discourse-dependency.[41] By contrast, applying event-related dependency to (46-a), there is a total of nine atomic subevents, each relating a boy to a particular girl. For (46-b), the group of three students is as a whole related by one writing event to the (collectively) written article. The case of (46-c) is similar. There is only one event of sort 'like' which relates John to Mary. Hence, event-related dependency is more general than discourse-dependency.

On the other hand, there are cases which fall under discourse-dependency but which are not instances of event-related dependency. One example are the 'spurious' dependencies from above.

[41] We consider only one plural information state which results when the sentence is interpreted with an empty input possibility.

(47) Two students$_1$ each collaborated with a girl$_2$. They$_2$ each collaborated with a professor$_3$.

The students and the girls are dependent on each other due to the event of collaboration introduced in the first sentence. Similarly, the girls are dependent on the professors due to the event of collaboration in the second sentence. From these two dependencies no dependency between the boys and the professors can be inferred. Any such dependency has to be introduced either relative to one of these two events of collaboration or by a third event introduced into the discourse. Hence, event-related dependency is more restrictive than discourse-dependency.

Having defined discourse dependency, we are finally ready to define a distribution operator at the VP-level which makes essential use of the notion of discourse event-related dependency.

4.9 Creating dependencies: distributivity at the VP-level

Following Nouwen (2003, pp.135), we define a distribution operator as a predicate modifier of type $\langle\langle\iota, T\rangle, \langle\iota, T\rangle\rangle$. This has the effect that its selective character is taken care of in the syntax. The index i contains the set of objects over the elements of which the distributive loop is executed. The scope P is interpreted w.r.t. each atomic value, the loop object, that is stored in i. In an application, i always stores the discourse object introduced by interpreting the subject NP.[42] The interpretation of the subject possibly introduces further discourse objects, yielding a context c. Inside the loop c is temporarily extended by the loop object together with its discourse dependencies, yielding $c' = c^\frown\{\langle d, f_d\rangle\}^\frown c_{dep(d,f_d,c,w)}$.[43] The VP is processed relative to the loop object stored at position $c'[|c|]$ and the possibility $\langle c', w\rangle$ i.e. the extended context c'. Processing of P relative to these two arguments must yield an output for any choice of the loop object. This is ensured by the first conjunct $\forall\langle d, f_d\rangle \in c[i] : \exists c'.c' = c^\frown\{\langle d, f_d\rangle\}^\frown c_{dep(d,f_d,c,w)} \wedge P(c'[|c|])(\langle c', w\rangle) \neq \emptyset$ which requires that for each element stored in $c[i]$ the VP can successfully be processed, i.e. there is an output with this element as input. The input and output may differ at most w.r.t. objects which are introduced in the scope P, i.e. the VP. This is ensured as follows. All extensions c'' to the input context c' triggered by processing the VP relative to c' (and $c'[|c|]$) have a common length n: $\forall\langle d, f_d\rangle \in c[i].\forall c'.\forall c''(c' = c^\frown\{\langle d, f_d\rangle\}^\frown c_{dep(d,f_d,c,w)} \wedge c'' \in \{c \mid \langle c'^\frown c''\rangle \in P(c'[|c|])(\langle c', w\rangle)\} \rightarrow |c''| = n)$. The global extension c^* that is added to c is required to have this length too: $|c^*| = n$. When processing P separately for each element $\langle d, f_d\rangle \in c[i]$ we want in the global output context one value that results from a successful execution for each $\langle d, f_d\rangle$. We achieve this by using a

[42] This is possible because similar to indefinite NPs pronouns are interpreted as domain extension operations, i.e. they push a discourse object on the stack.

[43] The idea of temporarily extending the stack with the dependencies of the loop object is inspired by the proposal in Nouwen (2007), which also relates this idea to the way dependencies are analyzed in DRT.

function F defined as follows. F is a function with domain $c[i]$, i.e. the elements of the discourse objects over which the loop is executed. The value of F for a pair $\langle d, f_d \rangle$ is an element of the extensions that result if the VP is processed relative to $c'[|c|]$ and the possibility $\langle c', w \rangle$ with $c' = c^\sqcap \{\langle d, f_d \rangle\}^\sqcap c_{dep(d,f_d,c,w)}$ and c the input context to the loop: $F(\langle d, f_d \rangle) \in \{c''' \mid \langle c'^\sqcap c''', w \rangle \in P(c'[|c|])(\langle c', w \rangle)\}$. Hence, F has the effect of assigning to each pair $\langle d, f_d \rangle$ relative to which the loop is executed a possible extension that this execution triggers to the input context when the VP is processed. The global extension c^* to the input c of the whole loop, which has the same length as the contexts in the range of F, is construed in terms of this range. The value of position j of c^* is the union of the values at position j for the contexts in the range of F: $c^*[j] = \{\langle d', f_{d'} \rangle \mid \exists c. \exists \langle d, f_d \rangle . F(\langle d, f_d \rangle) = c \wedge \langle d', f_{d'} \rangle \in c[j]\} \, for 0 \leq j < n$. The definition of the distributivity operator is given in (48). As usual, there is the definedness condition that the index i be an element of the domain of s, i.e. $i < |s|$.

(48) $\delta^* := \lambda P. \lambda i. \lambda s. \lambda s'. \exists c. \exists c^*. \exists w. \exists n. \exists F(s = \langle c, w \rangle \wedge s' = \langle c^\sqcap c^*, w \rangle \wedge$
$\forall \langle d, f_d \rangle \in c[i] : (\exists c'. c' = c^\sqcap \{\langle d, f_d \rangle\}^\sqcap c_{dep(d,f_d,c,w)} \wedge$
$P(c'[|c|])(\langle c', w \rangle) \neq \emptyset) \wedge$
$\forall \langle d, f_d \rangle \in c[i]. \forall c'. \forall c''(c' = c^\sqcap \{\langle d, f_d \rangle\}^\sqcap c_{dep(d,f_d,c,w)} \wedge$
$c'' \in \{c''' \mid \langle c'^\sqcap c''', w \rangle \in P(c'[|c|])(\langle c', w \rangle)\} \rightarrow |c''| = n) \wedge |c^*| = n \wedge$
$c^*[j] = \{\langle d', f_{d'} \rangle \mid \exists c. \exists \langle d, f_d \rangle . F(\langle d, f_d \rangle) = c \wedge \langle d', f_{d'} \rangle \in c[j]\} \, for 0 \leq j <$
$n)$

The stack $c_{dep(d,f_d,c,w)}$ is defined as follows. Recall that the set of dependencies is given by (49).

(49) $dep(d, f, c, w) :=$
$\{\langle D', F_{D'} \rangle \mid \exists E'. \exists F_{E'}. dep(\langle d, f \rangle, \langle D', F_{D'} \rangle, \langle E', F_{E'} \rangle, w, c)\}$.

Recall furthermore that each $\langle D', F_{D'} \rangle$ uniquely corresponds to a set of pairs $\langle d, f_d \rangle$. Extending a stack by a set of sets of pairs $\underline{A}$ is defined as a repeated basic extension operation. Instead of $c^\sqcap \underline{A}$ we write $c^\sqcap c_{\underline{A}}$, i.e. we take $\underline{A}$ as a stack.

Definition 23 (Domain extension by a set $\underline{A}$) *Given a stack c and a set of sets of pairs $\underline{A} = \{A_0, \ldots, A_n\}$, the extension of c by $\underline{A}$, denoted $c^\sqcap \underline{A}$, is defined by*
$c^\sqcap \underline{A} = ((\ldots((c^\sqcap A_0)^\sqcap A_1) \ldots)^\sqcap A_n)$.

In the next two sections we will show our frame theory at work. We start by deriving our two running examples of the three students each either writing one or two articles and sending them collectively or distributively to L&P. These derivations assume knowledge of the actual truth conditions of sentences in order to exclude possible but false readings. In the next section but one we show how pronoun resolution based on bottom-up information can proceed in our theory by relying solely on constraints imposed on the sortal and relational structures of the frame components of discourse objects.

4.10 Frame theory at work: accounting for dependent and independent readings in discourse

In this section we illustrate the workings of our frame theory by means of some derivations.[44]

(50) Three students each wrote a paper.

1. *a* $\rightsquigarrow \lambda P.\lambda Q.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot P(|\pi^1(s')|) \cdot Q(|\pi^1(s')|))s'$
2. *paper* $\rightsquigarrow PAPER^*$
3. *a paper* $\rightsquigarrow \lambda Q.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot Q(|\pi^1(s')|))s'$
4. *wrote* $\rightsquigarrow WROTE^*$
5. *wrote a paper* $\rightsquigarrow \lambda j.(\lambda Q.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot Q(|\pi^1(s')|))s')(\lambda i.(WROTE^* i)j) = \lambda j.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot WROTE^*(|\pi^1(s')|)j)s')$
6. *each* $\rightsquigarrow \delta^*$
7. *each wrote a paper* $\rightsquigarrow \delta^*(\lambda j.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot WROTE^*(|\pi^1(s')|)j)s')$
8. *three* $\rightsquigarrow \lambda P.\lambda Q.\lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|) \cdot P(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
9. *students* $\rightsquigarrow STUDENT^*$
10. *three students* $\rightsquigarrow \lambda Q.\lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|) \cdot STUDENT^*(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
11. *Three students each wrote a paper* $\rightsquigarrow \lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|) \cdot STUDENT^*(|\pi^1(s)|) \cdot [\delta^*(\lambda j.\lambda s'.(\exists^* \cdot 1^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot WROTE^*(|\pi^1(s')|)j)s')](|\pi^1(s)|))s$

Assume that (11) is interpreted in an empty possibility $s = \langle\langle\rangle, w\rangle$, i.e. the context c is empty: $|c| = 0$. The interpretation of 'three students' introduces a set of three students, say $\{s_1, s_2, s_3\}$, yielding the possibility $s' = \langle c', w\rangle$ with the object component of c' below (To ease readability, only the object component and not the frame component is shown.)

0
$\{s_1, s_2, s_3\}$

Table 7. Object component of output context after processing the subject

Next, the distributive loop is executed. Its first two inputs are the VP in (51) below and the index at which the object over which the loop is executed is stored. This is $|c| = 0$, i.e. the length of the global input which is 0. Hence, one has $\delta^*(V)(0)$.

[44] Note that the derivation proceeds in exactly the same way as in Nouwen's account (Nouwen, 2003). This is the case because there are no differences at the phrasal level. The mode of combination is always functional application except for the combination of a transitive verb with the (object) NP: $\lambda u.(NP(\lambda v.(TV v)u))$.

(51) $\lambda j.\lambda s'.(\exists^* \cdot 1^*(|s'|) \cdot \mathit{PAPER}^*(|\pi^1(s')|) \cdot \mathit{WROTE}^*(|(|\pi^1(s')|)|)j)s')$.

The input to the loop is $s' = \langle c', w\rangle$, in which c' has length 1. Inside the loop the context c' is temporarily extended by (the singleton of) the loop object $\langle s_i, f_{s_i}\rangle$ and the discourse dependencies of this object $dep_{s_i, f_{s_i}, c', w}$. Since the set of three students has just been introduced and the subject NP is not modified, say 'students who read a book', there are so far no such dependencies. Hence, $dep_{s_i, f_{s_i}, c', w}$ is empty. The object component of the extended stack therefore consists of the set of three students and (the singleton set of) the single student over which the loop is executed. The VP is interpreted relative to position 1, the loop object, and the possibility $s'' = \langle c'', w\rangle$ with c'' the extended context. The interpretation of 'an article' puts (non-deterministically) a_i on the stack, yielding the output context $\{s_1, s_2, s_3\}^\sqcap\{s_i\}^\sqcap\{a_i\}$ which is the input context for the interpretation of the verb.[45] Its two arguments interpreted w.r.t. to position 1 (subject) and position 2 (object). First, an event e_{write_i} is introduced at position 3. The decomposition of 'write' requires (52).

(52) $[\![\textbf{write}]\!](e_{write_i})(f_{e_{write_i}}) \wedge [\![\textsc{actor}]\!](e_{write_i})(s_i)(f_{e_{write_i}}) \wedge$
$[\![\textsc{theme}]\!](e_{write_i})(a_i)(f_{e_{write_i}})$.

The 'write'-frame $f_{e_{write_i}}$ links the event e_{write_i} to student s_i by the attribute ACTOR and to article a_i by the attribute THEME. The object component of the output context is $\{s_1, s_2, s_3\}^\sqcap\{s_i\}^\sqcap\{a_i\}^\sqcap\{e_{write_i}\}$. The loop is executed three times, one execution for each of the three students. In the output context the temporary extensions are discarded and the values at positions are cumulated. This yields the object component of the global output context below.

0	1	2
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$

Table 8. Object component of output context after processing 'Three students wrote an article'

At the end of processing the first sentence each of the three 'write'-frames stores a dependency between a student and the article he wrote. These dependencies can be accessed in subsequent discourse, as will be shown when we analyze the second sentence to which we turn next.

(53) They sent it to L&P.
1. $\mathit{it} \rightsquigarrow \lambda P.\lambda s'.(\exists^*_{pro} \cdot 1^*(|\pi^1(s')|) \cdot Q_{it}(|\pi^1(s')|) \cdot P(|\pi^1(s')|))s'$
2. $\mathit{sent\text{-}to\text{-}L\&P} \rightsquigarrow \mathit{SENT\text{-}TO\text{-}L\&P}$
3. $\mathit{sent\ it\ to\ L\&P} \rightsquigarrow \lambda j.(\lambda P.\lambda s'.(\exists^*_{pro} \cdot 1^*(|\pi^1(s')|) \cdot Q_{it}(|\pi^1(s')|) \cdot P(|\pi^1(s')|))s')(\lambda i.(\mathit{SENT}^* i)j) = \lambda j.\lambda s'.(\exists^*_{pro} \cdot 1^*(|\pi^1(s')|) \cdot Q_{it}(|\pi^1(s')|) \cdot \mathit{SENT}^*(|\pi^1(s')|)j)s'$

[45] As is usual, we discuss a successful processing.

4. $each \rightsquigarrow \delta^*$
5. $each\ sent\ it\ to\ L\ \&P \rightsquigarrow \delta^*(\lambda j.\lambda s'.(\exists^*_{pro} \cdot 1^*(|\pi^1(s')|) \cdot Q_{it}(|\pi^1(s')|) \cdot SENT^*(|\pi^1(s')|)j)s')$
6. $they \rightsquigarrow \lambda Q.\lambda s.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
7. $They\ each\ sent\ it\ to\ L\&P \rightsquigarrow \lambda s.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s)|) \cdot [\delta^*(\lambda j.\lambda s'.(\exists^*_{pro} \cdot 1^*(|\pi^1(s')|) \cdot Q_{it}(|\pi^1(s')|) \cdot SENT^*(|\pi^1(s')|)j)s')](|\pi^1(s)|))s$

If (53) is interpreted in the context of (50), the input possibility s is the output possibility of (50). The object component of the context c is repeated below for convenience.

0	1	2
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$

Table 9. Input for 'They sent it to L&P'

'They' introduces a new object on the stack that is already an element of its domain. The two possible choices are the set of three students at position 0 and the set of three articles at position 1.[46] The distribution is over this newly introduced object at position $3 = |c|$. Let this possibility be c' with corresponding possibility $s' = \langle c', w\rangle$.[47]

0	1	2	3
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$	$\{s_1, s_2, s_3\} \vee \{a_1, a_2, a_3\}$

Table 10. Object component of output context after processing 'They'

Hence, one has $\delta^*(V)(3)(s')$ for $V = (54)$.

(54) $\lambda j.\lambda s'.(\exists^*_{pro} \cdot 1^*(|s'|) \cdot SENT^*(|s'|)j)s'$

The distribution operator extends c' by $\{s_i\}$ (if 'they' is assigned the set of three students or $\{a_i\}$ (if 'they' is assigned the set of three articles) and the discourse event-related dependencies. In this case this set is non-empty and contains the paper a_i written by s_i or the student s_i who wrote paper a_i. This yields the two contexts c'' below.
At position $|c''| = 4$ the object is stored relative to which the loop is executed. The interpretation of 'it' introduces an object on the stack at position 6 which is required to be already on the stack. The cardinality condition excludes all

[46] For clarity we left out the third possibility, i.e. the set of three writing events.
[47] Note that s' are in effect two different possibilities.

0	1	2	3	4	5
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$	$\{s_1, s_2, s_3\}$	$\{s_i\}$	$\{a_i\}$
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$	$\{a_1, a_2, a_3\}$	$\{a_i\}$	$\{s_i\}$

Table 11. Temporary extended context inside the loop

antecedents whose value is not a singleton set. This leaves only the objects at positions 4 or 5. Note that it is at this point that the extension of the input stack not only by the loop object s_i but also by the object dependent on it is required. Without adding the dependencies only s_i could be assigned to 'it', resulting in a false interpretation. Hence, the temporary extension of the context makes it possible that a pronoun can get the (or a) value that yields correct truth conditions for a dependent reading though this extension is no longer part of the global output context of the loop. Note that the dependencies are determined solely on the basis of information that is available in the (local) current possibility since only the frame components of discourse objects on the stack are considered in determining the dependencies.

Since a_i did not sent s_i, the second possibility in Table 11 is discarded. The event of sending is introduced at position 7. The object component of the output context after processing the second sentence is shown in the table below.

0	1	2	3	4	5
$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$	$\{s_1, s_2, s_3\}$	$\{a_1, a_2, a_3\}$	$\{e_{send_1}, e_{send_2}, e_{send_3}\}$

Table 12. Object component of the output context after processing the second sentence

Next we turn to our second running example.

(55) Three students each wrote two papers. They each sent them to L&P.

Before we tackle this derivation we have to be more explicit about how we model the collective/distributive distinction. As already said above in section 4.8, we follow Landman (2000) and model collective readings in terms of groups. Having groups as a separate domain of objects, basic predicates nominal as well as verbal, are semantically interpreted as sets of atoms. They differ w.r.t. what kinds of atoms they take in their extension. For example, $[\![student]\!]$ and $[\![walk]\!]$ contain only individual atoms whereas $[\![write]\!]$ and $[\![send]\!]$ contain both individual atoms and group atoms. Collective predication is singular predication to groups and distributive predication is plural predication to sets of atoms, either individuals, events or groups. Hence, there are two modes of predication: singular and plural predication. Singular predication applies to atoms in the domain, that is either individuals, like John or Mary, or groups, like the group of three students or the group of articles written by a particular student. By contrast, plural predication applies to plural objects, i.e. sums of atoms, and is defined in terms of singular predication to the atomic parts of that sum. As a result, a verb is always evaluated w.r.t. atomic values, i.e. discourse objects that are singletons. We assume that the type shifting operation from a set to its corresponding group is always available. Hence, for the value at position i in context c, both the set and the corresponding group are available. If the object component of a discourse object in subject position is not a singleton, δ applies to the VP.

In accordance with the above considerations, the interpretation of verbs in the lexicon in (32) above assumes that its arguments are atomic because only a single pair $\langle e, f_e \rangle$ is introduced. Using δ at the VP-level, accounts for the atomicity of the external argument. One either distributes over its elements, triggered by δ^*, or, on a collective reading, one gets a group, and hence again a singleton. In order to account for the atomicity of the internal argument the interpretation of transitive verbs in the lexicon has to be changed. One way of doing this is to define a distribution operator which operators at the level of these verbs. This operator has to have access to the event parameter because there needs to be distribution over it. However, in our theory the event parameter is not an argument of the verb. We have refrained from introducing an explicit event argument due to the well-known complications this engenders in relation to quantification (see Champollion, 2015, and the references cited therein for a recent and state-of-the-art discussion). Furthermore, for our data a distribution at the level of verbs is not strictly necessary. Recall that in the second sentence of (55) 'it' is infelicitous so that a distribution over the articles written by one of the students is not possible. Instead, a partitive construction has to be used: 'They each sent each of them to L&P'. Hence, a distribution over the internal argument is possible only in the first sentence. The difference between the two readings for this argument is that one gets at the object component the set of three groups $\{\uparrow \{ a_1, a_2\}, \uparrow \{ a_3, a_4\}, \uparrow \{ a_5, a_6\}\}$ and the set $\{a_1, \ldots, a_6\}$. The dependencies for student s_1 are $\{\uparrow \{ a_1, a_2\}\}$ and $\{a_1, a_2\}$, respectively. As will be shown below, this does not affect the anaphoric potential. One way of getting a distributive reading of the internal argument in our theory is to integrate the distribution into the interpretation of a verb. Let $A_{E_w} = \{\langle e, f_e \rangle \,|\, root(f) = e \wedge IN(f) = w \wedge \theta(f) = \{\Delta \cap \downarrow \mathbf{write}, [\![\textsc{actor}]\!], [\![\textsc{theme}]\!]\}\}$; $\exists! :=$ there is a unique.

(56) $\lambda j.\lambda i.\lambda s.\lambda s'.\exists c.\exists w.\exists c'.\exists A.\exists d.\exists f_d(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge c[i] = \{\langle d, f_d\rangle\} \wedge c' = c^\sqcap A \wedge \forall \langle e, f_e\rangle \in A([\![write]\!](e)(f_e) \wedge [\![\textsc{actor}]\!](e)(d)(f_e)) \wedge \forall \langle e, f_e\rangle \in A.\exists! \langle d, f_d\rangle \in c[j] : [\![\textsc{theme}]\!](e)(d)(f_e) \wedge \forall \langle d, f_d\rangle \in c[j].\exists! \langle e, f_e\rangle \in A : [\![\textsc{theme}]\!](e)(d)(f_e))$

In (56) a plural discourse object A of sort **event** is introduced. For each element of A, it is required that there is exactly one element of the internal argument stored in $c[j]$ to which it is related by the theme attribute in its corresponding event frame. Conversely, for each element in $c[j]$ there is exactly one element in A that bears the THEME-attribute to it in the frame component of the event. Now we are ready for analyzing the example. First the derivations.

1. *two* $\rightsquigarrow \lambda P.\lambda Q.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|) \cdot P(|\pi^1(s')|) \cdot Q(|\pi^1(s')|))s'$
2. *paper* $\rightsquigarrow PAPER^*$
3. *a paper* $\rightsquigarrow \lambda Q.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|) \cdot PAPER^*(|\pi^1(s')|) \cdot Q(|\pi^1(s')|))s'$
4. *wrote* $\rightsquigarrow WROTE^*$
5. *wrote two papers* $\rightsquigarrow \lambda j.(\lambda Q.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|) \cdot PAPER^*(\pi^1(s'))\cdot Q(|\pi^1(s')|))s')(\lambda i.(WROTE^* i)j) = \lambda j.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|) \cdot PAPER^*(\pi^1(s'))\cdot WROTE^*(|\pi^1(s')|)j)s')$
6. *each* $\rightsquigarrow \delta^*$
7. *each wrote two papers* $\rightsquigarrow \delta^*(\lambda j.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|) \cdot PAPER^*(\pi^1(s'))\cdot WROTE^*(|\pi^1(s')|)j)s')$
8. *three* $\rightsquigarrow \lambda P.\lambda Q.\lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|) \cdot P(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
9. *students* $\rightsquigarrow STUDENT^*$
10. *three students* $\rightsquigarrow \lambda Q.\lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|) \cdot STUDENT^*(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
11. *Three students each wrote two papers* $\rightsquigarrow \lambda s.(\exists^* \cdot 3^*(|\pi^1(s)|)\cdot STUDENT^*(|\pi^1(s)|)\cdot [\delta^*(\lambda j.\lambda s'.(\exists^* \cdot 2^*(|\pi^1(s')|)\cdot PAPER^*(\pi^1(s'))\cdot WROTE^*(|\pi^1(s')|)j)s')](|\pi^1(s)|))s$

(57) They sent it to L&P.

1. *them* $\rightsquigarrow \lambda P.\lambda s'.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s')|) \cdot P(|\pi^1(s')|))s'$
2. *sent-to-L&P* $\rightsquigarrow SENT\text{-}TO\text{-}L\&P$
3. *sent them to L&P* $\rightsquigarrow \lambda j.(\lambda P.\lambda s'.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s')|)\cdot P(|\pi^1(s')|))s')(\lambda i.(SENT^* i)j) = \lambda j.\lambda s'.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s')|)\cdot SENT^*(|\pi^1(s')|)j)s'$
4. *each* $\rightsquigarrow \delta^*$
5. *each sent them to L &P* $\rightsquigarrow \delta^*(\lambda j.\lambda s'.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s')|)\cdot SENT^*(|\pi^1(s')|)j)s')$
6. *they* $\rightsquigarrow \lambda Q.\lambda s.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s)|) \cdot Q(|\pi^1(s)|))s$
7. *They each sent it to L&P* $\rightsquigarrow \lambda s.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s)|)\cdot[\delta^*(\lambda j.\lambda s'.(\exists^*_{pro} \cdot > 1^*(|\pi^1(s')|) \cdot SENT^*(|\pi^1(s')|)j)s')](|\pi^1(s)|))s$

We begin with the reading of the first sentence in which the internal argument gets a collective (group) reading. First, a set of three students is introduced. After temporarily extending the context with one of the students (similar to example (50) there are yet no dependencies for the student) inside the loop a group of two articles and a writing event are introduced. The object component of the output context is given in Table 13.

Next we turn to the second sentence. Recall the two interesting readings one gets if this sentence gets a distributive reading at the VP-level. The pronoun 'them' can refer either to the set of articles written by the student over whom the loop is executed or it can refer to the set of articles cumulatively written

0	1	2
$\{s_1, s_2, s_3\}$	$\{\uparrow \{a_1, a_2\}, \uparrow \{a_3, a_4\}, \uparrow \{a_5, a_6\}\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$

Table 13. Output after processing 'Three students wrote two articles'

by the three students. The problem of previous approaches is that inside a distributive loop only the dependent set but not the independent or global set is available. Let's see how our frame theory handles this sentence. Processing 'they' introduces either the set of three students, the set of (groups of) articles or the set of events. The latter two are eventually discarded when the verb is processed. The distributive loop temporarily extends the input stack with student s_i and its discourse dependencies. On a collective reading of the direct object in the first sentence each student s_i is related to the (atomic) group of two articles written by him: $\uparrow \{a_{i_1}, a_{i_2}\}$. Hence, there is a single event e_i of writing s.t. one has $[\![\otimes\text{ACTOR} \bullet \text{THEME}]\!](s_i)(\uparrow \{a_{i_1}, a_{i_2}\})(f_{e_i})$. There therefore is a discourse dependency between s_i and the group $\uparrow \{a_{i_1}, a_{i_2}\}$ of cardinality two. The pronoun 'them' can therefore take its value from context in Table 14.

0	1	2	3	4
$\{s_1, s_2, s_3\}$	$\{\uparrow \{a_1, a_2\}, \uparrow \{a_3, a_4\}, \uparrow \{a_5, a_6\}\}$	$\{e_{write_1}, e_{write_2}, e_{write_3}\}$	$\{s_i\}$	$\uparrow \{a_{i_1}, a_{i_2}\}$

Table 14. Object component of the output context after temporarily extending the input stack

Since 'them' requires a cardinality greater 1, possible objects are $\{s_1, s_2, s_3\}$, $\{\uparrow \{a_1, a_2\}, \uparrow \{a_3, a_4\}, \uparrow \{a_5, a_6\}\}$, $\{e_{write_1}, e_{write_2}, e_{write_3}\}$ and $\uparrow \{a_{i_1}, a_{i_2}\}$. The assignments $\{s_1, s_2, s_3\}$ and $\{e_{write_1}, e_{write_2}, e_{write_3}\}$ are eventually discarded when the verb is being processed. The assignment $\uparrow \{a_{i_1}, a_{i_2}\}$ corresponds to the reading according to which each student sent his own articles to L&P and no other articles. By contrast, the assignment $\{\uparrow \{a_1, a_2\}, \uparrow \{a_3, a_4\}, \uparrow \{a_5, a_6\}\}$ yields the reading according to which each student sent all six articles to the journal.[48]

Finally, consider the reading on which the internal argument gets a distributive reading. In this case processing 'two articles' inside the distributive loop introduces a set $\{a_{i_1}, a_{i_2}\}$ and a set of two writing events $\{e_{write_{i_1}}, e_{write_{i_2}}\}$. Each

[48] A question that must be left unanswered is what object should be assigned to 'them' in the output possibility. According to the definition of δ, this is the set of six articles. However, the editors of L&P received 18 'objects'. Hence, what is at stake here is the distinction between an article as a (probably) abstract object and a physical or electronic copy of this object.

writing event must be related to exactly one article and vice versa. After processing the first sentence one therefore gets three discourse objects: $\{s_1, s_2, s_3\}$, $\{a_1, \ldots, a_6\}$ and $\{e_{write_1}, \ldots, e_{write_6}\}$. Furthermore, there is a discourse dependency between student s_i and the set $\{a_{i_1}, a_{i_2}\}$, using Definition 22, since one has in the associated event frames: $[\![\otimes\text{ACTOR} \bullet \text{THEME}]\!](s_i)(a_{i_1})(f_{e_{i1}})$ and $[\![\otimes\text{ACTOR} \bullet \text{THEME}]\!](s_i)(a_{i_2})(f_{e_{i2}})$. When processing the second sentence this dependency is temporarily added to the stack together with the loop object. Since this discourse object satisfies the cardinality constraint imposed by 'them', it is a possible antecedent for this pronoun. Shifting the set to the group level in order to get the required collective reading yields the reading where each student sent his own articles to L&P. This reading corresponds to the one we already got if the first sentence interprets the internal argument collectively.

Inspecting the above derivations one see that the downdate problem of other approaches is avoided due to the following two factors. First, and most importantly, dependencies can be accessed without having to use a substate of the present information state. This is the fact because dependencies are stored in frames, and hence semantically, and not, as this is the case in other approaches, structurally in the rows of variable assignments. Second, using Incremental Dynamics it becomes possible to temporarily extend the stack inside a distributive loop by adding the object relative to which the loop is executed together with its discourse dependencies.

The above derivations also show that using frames a context contains additional information that is needed to know which reading a sentence got. For example, only looking at the object component of the interpretation of 'They sent them to L&P', no dependencies are visible. Hence, both a dependent and an independent reading are compatible with this information. One rather has to inspect the associated frame components, in particular that of the sending events. On the independent reading, each e_{send_i} is related to the set of all six articles by the THEME-attribute (or the corresponding group of six articles. By contrast, on a dependent reading, e_{send_i} is related only to the set of two articles (group of two articles) written by student s_i.

4.11 Frame theory at work: the resolution of pronouns

The derivations in the last section have shown how pronouns can be assigned discourse objects in such a way that one gets correct truth conditions. Pronouns are interpreted as restricted domain extension operations. They pick up a discourse object already on the stack. Inside a distributive loop the current context is temporarily extended by the loop object and its discourse dependencies. As a result, both dependent and independent antecedents become available. Using information about the actual truth of sentences in a possible world, possibilities assigning discourse objects which result in false sentences are discarded. This last step is not open to a comprehender (hearer) since he does not know which assignments of discourse objects yield true sentences. Hence, he has to use other strategies in order to exclude possible assignments of discourse objects to eventually arrive at a single assignment. By way of further illustration, consider as another example (58).

(58) Amanda amazed Brittany. Shed had passed the exam.

Not knowing whether Amanda or Brittany had passed the exam, a comprehender has to use other kinds of information for resolving the pronoun in the second sentence. There are at least four sources of information which he can use in this process: (i) accessibility of an antecedent, (ii) expectations about which objects are mentioned next, independent of the way they are referred to, (iii) the probability that a reference to an object will be by a pronoun and (iv) bottom-up information a comprehender gets about possible referents of the pronoun after it has been introduced into the discourse.

Information based on expectations (sources (ii) and (iii)) has been extensively investigated by Kehler and colleagues (Kehler et al., 2008; Kehler and Rohde, 2013; Rohde and Kehler, 2014; Kehler and Rohde, 2017). They found that the interpretation of pronouns is the result of integrating top-down expectations about who or what will be mentioned next with the pronoun's linguistic function of indicating a continuation of the current topic (Kehler and Rohde, 2013; Kehler et al., 2016). In contrast to both centering theory and coherence-driven approaches this theory is based on a probabilistic Bayesian model. The key equation is given in (59).

(59)

$$pr(\mathit{referent}|\mathit{pronoun}) = \frac{pr(\mathit{pronoun}|\mathit{referent}) \cdot pr(\mathit{referent})}{\sum_{\mathit{referent} \in \mathit{referents}} pr(\mathit{pronoun}|\mathit{referent}) \cdot pr(\mathit{referent})}$$

According to (59), the probability $pr(\mathit{referent}|\mathit{pronoun})$ that a given pronoun *pronoun* refers to the object denoted by *referent* is the product of two probabilities.[49] $pr(\mathit{referent})$ represents the next-mention bias (source (ii)): the probability that the object denoted by *referent* will get mentioned next, regardless of the expression used. By contrast, $pr(\mathit{pronoun}|\mathit{referent})$ is the production bias: given that a speaker wants to refer to the object denoted by *referent*, it is the probability that a pronoun is used.

The approach of Kehler and colleagues accounts for expectations which comprehenders have w.r.t. which objects are mentioned next and the reference to which objects is most likely to be in form of a pronoun. However, expectations can go wrong or, more precisely, they have to be confirmed by bottom-up information that is got by processing the sentence in which the pronoun occurs and also subsequent discourse following this sentence. Hence, pronoun interpretation is a non-monotone process which is based both on predictions and subsequent bottom-up information (modulo accessibility). In the following section we will

[49] The denominator is used to normalize the probabilities to 1 so that one gets a probability distribution. It is the probability that a pronoun is the form of reference chosen by the speaker, $pr(\mathit{pronoun})$, which is computed by summing over all referents that are compatible with the pronoun (Kehler et al., 2008; Kehler and Rohde, 2013).

show how frame theory can be used to exclude possible antecedents based solely on reasoning about sorts and relations and therefore independently of any considerations related to actual truth conditions of the sentences in a text and the actual values of chains of attributes.

4.12 Resolution based on bottom-up information

The information stored in the frame component of discourse objects on the stack in a possibility involves information about values, i.e. objects that have been introduced during semantic processing. By contrast, the values of θ, θ^* and θ_{sort} provide information about the sortal and relational structure of a frame that is independent of any values that satisfy the relation in the chain sets for a given frame. Hence, this information can be used by a comprehender independently of knowing these values and the truth conditions of sentences. In particular, a comprehender uses this information as part of his strategy in the process of resolving pronouns.

If a pronoun is assigned a discourse object already on the stack, this means that a comprehender gets additional information about this object. In our frame theory this information has the form of derived event-related frames which contain information about the object's participation in an event or a state as well as frame information provided by the pronoun, for instance sortal information about the value of the SEX attribute in the case of 'it'. Both kinds of information are got from bottom-up processing. Let's illustrate this with our running example.

(60) Three students each wrote an article. They each sent it to L&P.

For assigning discourse objects to pronouns, a comprehender uses the strategy applied in the derivations in section 4.10. Suppose 'they' is assumed to be anaphorically related to 'three students' in the first sentence. In this case one gets the following sources of information: (i) information associated with 'three students', (ii) information associated with the writing event: the three students are the actor of this event and (iii) information associated with the sending event: the students are the actor of this event. In our frame theory this information is given in the frame components of the corresponding objects.[50]

A discourse object on the stack can only be the antecedent of the pronoun if the information from these sources is consistent, otherwise the corresponding possibility is discarded. One way of defining consistency of a set of frames F_{D_a} associated with a set of objects D_a is in terms of the existence of a frame f which satisfies the sortal and relational constraints imposed by the f_i. f can exist only if the sortal information at the roots of the frames in the set F_{D_a} is consistent, i.e. if the following condition holds. Let $s_1, \ldots, s_n$ be the sorts at the root of the frames in F_{D_a}. If $s_i \sqcap s_j = \bot$ for $1 \leq i, j \leq n$, the frame f cannot exist. Hence

[50] Hence for ambiguous pronouns, a comprehender waits until the sentence is completed. This is in accordance with empirical data from the way pronouns are resolved (see for example Corbett and Chang, 1983).

the condition $s_i \sqcap s_j \neq \bot$ for $1 \leq i, j \leq n$ is a necessary condition for f to exist. This leads to rule (i).

(61) An object D_a on the stack can be a possible referent for a pronoun only if the elements of the set of root sorts of the frames in F_{D_a} are pairwise consistent.

For a discourse object with object component D_a it is sufficient to check (61) for an element $d_a \in D_a$ since the elements of D_a do not differ w.r.t. sortal information. (61) is checked by a comprehender using θ. Recall that sortal information at the root is expressed by the relation formula $\Delta \cap \downarrow s$ and that $[\![\Delta \cap \downarrow s]\!] \in \theta(f)$ for a frame f of sort s. One then gets (62).

(62) $\exists d_a.\exists f : root(f) = d_a \wedge [\![\Delta \cap \downarrow s_1 \cap \ldots \cap \downarrow s_n]\!](d_a)(d_a)(f)$.

(62) requires that there be a frame that satisfies all sortal information at its root. Rule (i) must be generalized to the sortal condition for each chain common to two or more frames in F_{D_a}.

(63) An object D_a on the stack can be a possible referent for a pronoun only if the frames in F_{D_a} are pairwise sort consistent for each chain common to both chain sets θ.

Let $F'_{D_a} = \{f_1, \ldots, f_j\} \subseteq F_{D_a}$ s.t. $[\![\pi \cap \uparrow s_i]\!] \in \theta(f_i), 0 \leq i \leq j$ for some chain π. Note that rule (ii) refers to elements of θ_{sort}. Similar to the first rule, it is sufficient to test the requirement for a single element $d_a \in D_a$. One then gets (64).

(64) $\exists d_a.\exists d'.\exists f : [\![\pi \cap \uparrow s_1 \cap \ldots \cap \uparrow s_j]\!](d_a)(d')(f)$.

(64) imposes the same condition as (62) except that it projects not at the root but at the end of a chain π.

Rules (i) and (ii) refer to sortal and relational information. Application of these rules need not necessarily yield a unique antecedent. In such cases other sources of information have to be applied. One kind of such information is dependency information. Recall that at least in general a dependent reading is preferred relative to an independent one if both readings are possible. Consider again (65).

(65) Three students each wrote two articles. They each sent them to L&P.

On a dependent reading each students sent his (two) articles to L&P whereas on the independent reading each student sent all six articles to this journal. Though both readings are possible, the dependent one is preferred. This yields rule (iii).

(66) If after application of rules (i) and (ii) two objects D_a and D'_a are possible antecedents, dependency information can be used to arrive at a decision.

Rule (iii) is not strict in the sense that it excludes a possibility. Rather, it allows to take a decision based on predictions and/or expectations.[51] In the next section these rules will be applied to our running examples.

4.13 Applying the rules

We start again with our running example, this time with the first sentence getting a distributive reading and the second a dependent one.

(67) Three students each wrote an article. They each sent it to L&P.

Let us begin with the information a comprehender gets from semantically processing the second sentence. First, there is information about the dependency between the two looked for antecedents in the sending event. They are related by the chain ⊗ACTOR • THEME. Second, there is sortal information about these two objects which is given by the constraints on the source sort and the target sort of the chain connecting the two objects in a frame of the sending event. The actor must be of type **person** and the theme must be a physical object that can be sent.[52] Hence, the elements d_a of possible antecedents D_a must satisfy the constraints in (68).[53]

(68) 1. $f_{\text{ACTOR}}, d_a \models do(\otimes\text{ACTOR} \bullet \text{THEME} \cap \downarrow \textbf{person})$.
2. $f_{\text{THEME}}, d'_a \models do(\otimes\text{THEME} \bullet \text{ACTOR} \cap \downarrow \textbf{physical\_object}_{sendable})$.

In (68) the frames are event-related frames relative to the ACTOR and the THEME attribute that are derived from the event frame. Let's begin with 'they'. Possible antecedents are the sets $\{s_1, s_2, s_3\}$, $\{e_1, e_2, e_3\}$ and $\{a_1, a_2, a_3\}$. Applying rule (i) which checks the consistency of the sortal information, leaves $\{s_1, s_2, s_3\}$ as the only possible antecedent because only the students satisfy the sortal restriction to persons imposed by 'send' on the value of the ACTOR attribute.[54]

Next, we will show how 'it' can be resolved. Recall that s_1 and a_1 are the only options. First, there is the sortal information which must be satisfied. In this case the only information one has got is that it is of sort $\textbf{physical\_object}_{sendable}$. Let's assume that this condition is satisfied by both s_1 and a_1. However, there is also sortal information about an attribute. 'It' imposes the sortal constraint on the value of the SEX-attribute: it must be neutral. This constraint is not satisfied by an object of sort **person** and, therefore, by an object of sort **student**. Importantly, f_{d_a}, and hence $\theta_{sort}(f_{d_a})$ need not itself contain information about

[51] Though Kehler & Rohde have not investigated dependent uses of pronouns, it is in relation to such uses that their probabilistic model can be applied.

[52] Recall that sortal constraints in chains are in general stronger than the sortal restrictions imposed by the source and the target sort. For example, for the theme of a sending event the sortal restriction is given by the condition that it must be a sendable physical object.

[53] In this section we use the frame definition from the first part of this paper.

[54] One may argue that also objects like computers can send something. In this case the sortal information has to be refined.

the SEX-attribute. This information can be given in one of its extensions f' given in the value of $\theta_{sort}(f')$. Hence, s_1 is excluded as a possible antecedent of 'it' and the article a_1 is the only remaining candidate.

As a second example we consider the plural variant of the first example in which each student wrote two articles.

(69) Three students each wrote two articles. They each sent them to L&P.

The resolution of 'they' proceeds in exactly the same way as in the case of the first example since the newly introduced discourse objects do not differ at the level of sortal information. For 'them', a comprehender gets the sets $\{a_{i_1}, a_{i_2}\}$ and $\{a_1, \ldots, a_6\}$ since both pass the test imposed by rules (i) and (ii). So far dependency relations have not been considered. On the first reading, 'them' anaphorically refers to the set of six articles cumulatively written by the three students. However, on a distributive reading there is no dependency between the set of three students and the set of six articles. By contrast, if 'them' is taken to be anaphorically related to the set $\{a_{i_1}, a_{i_2}\}$, there is a dependency between $\{s_i\}$ and this set because it relates subsets of the drefs $\{s_1, s_2, s_3\}$ and $\{a_1, \ldots, a_6\}$. Applying rule (iii), one gets that the dependent reading is the preferred one.

From the considerations in this section the following broad picture of the strategy used by a comprehender in the resolution of pronouns emerges. It can be seen as a three-stage process: (i) application of the Kehler & Rohde criteria based on expectations and predictions, yielding a probability distribution for (accessible) antecedents, i.e. discourse objects on the stack, and (ii) update of this distribution with bottom-up information based on sortal information coming from θ_{sort} as outlined in this section.[55]

The part of the resolution strategy considered in this section not only used frames of sort 'individual' but also event-related frames. However, above in section 4.7 we only considered frames of sort individual and did not store the event-related frames. This was possible because we were concerned only with the truth-conditional aspect. At that level event-related frames are not needed because a sort or relation formula that is true in such derived frames always has a corresponding formula that is true in the original event frame. As the above examples have shown, by not storing information related to derived event-related frames of sort 'individual' with the object one looses discourse information that is relevant for a comprehender to link different expressions referring to one and the same object, as in the case of anaphoric relations linking a pronoun to an antecedent. In order to integrate this kind of information the following changes are necessary: (i) the kind of objects stored in a position on a stack has to be changed, and (ii) the interpretation of verbs in the lexicon has to be modified.

The two different kinds of frames are related to two different kinds of expressions: verbs are related to event-related frames whereas common nouns and pronouns are related to frames of sort 'individual'. This distinction can be cap-

[55] In Naumann et al. (2017) we provide a first account of integrating a probabilistic component in our theory. Naumann and Petersen (2017) applies this extended frame theory to bridging phenomena.

tured by having the frame component of a discourse object be pair s.t. the first projection is a frame of sort 'individual' and the second projection is a set of event-related frames. On this modelling, common nouns and pronouns always update the first projection of the frame component whereas verbs update the second projection.

Updates related to the first projection are already accounted for in the interpretation of CN's and pronouns by the requirement that there has to be an extension of the frame in the input possibility that satisfies the sortal and/or relational information provided by the expression. Below the revised interpretations for common nouns and 'write' are given.

(70) $\lambda j.\lambda i.\lambda s.\lambda s'.\exists c.\exists w.\exists c'.\exists e.\exists f.\exists d.\exists f_d.\exists F_d.\exists f_{d_{\text{ACTOR}}}.\exists d'.\exists f_{d'}.\exists F_{d'}.\exists f_{d'_{\text{THEME}}}$ $(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge c' = c^{\frown}\langle e, f\rangle \wedge root(f) = e \wedge IN(f) = w \wedge [\![write]\!](e)(f) \wedge [\![\text{ACTOR}]\!](e)(d)(f) \wedge [\![\text{THEME}]\!](e)(d')(f) \wedge$ $\theta(f) = \{[\![\text{ACTOR}]\!], [\![\text{THEME}]\!]\} \cup upset_{\Delta}(write) \wedge c[i] = \{\langle d, \langle f_d, F_d\rangle\rangle\} \wedge$ $c'[i] = \{\langle d, \langle f_d, F_d \cup \{f_{d_{\text{ACTOR}}}\}\rangle\rangle\} \wedge c[j] = \{\langle d', \langle f_{d'}, F_{d'}\rangle\rangle\} \wedge$ $c'[j] = \{\langle d', \langle f_{d'}, F_{d'} \cup \{f_{d'_{\text{THEME}}}\}\rangle\rangle\} \wedge \forall k(0 \leq k < |c| \wedge k \neq i \wedge k \neq j \rightarrow$ $c'[k] = c[k]))$

(71) $\lambda i.\lambda s.\lambda s'.\exists c.\exists w.\exists c'(s = \langle c, w\rangle \wedge s' = \langle c', w\rangle \wedge |c'| = |c| \wedge c'[j] = c[j]\, for (0 \leq j < |c| \wedge j \neq i) \wedge \forall\langle d, \langle f_d, F_d\rangle\rangle \in c[i].\exists f'([\![s]\!](d)(f') \wedge f \sqsubseteq_F f' \wedge \theta(f') = \theta(f_d) \cup upset_{\Delta}(s)) \wedge c'[i] = \{\langle d, \langle f', F_d\rangle\rangle \mid \exists\langle d, \langle f_d, F_d\rangle\rangle \in c[i] : f \sqsubseteq_F f' \wedge [\![s]\!](d)(f') \wedge \theta(f') = \theta(f_d) \cup upset_{\Delta}(s)\})$

One additional advantage of also storing event-related frames is that if a discourse object is reintroduced by the interpretation of a pronoun, all its dependencies are automatically accounted for.[56]

4.14 Event-related frames

Each frame for an event e is related to a frame of sort **individual** for one of the objects participating in the event. This frame specifies the role and dependencies which this individual has in the event frame. Since an individual can be related to an event by more than one attribute, the frame is relativized to a particular attribute.[57]. Such derived frames will be called *event-related frames relative to an attribute* or *event-related frames* for short. Consider again the example of a frame for a writing event below.

In the above example one has the frames $f_{\text{THEME},e}$ and $f_{\text{ACTOR},e}$, for e the event of writing. Since an event-related frame is derived from an event frame, their chain sets are related in a particular way. Given the chain set $\theta(f)$ of an event frame f, the chain set of the event-related frame relative to attribute ATTR is construed as follows. Chains of the form $\pi = \text{ATTR}\bullet\pi'$ get replaced by π and chains $\otimes\text{ATTR}\bullet\pi$ replace chains of the form $\pi = \text{ATTR}' \bullet \pi'$ with $\text{ATTR}' \neq \text{ATTR}$. For the root one

[56] See the discussion in section 4.6 above.

[57] It is assumed that ATTR is an attribute of depth 1, i.e. an attribute at the root of the frame.

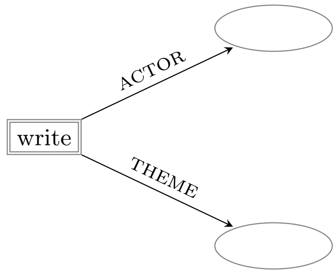

Fig. 4. Frame for 'write'

has: $root(f_{\text{ATTR},e}) = \iota d.[\![\text{ATTR}]\!](e)(d)(f_e)$. For the truth of relation formulas one has: $\forall d'.R(e)(d')(f_e) \leftrightarrow R'(d)(d')(f_{\text{ATTR},e})$ for $R = [\![\text{ATTR} \bullet \pi]\!]$ and $R' = [\![\pi]\!]$ or $R = [\![\pi]\!]$ and $R' = [\![\otimes\text{ATTR} \bullet \pi]\!]$, $\pi = \text{ATTR}' \bullet \pi'$ and $\text{ATTR}' \neq \text{ATTR}$.

5 Conclusion

In this article we have presented a formal theory of frames. Let us summarize the main points in using frame theory.

– Dependency information is defined purely semantically. Discourse dependency is the restriction of dependency to objects that have been introduced into a discourse.
– Dependency information is stored locally in the frame component of events.
– Information about objects got from bottom-up processing is stored in the frame component instead of in variable assignments. Hence, discourse information is not related to the use of variables.
– Resolution of pronouns based on information got from bottom-up processing can partly be done on the basis of information about the sortal and relational structure of frames without knowing the actual truth values of sentences.

Information about dependencies between discourse objects is necessary in cross-sentential anaphora in order to arrive at correct truth condition. Hence, this information has to be stored in an information state. Storing discourse dependencies 'vertically' in sets of assignments leads to the problem of information downdate: inside of a distributive loop the global objects of which the dependent ones are elements are no longer available. Storing dependencies semantically in frames solves this problem. The discourse dependency information for an object is stored in the event frames of events in the context in which this object participates. This information is available both inside and outside of a distributive loop. This global availability can be used to temporarily extend the current context inside a loop both with the loop object and all its discourse dependencies. Since these objects are added, the 'global' discourse objects of which the dependent ones are elements are still on the stack and therefore available for anaphoric reference. This global availability is linked to the fact that frames are one building block of a possibility.

The resolution strategy used by a comprehender also relies on this temporary extension of the current context with the objects dependent on the loop object in order to assign discourse objects to pronouns. But it goes beyond this use of frames. Semantically relating a pronoun to an object already on the stack means getting more information about this object. In our frame theory this information is stored in the frame components of discourse objects. This information must be consistent. In our frame theory consistency checking is done using the information about the sortal and relational structure of frames given by the values of the function θ_{sort} for the frames associated with a possible antecedent object as well as extensions of θ_{sort}. This information is partly independent of discourse information because it can involve sortal constraints on attributes whose values are not (yet) given in the discourse. In addition, this information does not rely on knowledge of the actual truth or falsity of sentences in a discourse.

Our results also support Muskens (2013) position which claims that the frame hypothesis in (1) is false. Frames apply to the lexical layer but not to the phrasal layer. The latter level is concerned with building larger information units from simpler ones, ultimately relying on atomic (lexical) units.

Future applications that are close to the two applications considered here are the following: (i) quantificational NPs. So far, only indefinites have been considered. Quantificational NPs like 'most N' or 'at least N' pose additional questions like maximality ('At most three students wrote an article') and the weak/strong distinction ('Every farmer who owns a donkey beats it' (strong) vs. 'Every person who had a dime in his pocket put it in the parking meter' (weak)); (ii) extending the resolution process for pronouns by integrating the Kehler-Rohde approach and accounting for accessibility constraints; this means that the current theory has to be extended by a probabilistic component; (iii) incorporating bridging phenomena, in particular those involving definite NPs of the form 'the N' (see Naumann and Petersen, 2017, for a proposal). A more general question is: how can the strategy for pronoun resolution be integrated into the compositional derivation of sentences/texts? Another topic is the interpretation of verbs. How can verb meanings be represented beyond the decomposition using thematic relations? (see Naumann et al., 2017, for an analysis of 'rise'). More generally: How can tense and aspect be modelled in our frame theory?

6 Appendix

In this appendix we define the type system and the polymorphic character of Incremental Dynamics. The set of types is defined in (72).

(72)
1. $N := 0 \,|\, 1 \,|\, 2 \,|\, \ldots$
2. $Type := o \,|\, d \,|\, e \,|\, \eta \,|\, s \,|\, f \,|\, t \,|\, N \,|\, [N] \,|\, Type \rightarrow Type \,|\, Type \circ Type$

The type of objects is o with subtypes e (event) and d (individual). s is the type of possible worlds and f the type of frames. Both types are subtypes of the type

η. N is the type of context indices with domain $D_N = \mathbb{N}^0$.[58] t is the (logical) type of truth values with domain $D_t = \{0, 1\}$. $[N]$ is the type of stacks (or contexts) of length n with domain $D_{[N]}$ the set of stacks of finite length. $Type_1 \to Type_2$ is the type of functions from $Type_1$ to $Type_2$. We follow standard practice and write $\langle Type_1, Type_2 \rangle$ for $Type_1 \to Type_2$. $Type_1 \circ Type_2$ is the type of pairs (product type) with first projection of type $Type_1$ and second projection of type $Type_2$. Hence, $o \circ f$ is the type of pairs whose domain consists of pairs of an object and a frame. For sets of such pairs, we use ρ, i.e. $\rho = \langle o \circ f, t \rangle$. For $[n]$ we use $[\rho]_i$ with the index variable indicating the length of the context, i.e. $i :: n$.

Incremental Dynamics is inherently polymorphic. Consider e.g. the following definition of context extension where a context is taken to be stack whose values are discourse objects.[59]

(73) $\exists^0 := \lambda c.\lambda c'.\exists A(c' = c^\frown A)$

In order for this operation to be well-defined the lengths of the contexts must match. If c has length n, c' must be of length $n + 1$. Hence, $\exists$ relates contexts of length n to contexts of length $n + 1$. Therefore, the type of $\exists$, which is $[\rho] \to ([\rho] \to t)$, is a polymorphic type. This type polymorphism can be made explicit by writing the type of a context c as $[\rho]_i$, with i a type variable indicating the context length. Using $[\rho]_i$, the type scheme for $\exists$ is $[\rho]_i \to ([\rho]_{i+1} \to t)$ where $i :: n$ for any $n \in N$. Hence this type scheme generalizes over the types $[\rho]_0 \to ([\rho]_1 \to t)$, $[\rho]_1 \to ([\rho]_2 \to t)$. Consider as a second example of the polymorphic character $\lambda c\lambda j.c[j]$ with $|c| = i$ and $i :: n$ so that one has $c :: [\rho]_i$, i.e. c is a context of length n. In order that the type of i fits the size of the context c, j must be of a type m with $m \leq n$. Assuming the Von Neumann definition of the natural numbers, an index of type n ranges over $\{0, \ldots, n-1\}$. Hence, it is sufficient to require $j :: n$ in order for j to fit the size of the context. For example, for $n = 2$, one gets the type $[\rho]_2 \to (2 \to e)$. Since the length of the context is not known in advance, one has to generalize over the length of the context. We follow Van Eijck and use ι as a type for context indices of arbitrary length and assume that the indices always fit the size of the context. Hence, one has $\lambda c\lambda j.c[j] :: [\rho] \to (\iota \to \rho)$, or, if it is not assumed that the index fits the size: $\lambda c\lambda j.c[j] :: [\rho]_\iota \to (\iota \to \rho)$. Again $[\rho]_\iota \to (\iota \to \rho)$ is a type scheme rather than a type because it is polymorphic, abbreviating the (infinite) set of types $[\rho]_0 \to (0 \to \rho)$, $[\rho]_1 \to (1 \to \rho)$, $[\rho]_2 \to (2 \to \rho)$ and so on. More generally, in a type scheme of the form $\iota \to ([\rho] \to ([\rho] \to \alpha))$, one generalizes over the length of the initial context assuming that the index fits the length of the initial

[58] The definition of the type of indices is based on the Von Neumann encoding of natural numbers: $n = 0, \ldots, n-1$. From this it follows that an index of type n is an index ranging over all 'smaller' indices $0, \ldots, n-1$. More formally one has:

(i) 1. $0 := \emptyset$.
2. $s(n) := n \cup \{n\}$.

[59] Hence, this definition is different from the one given in the text where a context (= possibility) is a pair consisting of a stack and a possible world.

context. Hence, the above type scheme is shorthand for $0 \to ([\rho]_0 \to ([\rho] \to \alpha))$, $1 \to ([\rho]_1 \to ([\rho] \to \alpha))$, $2 \to ([\rho]_2 \to ([\rho] \to \alpha))$ and so on.

Bibliography

Barsalou, L. W. (1992). Frames, concepts, and conceptual fields. *Frames, Fields and Contrasts. New Essays in Semantic and Lexical Organization*, 21–74.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences 22*, 577–660.

Brasoveanu, A. (2008, April). Donkey pluralities: plural information states versus non-atomic individuals. *Linguistics and Philosophy 31*(2), 129–209.

Champollion, L. (2015, Feb). The interaction of compositional semantics and event semantics. *Linguistics and Philosophy 38*(1), 31–66.

Cooper, R. (2010). Frames in formal semantics. In H. Loftsson, E. Rögnvaldsson, and S. Helgadóttir (Eds.), *Advances in Natural Language Processing: 7th International Conference on NLP, IceTAL 2010, Reykjavik, Iceland, August 16-18, 2010*, pp. 103–114. Berlin, Heidelberg: Springer Berlin Heidelberg.

Corbett, A. T. and F. R. Chang (1983, May). Pronoun disambiguation: Accessing potential antecedents. *Memory & Cognition 11*(3), 283–294.

de Rijke, M. (1994). Meeting some neighbours. In J. van Eijck and A. Visser (Eds.), *Logic and Information Flow*, pp. 170–195. Cambridge, MA, USA: MIT Press.

De Rijke, M. (1995, Sep). The logic of Peirce algebras. *Journal of Logic, Language and Information 4*(3), 227–250.

Dekker, P. (1993). *Transsentential meditations - Ups and downs in dynamic semantics*. Ph. D. thesis, ILLC, University of Amsterdam.

Fernando, T. (2016). Types from frames as finite automata. In A. F. et al. (Ed.), *Formal Grammar; 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*, Volume 9804 of *Theoretical Computer Science and General Issues*, pp. 19–40. Springer.

Groenendijk, J., M. Stokhof, and F. Veltman (1997). Coreference and modality. In S. Lappin (Ed.), *Handbook of Contemporary Semantic Theories* (1st. ed.)., pp. 179–213. Oxford: Blackwell.

Heim, I. (1990). E-type pronouns and donkey anaphora. *Linguistics and Philosophy 13*(2), 137–77.

Kallmeyer, L. and R. Osswald (2013). Syntax-driven semantic frame composition in lexicalized tree adjoining grammars. *J. Language Modelling 1*(2), 267–330.

Kehler, A., L. Kertz, H. Rohde, and J. Elman (2008). Coherence and coreference revisited. *Journal of Semantics 25*(1), 1–44.

Kehler, A., L. Kertz, H. Rohde, and J. Elman (2016). Evaluating an expectation-driven question-under-discussion model of discourse interpretation. *Discourse Processes 53*, 1–20.

Kehler, A. and H. Rohde (2013). A probabilistic reconciliation of coherence-driven and centering-driven theories of pronoun interpretation. *Theoretical Linguistics 39*(1-2), 1–37.

Kehler, A. and H. Rohde (2017). Evaluating an expectation-driven question-under-discussion model of discourse interpretation. *Discourse Processes 54*(3), 219–238.

Krifka, M. (1996). Parameterized sum individuals for plural anaphora. *Linguistics and Philosophy 19*(6), 555–598.

Landman, F. (2000). *Events and Plurality*. Dordrecht: Kluwer Academic Publishers.

Löbner, S. (2011). Concept types and determination. *Journal of Semantics 28*(3), 279–333.

Löbner, S. (2013). *Understanding Semantics* (2nd ed.). Understanding Language Series. Oxford: Routledge.

Löbner, S. (2014). Evidence for frames from human language. In T. Gamerschlag, D. Gerland, R. Osswald, and W. Petersen (Eds.), *Frames and Concept Types*, Volume 94 of *Studies in Linguistics and Philosophy*, pp. 23–67. Springer International Publishing.

Löbner, S. (2017). Frame theory with first-order comparators: Modeling the lexical meaning of punctual verbs of change with frames. In H. H. Hansen, S. E. Murray, M. Sadrzadeh, and H. Zeevat (Eds.), *Logic, Language, and Computation: 11th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2015, Tbilisi, Georgia, September 21-26, 2015, Revised Selected Papers*, pp. 98–117. Berlin, Heidelberg: Springer Berlin Heidelberg.

Muskens, R. (2013). Data semantics and linguistic semantics. In M. Aloni, M. Franke, and F. Roelofsen (Eds.), *The dynamic, inquisitive, and visionary life of* $\phi, ?\phi, and \diamond\phi$*; A festschrift for Jeroen Groenendijk, Martin Stokhof, and Frank Veltman*, pp. 175–183. UvA.

Naumann, R. and W. Petersen (2017). Underspecified change - bridging inferences in a dynamic, probabilistic frame theory. paper presented at TbiLLC 2017, to be submitted to the proceedings.

Naumann, R., W. Petersen, and T. Gamerschlag (2017). Underspecified changes: a dynamic, probabilistic frame theory for verbs. paper presented at Sinn und Bedeutung, to appear in the proceedings.

Nouwen, R. (2003). *Plural pronominal anaphora in context*. Ph. D. thesis, Netherlands Graduate School of Linguistics Dissertations, LOT, Utrecht.

Nouwen, R. (2007). On dependent pronouns and dynamic semantics. *Journal of Philosophical Logic 36*(2), 123–154.

Petersen, W. (2007). Representation of concepts as frames. In J. Skilters, F. Toccafondi, and G. Stemberger (Eds.), *Complex Cognition and Qualitative Science*, Volume 2 of *The Baltic International Yearbook of Cognition, Logic and Communication*, pp. 151–170. University of Latvia.

Rohde, H. and A. Kehler (2014). Grammatical and information-structural influences on pronoun production. *Language, Cognition, and Neuroscience 29*(8), 912–927.

Smolka, G. (1988). A feature logic with subsorts. Technical report, IBM Deutschland.

van Benthem, J. (1996). *Exploring Logical Dynamics*. Center for the Study of Language and Information.

Van den Berg, M. (1996). *Some aspects of the internal structure of discourse. The dynamics of nominal anaphora*. Ph. D. thesis, University of Amsterdam.

van den Berg, M. (1996). *Some aspects of the internal structure of discourse: the dynamics of nominal anaphora*. Ph. D. thesis, ILLC, Amsterdam.

van Eijck, J. (2001a). Border crossings. In *Logic in Action*, pp. 51–74. Amsterdam: ILLC.

van Eijck, J. (2001b). Incremental dynamics. *Journal of Logic, Language and Information 10*(3), 319–351.