

A Logical Approach to Data-Driven Classification

Rainer Osswald¹ and Wiebke Petersen²

¹ Praktische Informatik VII, FernUniversität Hagen,
Universitätsstraße 1, 58084 Hagen, Germany,

`rainer.osswald@fernuni-hagen.de`,

² Institut für Sprache und Information,

Abteilung Computerlinguistik, Heinrich-Heine-Universität Düsseldorf,
Universitätsstraße 1, 40225 Düsseldorf, Germany,

`wiebke.petersen@uni-duesseldorf.de`

Abstract. We present a flexible approach for extracting hierarchical classifications from data, which employs the logic of affirmative assertions. The basic observation is that each set of rules induced by the data canonically determines a classificational hierarchy. We give a characterization of how the chosen rule type affects the structure of the induced hierarchy. Moreover, we show how our approach is related to Formal Concept Analysis. The framework is then applied to the induction of hierarchical classifications from an amino acid database. Based on this example, the pros and cons of several types of hierarchies are discussed with respect to criteria such as compactness of representation, suitability for inference tasks, and intelligibility for the human user.

1 Introduction

The logic of affirmative assertions, which has its origins in domain theory and program semantics [16, 1], has proved useful for the study of formal classification as well [10, 11]. Its close connection with Formal Concept Analysis [4] has already been observed in [13], where the emphasis was on the classification of linguistic data; see also [15, 14]. In this paper, the framework is illustrated by inducing hierarchical classifications from biochemical data. Another potential area of application is the generation of ontologies for the Semantic Web.

The key observation underlying our approach is that, first, any set of implicational statements which correctly describe a given classification table uniquely determines a hierarchical classification of that data, and, second, restrictions on the form of the statements systematically correspond to structural properties of the induced classificational hierarchy. For instance, if only atomic attributes are allowed as premise and conclusion, then the resulting hierarchy is a distributive lattice. Viewed from another perspective, our approach helps to clarify the relation between a theory and its *information domain* [2], where the latter is the ordered universe of admissible combinations of atomic attributes.

The rest of the paper is organized as follows: In Section 2, each theory consisting of universally quantified Boolean predicates is shown to determine an information domain. In addition, it is shown how the information domain of a theory depends on the class the theory belongs to. This result is used in Section 3 for inducing different types of conceptual hierarchies from classification tables. In the case of Λ -free Horn theories, the induced hierarchies essentially coincide with the concept lattices of Formal Concept Analysis.³ In Section 4, we give an illustration of how our framework can be used to generate hierarchical classifications from biochemical data. We discuss the effects of varying the underlying theory class. Moreover, we consider the selective addition of disjunctive rules.

2 Theories and Information Domains

2.1 Terms and Theories

Let Σ be a set of *atomic* one-place predicates. The logical framework employed in the following is a small fragment of first-order predicate logic. We will frequently make use of a variable-free notation. For instance, if ϕ and ψ are one-place predicates, then $\phi \wedge \psi$ is their logical conjunction. In addition, we introduce two special one-place predicates V and Λ that are respectively satisfied by everything and nothing of the universe of discourse. Moreover, $\forall\phi$ stands for $\forall x(\phi x)$, where ϕ is a one-place predicate. Finally, let \preceq and \equiv be two binary term operators such that $\phi \preceq \psi$ and $\phi \equiv \psi$ are $\forall(\phi \rightarrow \psi)$ and $\forall(\phi \leftrightarrow \psi)$, respectively.

As the attentive reader will notice, we could equally well make use of plain propositional logic instead. The reason for adopting the predicational viewpoint is conceptual clarity. For, in classification tasks, we basically have to deal with ascriptions of certain properties or attributes to certain entities, i.e., with predications. Furthermore, we are concerned with statements of the sort that everything with property ϕ also has property ψ , that is, with universally quantified conditionals. An additional point in favor of the predicational view is the fact that every theory consisting of such universal conditionals naturally defines a universal model of that theory, the elements of whose universe can be regarded as the “generic entities” classified by the theory (see also [12]).

Definition 1 (Term/Statement/Theory). *A Boolean term over Σ is inductively built by \wedge , \vee , \neg , and \rightarrow from elements of Σ plus V and Λ . A Boolean term is affirmative if it is free of \neg and \rightarrow . A universal statement over Σ is a statement of the form $\forall\phi$, with ϕ a Boolean term. A theory Γ over Σ is a set of universal statements over Σ .*

Given two theories Γ and Γ' over Σ , we say that Γ *entails* Γ' , in symbols, $\Gamma \vdash \Gamma'$, if Γ entails Γ' by any sound and complete inference calculus for first-order predicate logic. The theories Γ and Γ' are said to be *equivalent* if they entail each other.

³ The results of Section 2 and 3 are to a large part adapted from [11].

Definition 2 (Conditional Form). A statement has conditional (or biconditional) form if it is of the form $\phi \preceq \psi$ (or $\phi \equiv \psi$), with ϕ and ψ affirmative. A theory over Σ has conditional (or biconditional) form if its statements are of this form. The conditional form is normal, if ϕ is purely conjunctive (or V) and ψ is purely disjunctive (or Λ).⁴ The conditional normal form is reduced, if ϕ and ψ have no atom in common.

Proposition 1. Every theory is logically equivalent to a theory in reduced conditional normal form and to one in biconditional form.

Proof. By applying the standard transformations of propositional logic, every Boolean predicate has an equivalent conjunctive normal form. So every universal statement is equivalent to a finite set of statements of the form $\phi \preceq \psi$, with ϕ, ψ affirmative. Finally notice that $\phi \preceq \psi$ is equivalent to $\phi \equiv \phi \wedge \psi$.

Definition 3 (Horn/Simple Inheritance/Exclusion). A conditional statement $\phi \preceq \psi$ is a Horn statement if ϕ and ψ are free of disjunctions; it is a simple inheritance statement if ϕ and ψ are atomic; it is an exclusion statement if ψ is Λ . A Horn theory is a theory consisting of Horn statements, etc.

Remark 1 (Nonredundant Basis). A nonredundant basis of a theory Γ is a minimal subset of Γ that entails Γ . For finite Σ , [3] presents a construction of a nonredundant basis, which generalizes the approach of [6].

Interpretations and models of theories are defined as usual in standard first-order predicate logic.

Definition 4 (Interpretation/Satisfaction). A (set-valued) interpretation of Σ consists of a universe U and an interpretation function M from Σ to $\wp(U)$. The function M uniquely corresponds to a satisfaction relation \models from U to Σ , with $x \models p$ iff $x \in M(p)$.

An interpretation M can be inductively extended to all Boolean terms by $M(V) = U$, $M(\Lambda) = \emptyset$, $M(\phi \wedge \psi) = M(\phi) \cap M(\psi)$, $M(\neg\phi) = U \setminus M(\phi)$, etc.

Definition 5 (Truth/Model). A statement $\forall\phi$ is true with respect to an interpretation if ϕ is satisfied by all elements of the universe. A model of a theory Γ over Σ is an interpretation of Σ with respect to which all statements of Γ are true.

Definition 6 (Specialization). Given an interpretation M of Σ with universe U and two elements x and y of U , then x is specialized by y (with respect to M), in symbols, $x \sqsubseteq_M y$, if y satisfies every element of Σ that is satisfied by x .⁵

The specialization relation \sqsubseteq is reflexive and transitive, i.e. a preorder. If \sqsubseteq is antisymmetric and thus a partial ordering, we say that the interpretation satisfies *identity of indiscernibles*. It should be noticed that by stressing the importance of the specialization relation, we distinguish affirmative terms from Boolean terms in general. For one shows easily by term induction that $x \sqsubseteq y$ iff y satisfies every affirmative term over Σ that is satisfied by x .

⁴ Notice that a statement in normal conditional form is essentially one in *clausal* form.

⁵ In case M is clear from context, we drop the subscript.

2.2 Information Domains

We now show how to associate with each theory a *canonical model*. The ordered universe of that model, called the *information domain* of the theory, can be regarded as a conceptual hierarchy induced by the theory. We give a characterization of how the structure of these hierarchies depends on the type of the theory, and vice versa.

Definition 7 (Canonical Interpretation/Model). *The canonical interpretation of Σ has universe $\wp(\Sigma)$ and takes each $p \in \Sigma$ to the set $\{X \subseteq \Sigma \mid p \in X\}$, that is, $X \vDash p$ iff $p \in X$. Given a theory Γ over Σ , let $C(\Gamma)$ be the set of all $X \subseteq \Sigma$ which, under the canonical interpretation, satisfy ϕ for every statement $\forall \phi$ of Γ . The canonical model $M(\Gamma)$ of Γ takes $p \in \Sigma$ to $\{X \in C(\Gamma) \mid p \in X\}$.*

We call the elements of $C(\Gamma)$ the *consistently Γ -closed subsets* of Σ . Specialization on $C(\Gamma)$ is set inclusion and hence a partial order. Adapting the terminology of [2], we refer to $C(\Gamma)$, partially ordered by specialization, as the *information domain* of Γ .

For each interpretation M of Σ with universe U let ε_M be the function from U to $\wp(\Sigma)$ that takes x to $\{p \in \Sigma \mid x \vDash_M p\}$. By definition of specialization, $x \sqsubseteq y$ iff $\varepsilon_M(x) \subseteq \varepsilon_M(y)$. So ε_M is an order embedding of U into $\wp(\Sigma)$ if M satisfies identity of indiscernibles. Moreover, it follows by term induction that $x \vDash_M \phi$ iff $\varepsilon_M(x) \vDash \phi$. Consequently, if M is a model of a theory Γ then ε_M is a homomorphism of models from M to $M(\Gamma)$. The canonical model is thus the “largest” Γ -model satisfying identity of indiscernibles in the sense that every other such model M is embedded in $M(\Gamma)$ via ε_M (see also [12]). Another consequence is that $M(\Gamma)$ is *universal* in the sense that a statement is true in $M(\Gamma)$ iff it is true in all models of Γ , i.e., iff it is entailed by Γ .

Depending on the class of Γ , the information domain $C(\Gamma)$ can be characterized as a subset system as follows (see [11] for a proof):

Theorem 1. *If a theory Γ over a finite set Σ belongs to one of the classes listed on the left of Table 1 then its information domain $C(\Gamma)$ is closed with respect to the properties listed in the same row on the right. Conversely, if a subset system \mathcal{U} over Σ has closure properties that are listed in the right column then \mathcal{U} is the information domain of a theory over Σ of the corresponding class on the left.*

Remark 2. The finiteness of Σ in Theorem 1 is essential. If Σ is infinite then, e.g., closure with respect to directed union has to be added in the case of Horn theories. Moreover, it should be noted that the information domains of the various classes of theories can also be characterized order-theoretically. The information domains of Horn theories, for instance, are precisely the *Scott domains*. For both topics see [10, 11].

Definition 8 (Canonical Theory). *Let \mathcal{C} be a class of statements over Σ and \mathcal{U} a subset system over Σ . The canonical \mathcal{C} -theory $\Gamma_{\mathcal{C}}(\mathcal{U})$ associated with \mathcal{U} is the set of all \mathcal{C} -statements $\forall \phi$ such that $\forall X \in \mathcal{U} (X \vDash \phi)$.*

Class of Γ	Closure properties of $C(\Gamma)$
unrestricted	none
Horn	nonempty intersection
Λ -free Horn	intersection
simple inheritance	intersection + union
exclusion	subsets + bounded union
binary exclusion	subsets + pairwise bounded union
simple inheritance + exclusion	nonempty intersection + bounded union

Table 1. Relationship between Γ and $C(\Gamma)$

Let us say that \mathcal{U} is \mathcal{C} -*definable* if \mathcal{U} is the information domain of a \mathcal{C} -theory (which is the case, for instance, if \mathcal{C} is the class of Horn statements and \mathcal{U} is closed with respect to nonempty intersection). It is easy to see that \mathcal{U} is \mathcal{C} -definable just in case $\mathcal{U} = C(\Gamma_{\mathcal{C}}(\mathcal{U}))$. In general, $\Gamma_{\mathcal{C}}(\mathcal{U})$ is the *least \mathcal{C} -definable subset system containing \mathcal{U}* . Consequently, by Theorem 1:

Theorem 2. *The information domain of $\Gamma_{\mathcal{C}}(\mathcal{U})$ is the closure of \mathcal{U} with respect to the properties of Table 1 that correspond to class \mathcal{C} .*

3 Induction of Theories from Data

3.1 Complete Theories of Classification Tables

Consider the situation that a certain set U of objects is classified with respect to a set Σ of properties (or attributes). In other words, we are given a satisfaction relation \models from U to Σ , i.e. an interpretation function M from Σ to $\wp(U)$. The triple $\langle U, \Sigma, \models \rangle$ is henceforth called a *classification table*. Given a classification table one can ask for a theory that explains the data. To make this precise, we need to fix the type of theory we are interested in. For example, one can ask for a simple inheritance theory with or without exclusions, a Horn theory with or without Λ , or a theory in general.

Definition 9 (Complete Theory). *Let \mathcal{C} be a class of statements over Σ and M an interpretation of Σ . A \mathcal{C} -theory Γ is a complete \mathcal{C} -theory of M if, first, every statement of Γ is true with respect to M , i.e. M is a model of Γ , and, second, Γ entails every \mathcal{C} -statement that is true in M .*

It is an immediate consequence of definitions that a complete \mathcal{C} -theory of M is unique up to equivalence. Moreover, there is a trivial way to get a complete theory by taking the set $\Gamma_{\mathcal{C},M}$ of all \mathcal{C} -statements that are true with respect to M .

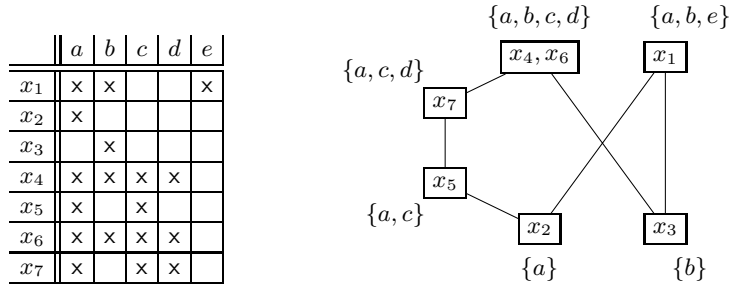


Fig. 1. Classification table and induced specialization order

Let us explore more closely the relation between a given classification table and the information domain of its complete \mathcal{C} -theory. As shown in Section 2, a classification table, i.e. a satisfaction relation \models from U to Σ , determines a specialization relation \sqsubseteq on U . Let \mathcal{U}_M be the image $\{\varepsilon_M(x) \mid x \in U\}$ of the (pre)order-preserving function ε_M from U to $\wp(\Sigma)$ that takes $x \in U$ to $\{p \in \Sigma \mid x \models p\}$. In general ε_M is not one-to-one because there is no guarantee of *identity of indiscernibles*, i.e. different elements of U may satisfy exactly the same elements of Σ . We have $\mathcal{U}_M \simeq U/\sim$, with $x \sim y$ iff $\varepsilon_M(x) = \varepsilon_M(y)$.

Now notice that the canonical \mathcal{C} -theory $\Gamma_{\mathcal{C}}(\mathcal{U}_M)$ associated with \mathcal{U}_M coincides with $\Gamma_{\mathcal{C},M}$, because $\varepsilon_M(x) \models \phi$ iff $x \models \phi$. So we can apply Theorem 2 to characterize the information domain of a complete \mathcal{C} -theory of M . For instance, if Γ is a complete Horn theory of M then $C(\Gamma)$ is the closure of \mathcal{U}_M with respect to nonempty intersection; similarly, if Γ is a complete simple inheritance theory of M then $C(\Gamma)$ is the closure of \mathcal{U}_M with respect to intersection and union.

Example 1. Let Σ be $\{a, b, c, d, e\}$. Suppose U consists of the seven elements x_1, x_2, \dots, x_7 which are classified according to the table of Figure 1. In addition, the figure shows the specialization order on U/\sim induced by the given classification table (where x_4 and x_6 are indiscernible, i.e. $x_4 \sim x_6$), as well as the corresponding subset system \mathcal{U}_M over Σ . Figure 2 provides an overview of the information domains of several complete \mathcal{C} -theories of M , with varying \mathcal{C} . At the top of the Figure there is the information domain of a complete simple inheritance theory of M ; it is the closure of \mathcal{U}_M with respect to intersection and union. A (nonredundant) complete simple inheritance theory of M is given by the statements $d \preceq c$, $c \preceq a$, $e \preceq a$, and $e \preceq b$. The diagram below the top on the left depicts the closure of \mathcal{U}_M with respect to intersection of nonempty subsets and union of bounded subsets. It is the information domain of the extension of the above simple inheritance theory by the exclusion statement $c \wedge e \preceq \perp$. Addition of the Horn statement $b \wedge c \preceq d$ further weakens the closure properties of the associated information domain. If the statement $b \wedge c \preceq d$ is added to the simple inheritance theory before the exclusion statement $c \wedge e \preceq \perp$, the resulting effect on the respective information domains is as depicted by the right branch

of Figure 2. Finally, adding the statements $V \preceq a \vee b$ and $a \wedge b \preceq c \vee e$ leads to a complete theory of M , whose information domain consequently is \mathcal{U}_M .

3.2 Formal Concept Analysis

Formal Concept Analysis ([4]) associates with each classification table $\langle U, \Sigma, \models \rangle$ a (complete) lattice of formal concepts. (In the terminology of *Formal Concept Analysis*, a classification table is called a *formal context*.) A *formal concept* of $\langle U, \Sigma, \models \rangle$ is a pair $\langle V, X \rangle$ consisting of a set $V \subseteq U$ of objects (the *extent*) and a set $X \subseteq \Sigma$ of attributes (the *intent*) such that X is the set of those attributes that are shared by all objects of V , whereas V consists of all objects that have all attributes of X . So $\langle V, X \rangle$ is a formal concept just in case $V^\blacktriangleright = X$ and $X^\blacktriangleleft = V$, where

$$\begin{aligned} V^\blacktriangleright &= \{p \in \Sigma \mid \forall x \in V (x \models p)\} = \bigcap \{\varepsilon_M(x) \mid x \in V\}, \\ X^\blacktriangleleft &= \{x \in U \mid \forall p \in X (x \models p)\} = \bigcap \{M(p) \mid p \in X\}, \end{aligned}$$

and M is the interpretation function associated with the classification table. Clearly $\langle (V^\blacktriangleright)^\blacktriangleleft, V^\blacktriangleright \rangle$ is a formal concept for each $V \subseteq U$. Furthermore, every formal concept is of the form $\langle (V^\blacktriangleright)^\blacktriangleleft, V^\blacktriangleright \rangle$. Within Formal Concept Analysis, a concept $\langle V_1, X_1 \rangle$ is said to be a *subconcept* of a concept $\langle V_2, X_2 \rangle$, notation: $\langle V_1, X_1 \rangle \leq \langle V_2, X_2 \rangle$, iff $V_1 \subseteq V_2$ or, equivalently, iff $X_1 \supseteq X_2$. The set of formal concepts ordered by \leq forms a complete lattice, the so-called *concept lattice*. Notice that the subconcept ordering is reverse to the specialization ordering.

By definition, the set $\{V^\blacktriangleright \mid V \subseteq U\}$ of intents is the closure of $\{\varepsilon_M(x) \mid x \in U\}$, i.e. of \mathcal{U}_M , with respect to intersection. Hence Theorem 2 gives us the following characterization:

Theorem 3. *There is an order-reversing one-to-one correspondence between the concept lattice of a (finite) classification table and the information domain of the complete Λ -free Horn theory of that table.*

The diagram of Figure 3 summarizes the relation between (finite) concept lattices and information domains of Λ -free Horn theories. For the rest of the paper, we take up the convention of Formal Concept Analysis to graphically depict more special elements *below* less special ones.

4 Application: Classification of Amino Acids

4.1 Conceptual Hierarchies via Formal Concept Analysis

The automatic induction of hierarchies is desirable both from a practical and a theoretical point of view. On the one hand, it makes the processing of large amounts of data possible and provides fast results. On the other hand, it is of theoretical interest to compare an automatically induced classification with existing descriptions, in order to reveal the assumptions made by the human

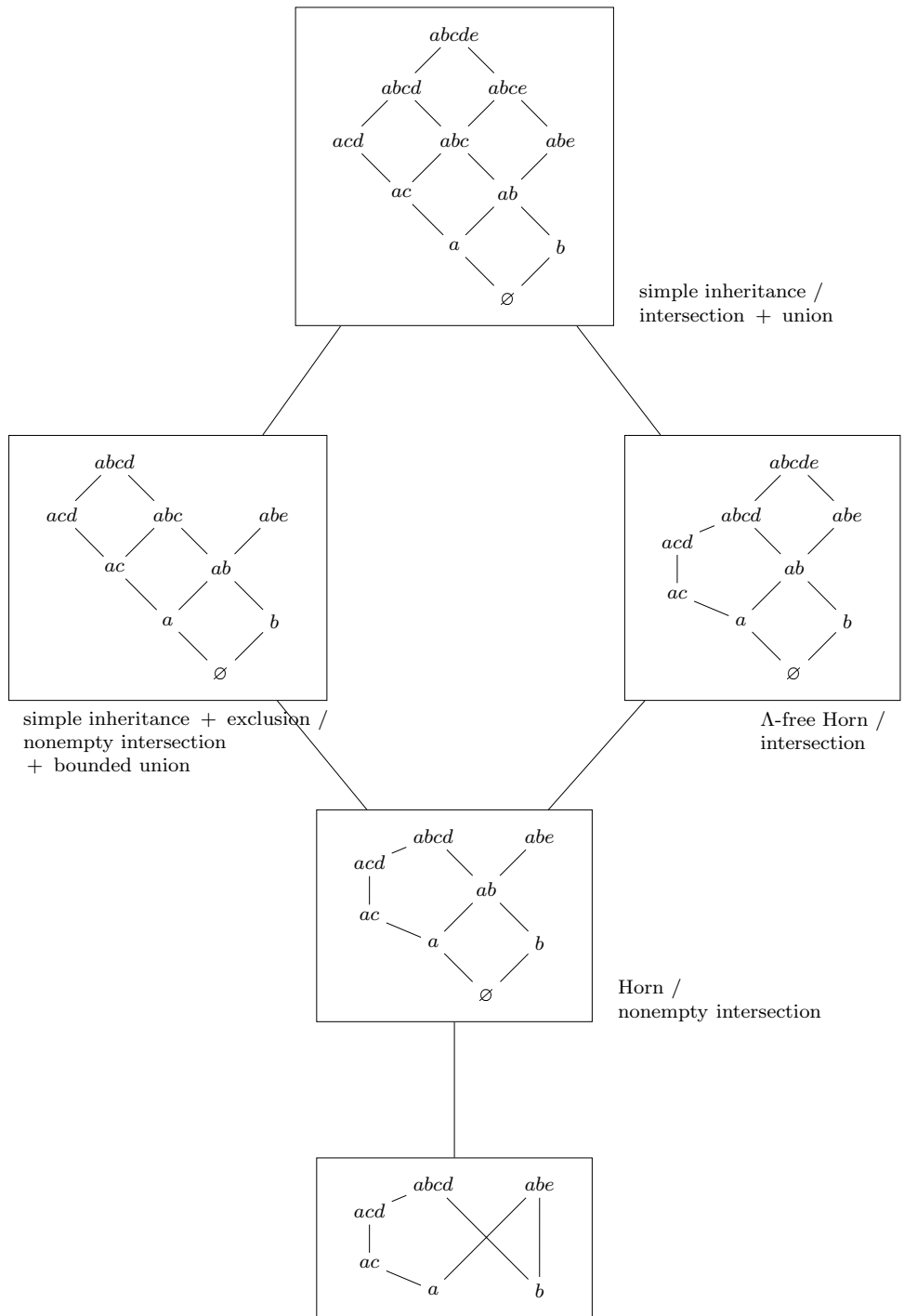


Fig. 2. Information domain of complete \mathcal{C} -theory with varying \mathcal{C}

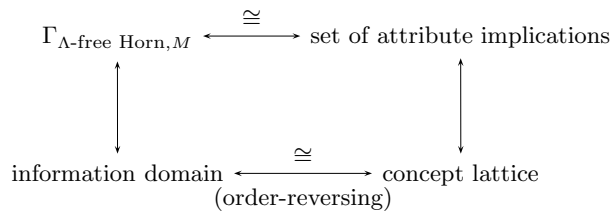


Fig. 3. Information domain approach vs. Formal Concept Analysis

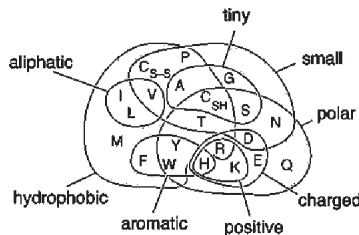


Fig. 4. Venn diagram taken from [7]

expert. Furthermore, an automatically induced hierarchy can guide the human expert in analyzing new data. To this end, the induced hierarchy should exhibit as much of the implicitly given information as possible, and the original data should always be reconstructible from the induced hierarchy.

We discuss these issues by means of an example from an amino acid data base. Proteins are essentially built from 20 different amino acids that are specified by the genetic code. The amino acids can be classified by physico-chemical properties. Often this classification is represented in form of a Venn diagram (see Figure 4). Table 2 shows a classification table of those nine polar amino acids which tend to be at the surface of proteins; they are characterized by 17 physico-chemical properties.⁶ It covers the properties: acidic, basic, neutral, charged, positive, negative, polar, hydrophobic, aliphatic, aromatic, buried, surface, acyclic, cyclic, large, medium, small.⁷ Details about the properties of amino acids can be found in any introductory book on biochemistry (e.g. [9]).

Classifying the example data with Formal Concept Analysis results in the concept lattice shown in Figure 5.⁸ As usual, only the attribute and the object concepts are labeled.⁹ The concept lattice represents a monotonic multiple

⁶ See http://www.rrz.uni-hamburg.de/biologie/b_online/d16/16j.htm

⁷ “Buried”, in contrast to “surface”, classifies amino acids that tend to be buried inside proteins. The three attributes of size refer to the number of atoms in the molecule.

⁸ The drawing was done by the software tool “Concept Explorer” written by Sergey Yevtushenko; see <http://www.sourceforge.net/projects/conexp>.

⁹ The *attribute concept* associated with an attribute p is the greatest concept whose intent contains p and the *object concept* of an object x is the smallest concept whose extent contains x .

	acidic	basic	neutral	charged	negative	positive	polar	hydrophobic	aliphatic	aromatic	buried	surface	acyclic	cyclic	large	medium	small
Arg:R		×		×		×	×					×	×		×		
Asn:N			×				×					×	×			×	
Asp:D	×			×	×		×					×	×			×	
Glu:E	×			×	×		×					×	×		×		
Gln:Q			×				×					×	×		×		
His:H		×	×	×		×	×			×		×		×	×		
Lys:K		×		×		×	×					×	×		×		
Ser:S			×				×					×	×				×
Thr:T			×				×					×	×			×	

Table 2. Classification of several amino acids by their physico-chemical properties

inheritance hierarchy, where a node inherits all the attributes labeled to its super-nodes. Notice that conflicting attributes cannot be inherited, since the hierarchy is constructed on the base of the subset relation of concept intents. Compared to Venn diagrams, which are frequently used in biochemistry books, the hierarchical representation by the concept lattice is in our view easier to access for human beings. Questions like ‘what are the common attributes of two amino acids?’ can be answered simply by looking at the smallest node above the amino acids.

Let us focus more closely on the six unlabeled nodes of the example concept lattice. The nodes of an inheritance hierarchy have essentially two roles: first, they can introduce new information, which will be inherited by subnodes and second, they “collect” information from their supernodes and transmit it “bundled up” to their subnodes. Unlabeled nodes are nodes which only perform information bundling and not information introduction. Nodes that do not bundle up information are necessarily labeled. Altering the hierarchy by varying the underlying theory which models the data of the classification table changes the proportion between the information introducing and the information bundling nodes.

4.2 Varying the Theory Class

Among the different hierarchical representations of a given data set there is none which is optimal in every respect. Rather, the question is to find the most appropriate representation depending on the task for which the hierarchy is built. Two criteria must be met by any reasonable representation: it must be complete and consistent with respect to the data. Furthermore, a good representation is

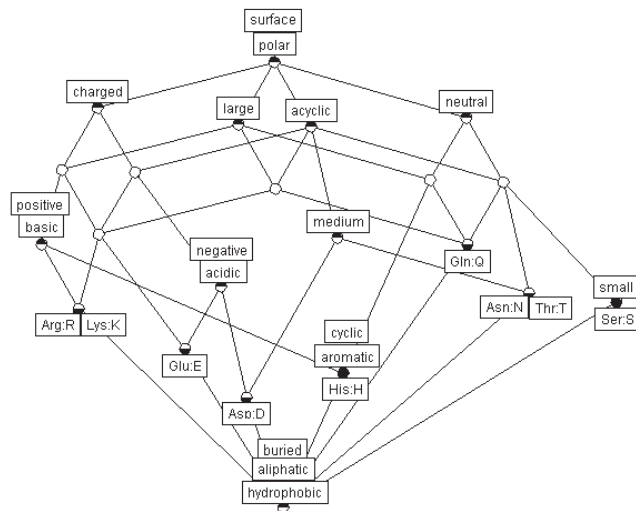


Fig. 5. Concept lattice corresponding to Table 2

maximally informative, maximally compact, and avoids redundancies by capturing generalizations. Unfortunately, it is not possible to construct an inheritance hierarchy which is optimal with respect to each of these criteria.

What does it mean to say that a hierarchical representation is maximally informative? In principle, every hierarchy which is consistent and complete with respect to the data is equally informative in the sense that the original classification table can be reconstructed from the hierarchy. But the information domain of the complete (unrestricted) theory of the example data consists only of seven attribute sets which are pairwise incommensurable. They correspond to the seven elements of the concept lattice of Figure 5 that are immediately above the bottom element. For the observer the resulting flat hierarchy is less informative than the concept lattice, although from the viewpoint of the underlying theories, the Horn theory is a subtheory of the unrestricted one and therefore less informative. Switching to a less restricted theory class and looking at the rules gives the possibility to gain extra information. For example, the complete Horn theory of our example data only expresses that amino acids which are “positive” (or “negative”) are also “charged”. But it fails to point out the connection that every amino acid which is “charged” is either “positive” or “negative”.

Since we are interested in the induction of hierarchical representations, we record that hierarchies differ with regard to the amount of information they exhibit explicitly. If the hierarchy is designed to be viewed by human beings it should maximize this amount of information. The compactness of a network can be measured in several respects, but in what follows we will only look at the number of nodes. The compactness criterion clearly favors the network of the unrestricted theory. A good representation avoids redundancy by capturing

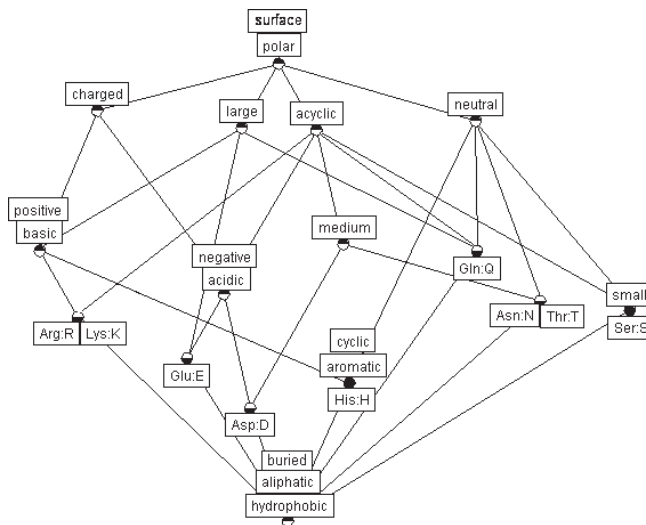


Fig. 6. AOC-poset corresponding to Table 2

generalizations. In the case of the flat information domain of the unrestricted theory no generalizations are captured and therefore, most attributes have to be stated more than once (e.g. “charged”). In other words, the unrestricted theory leads to “overfitting”. In the concept lattice (see Figure 5) all generalizations are captured and every attribute and every object occurs exactly once; such a representation is said to be free of redundancy.

Is there any representation that has this desirable property but is more compact than the concept lattice? According to the results of Section 2 and 3, such a representation is the information domain of an extension of a complete \mathcal{A} -free Horn theory by disjunctive rules that are consistent with respect to the data. Recalling the two different roles of nodes in inheritance hierarchies we can dispense with the six nodes which only bundle up information. This results in the inheritance network in Figure 6, which is the partially ordered set of the attribute and object concepts (*AOC-poset*).¹⁰ A theory of the data whose information domain coincides with the AOC-poset can be obtained by taking all rules of the complete \mathcal{A} -free Horn theory and by adding for each concept to be pruned from the concept lattice a rule whose premise is the conjunction of the intent X of the concept and whose conclusion is the disjunction of the conjunctions of its subconcept intents minus X . For example, to eliminate the node with intent {acyclic, neutral} (the unlabeled node on the right of Figure 5), we can add the rule

$$\text{acyclic} \wedge \text{neutral} \preceq \text{small} \vee \text{medium} \vee \text{large},$$

¹⁰ AOC-posets are also known as *pruned concept hierarchies* [5]. The present terminology has been introduced in [15].

which is clearly true with respect to Table 2. Then $\{\text{acyclic, neutral}\}$ is not consistently closed with respect to the extended theory and thus not an element of its information domain. Notice also that by definition the rule does not eliminate any other nodes.

Compared to the concept lattice, the AOC-poset is more compact and also free of redundancy. But it is not as informative as the concept lattice, since the information about common attributes is not captured in single nodes anymore. In the worst case, the AOC-poset has only two levels: the level of the attribute nodes and the level of the object nodes. This happens if, first, all objects intents and, second, all attribute extents are pairwise incomparable with respect to set inclusion. Nevertheless, the AOC-poset is more informative than the flat hierarchy since it simplifies the access to the information to which objects an attribute applies and it shows the hierarchical relations between the attributes.

The number of nodes in an AOC-poset is bounded by the sum of the number of attributes and the number of objects. In realistic data sets the difference in compactness between AOC-posets and concept lattices can be dramatic. For instance, take the lexical database CELEX, compiled by the Dutch Center for Lexical Information, which consists of three large electronic databases and provides users with detailed English, German and Dutch lexical data. The German database, which serves us as a test database, holds 51.728 lemmas with 365.530 corresponding word forms. Focusing at the stored derivational information of German lemmas, the number of nodes in the corresponding concept lattice is greater than 72.000, whereas the number of nodes in the AOC-poset is less than 4.000. (The underlying classification table consists of 9.567 objects and 2.032 attributes.) Hence, switching to the AOC-poset reduces the memory requirements. Moreover, since the AOC-poset is just the partial order of the attribute and object concepts, there is an efficient construction algorithm. To summarize, compared to concept lattices, AOC-posets provide a very simple method to induce redundancy-free inheritance hierarchies from huge databases. Inference tasks, however, are better supported by concept lattices, due to the explicit representation of shared attributes.

Having discussed the case of adding rules to a complete Horn theory, it remains to consider the omission of rules. Switching to the complete simple inheritance theory without exclusions seems to be overdone, because for the example data of Table 2 the resulting lattice has 100 concepts. Since many of the attributes of the example are incompatible, it makes sense to take the complete simple inheritance theory with exclusions instead. The corresponding hierarchy has 25 elements, witness Figure 7, and is hence less compact than the concept lattice. The simple inheritance theory is weaker than the one describing the AOC-poset or the concept lattice; it is thus more likely that a new object can be inserted without serious changes to the structure of the lattice.¹¹

¹¹ In biology, as well as in other sciences, hierarchical classifications are often presented in form of *taxonomic trees*. In [13], it is indicated how to “cut out” classification trees from concept lattices by adding appropriate disjunctive rules.

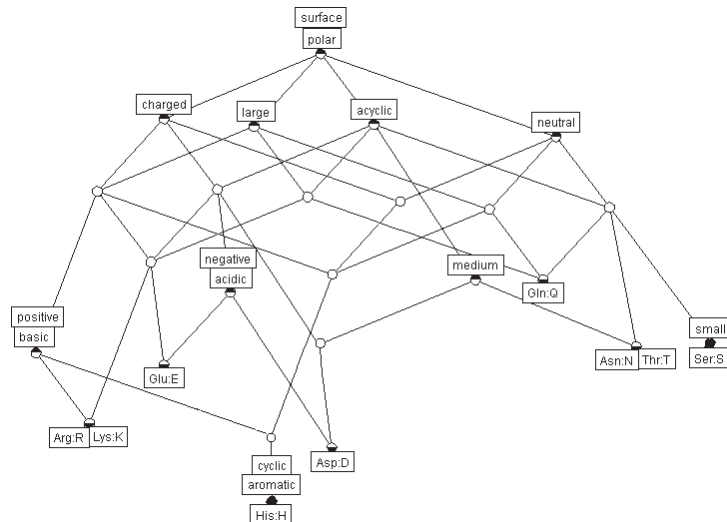


Fig. 7. Simple inheritance hierarchy with exclusions

5 Outlook

A natural task to pursue is to analyze the presented approach from the viewpoint of *machine learning* (e.g. [8]). The problem of inducing theories from classification tables can be characterized as follows: A given class \mathcal{C} of theories constitutes the *hypothesis space* \mathcal{H} of the learning problem, whereas the *version space* with respect to \mathcal{H} and M consists of all \mathcal{C} -theories with model M . The commitment to statement type \mathcal{C} determines the *inductive bias*: one can fit the data only as well as \mathcal{C} permits. On the other hand, if \mathcal{C} is too expressive, *overfitting* can occur: the induced theory explains the given data perfectly but does not allow generalizations. In addition to such general considerations, it seems worth to spell out the precise relation to *concept learning* and *inductive logic programming*.

A possible application of the presented approach is to allow disjunctive rules in attribute exploration tasks.¹² As discussed in Section 4.2, the problem is to avoid accepting too many disjunctive rules, since otherwise, in the case of incommensurable objects the exploration would always end in a flat hierarchy. One way to prevent this could be to introduce two steps: first, the standard attribute exploration is performed and second, each concept which is not yet an attribute or an object concept is tested to determine whether there is any object in the universe to which exactly the attributes of its intent apply. If so, the object is added to the context and if not, a disjunctive rule is added which

¹² Attribute Exploration allows to determine a typical set of objects given a set of classifying attributes. An interactive procedure presents implications to the user, who can either accept them or deny them by giving a counterexample. For more details see [3].

excludes the concept from the information domain. In an exploration tool the concept could be tested by presenting the corresponding disjunctive rule (see Section 4.2) and asking if there is any known counter example.

Furthermore it would be interesting to explore possible ways to automatically shift from one theory to another, based on parameters like compactness monitored during incremental construction of the inheritance hierarchy.

References

1. Samson Abramsky. Domain theory in logical form. *Journal of Pure and Applied Logic*, 51:1–77, 1991.
2. Manfred Droste and Rüdiger Göbel. Non-deterministic information systems and their domains. *Theoretical Computer Science*, 75:289–309, 1990.
3. Bernhard Ganter. Attribute exploration with background knowledge. *Theoretical Computer Science*, 217(2):215–233, 1999.
4. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, Berlin, 1999.
5. Robert Godin, Hafedh Mili, Guy W. Mineau, Rokia Missaoui, Amina Arfi, and Thuy-Tien Chau. Design of class hierarchies based on concept (Galois) lattices. *Theory and Application of Object Systems*, 4(2):117–134, 1998.
6. J.-L. Guigues and Vincent Duquenne. Familles minimales d’implications informatives resultant d’un tableau de données binaires. *Mathématiques et Sciences Humaines*, 95:5–18, 1986.
7. Rainer Merkl and Stephan Waack. *Bioinformatik Interaktiv – Algorithmen und Praxis*. Wiley-VCH, Weinheim, 2003.
8. Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
9. David L. Nelson and Michael M. Cox. *Lehninger Principles of Biochemistry*. Worth Publishers, New York, 3rd edition, 2000.
10. Rainer Osswald. Classifying classification. In Geert-Jan Kruijff, Larry Moss, and Dick Oehrle, editors, *Proceedings of the Joint Conference on Formal Grammar and Mathematics of Language (FG/MOL-01)*, Electronic Notes in Theoretical Computer Science 53, 2001.
11. Rainer Osswald. *A Logic of Classification – with Applications to Linguistic Theory*. PhD thesis, FernUniversität Hagen, Praktische Informatik VII, 2002.
12. Rainer Osswald. Generic ontology of linguistic classification. In Benedikt Löwe, Wolfgang Malzkorn, and Thoralf Räscher, editors, *Foundations of the Formal Sciences II – Applications of Mathematical Logic in Philosophy and Linguistics*, Trends in Logic, pages 203–212. Kluwer, Dordrecht, 2003.
13. Rainer Osswald and Wiebke Petersen. Induction of classifications from linguistic data. In *Proceedings of the ECAI-Workshop on Advances in Formal Concept Analysis for Knowledge Discovery in Databases*, Lyon, July 23 2002.
14. Wiebke Petersen. *Induction of linguistic classifications – supported by Formal Concept Analysis*. PhD thesis, Heinrich-Heine-Universität Düsseldorf, Institut für Sprache und Information, in preparation.
15. Wiebke Petersen. A set-theoretic approach for the induction of inheritance-hierarchies. In Geert-Jan Kruijff, Larry Moss, and Dick Oehrle, editors, *Proceedings of the Joint Conference on Formal Grammar and Mathematics of Language (FG/MOL-01)*, Electronic Notes in Theoretical Computer Science 53, 2001.
16. Steven Vickers. *Topology via Logic*. Cambridge Tracts in Theoretical Computer Science 5. Cambridge University Press, Cambridge, 1989.