A Set-Theoretical Approach for the Induction of Inheritance Hierarchies

Wiebke Petersen [1]

Institut für Sprache und Information
Abteilung Computerlinguistik
Heinrich Heine Universität
Düsseldorf, Germany

Abstract

An approach for the automatic construction of inheritance hierarchies is presented. It is based on the strict set-theoretical point of view in the mathematical theory of Formal Concept Analysis. The resulting hierarchies are concept lattices. An extension of the approach to the induction of nonmonotonic inheritance networks is also discussed. It turns out that the main ideas of Formal Concept Analysis, i. e. the formal context, the concept lattice and the set of implications, provide three different ways of looking at the data to be represented, each of which provides a different way to solve problems of knowledge representation.

1   Introduction

Since in modern linguistic theories (e.g. HPSG) more and more information is regarded as being lexical, the lexicon grows in size and complexity. Much of the redundancy of lexical information can be eliminated by factoring out those properties which are shared by a set of signs and representing them in an inheritance hierarchy. In this way redundancy is avoided, generalizations are captured and the required memory is minimized. In view of the size of realistic lexicons the manual construction of such hierarchies is time consuming and error-prone. Therefore techniques for automatically acquiring lexical knowledge are desirable. There are a number of approaches for automatically updating a lexicon by inserting new objects in an existing hierarchy: this task is known as the "insertion problem" or "learning unknown words" (e.g. Light 1994 [12], Kilbury et al. 1994 [9]). In the framework of linguistic lexicons there are mainly two non-incremental approaches. Barg presents an algorithm for inducing hierarchies represented in DATR (cf. Barg 1996 [1,2]), which is a

---

[1] Email: petersew@uni-duesseldorf.de

widespread formalism for representing lexical information in nonmonotonic multiple inheritance networks (cf. Evans & Gazdar 1996 [5]). Here the user has to supply criteria for the selection of intermediate hypotheses in order to restrict the search space. Depending on the criteria chosen, the resulting DATR theory can be very compact. The other approach, presented by Sporleder, involves an algorithm that infers decision trees for the construction of feature-structure hierarchies (cf. Sporleder 1999 [18], Lüngen & Sporleder 1999 [13]).

The present paper presents a different approach for the automatic acquisition of lexical knowledge from unstructured data. The main innovation is that the mathematical theory of *Formal Concept Analysis* (cf. Ganter & Wille 1999 [6]) is used to extract the monotonic inheritance relations which are inherently given by the data. This approach is implemented and has been successfully tested by computing a hierarchy of the derivational information of English resp. German lemmas in the lexical database CELEX.[2]

2 Inducing monotonic hierarchies from unstructured data

2.1 An approach for inducing compact hierarchies

Most problems involving the learning of hierarchies can be reduced to the following problem: Assume a given set of object-attribute pairs,[3] which is to be organized in an inheritance network; which inheritance hierarchy captures the inherent relations between these data best? There are at least two general demands on representations of the given data: they have to be consistent and complete with respect to the data. Furthermore, a good representation avoids redundancy by capturing generalizations; a representation is said to be redundancy-free if every attribute and every object is stated exactly once. In addition one could require the number of nodes in the constructed network to be minimal. Some theories which use inheritance network representations require the networks to have special properties; for instance they could allow only monotonic or single inheritance relations, or they could demand that the network form a join semilattice, so that every set of nodes has a least upper bound.

Table 1 gives an example data-set consisting of the inflectional paradigms and the genus information of seven German nouns (*Herr* 'mister', *Friede* 'peace', *Staat* 'state', *Hemd* 'shirt', *Farbe* 'color', *Onkel* 'uncle', *Ufer* 'bank'/

---

[2] CELEX is a large electronic database compiled by the Dutch Center of Lexical Information (http://www.kun.nl/celex/)

[3] A feature-value pair applying to an object can be seen as one of its attributes. This is a strong simplification, since the appropriateness of a feature for an object cannot be captured anymore. A possible way out is to increase the number of attributes by adding attributes of the kind "feature x is appropriate here."

	gender	sing nom	sing gen	sing dat	sing acc	plur nom	plur gen	plur dat	plur acc
Herr	masc	*	*_n	*_n	*_n	*_n	*_n	*_n	*_n
Friede	masc	*	*_ns	*_n	*_n	*_n	*_n	*_n	*_n
Staat	masc	*	*_s	*	*	*_n	*_n	*_n	*_n
Hemd	neut	*	*_s	*	*	*_n	*_n	*_n	*_n
Farbe	fem	*	*	*	*	*_n	*_n	*_n	*_n
Onkel	masc	*	*_s	*	*	*	*	*_n	*
Ufer	neut	*	*_s	*	*	*	*	*_n	*

Table 1

input data: many-valued context

	gender: masc	gender: fem	gender: neut	sing nom:*	sing gen:*	sing gen:*_s	sing gen:*_n	sing gen:*_ns	sing dat:*	sing dat:*_n	sing acc:*	sing acc:*_n	plur nom:*	plur nom:*_n	plur gen:*	plur gen:*_n	plur dat:*_n	plur acc:*	plur acc:*_n
Herr	x			x			x			x		x		x		x	x		x
Friede	x			x				x		x		x		x		x	x		x
Staat	x			x		x			x		x			x		x	x		x
Hemd			x	x		x			x		x			x		x	x		x
Farbe		x		x	x				x		x			x		x	x		x
Onkel	x			x		x			x		x		x		x		x	x	
Ufer			x	x		x			x		x		x		x		x	x	

Table 2

input data: represented in a one-valued context

'shore').[4] The data is in the form of feature-value pairs associated with objects (in Formal Concept Analysis three-tupels of this type are called many-valued contexts) and has to be transformed to a binary relation of object-attribute pairs. Table 2 shows the transformed data (the one-valued context) which is to be organized in an inheritance hierarchy. It is composed of seven objects (lexemes), for each of which a subset of the total set of 19 attributes applies.

One natural way of structuring these data is to take the corresponding set of attributes for every object and order these sets with respect to the superset relation. A bottom element can be added to get a connected partial order. Since the example data are of such a form that none of the attribute sets

---

[4] Throughout the present paper '*' represents the root of the derived word form. For example, if the feature "sing dat" has the value "*_n" at the object "Friede", that means that the singular dative form of *Friede* is *Frieden*.
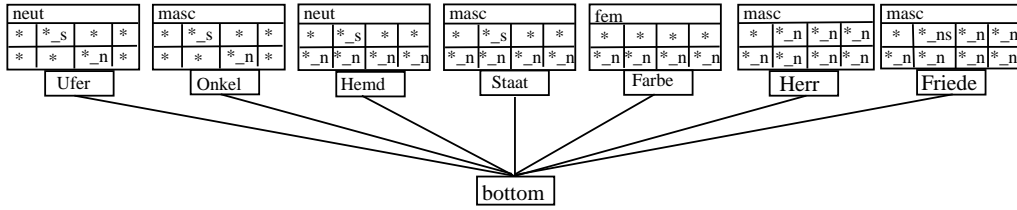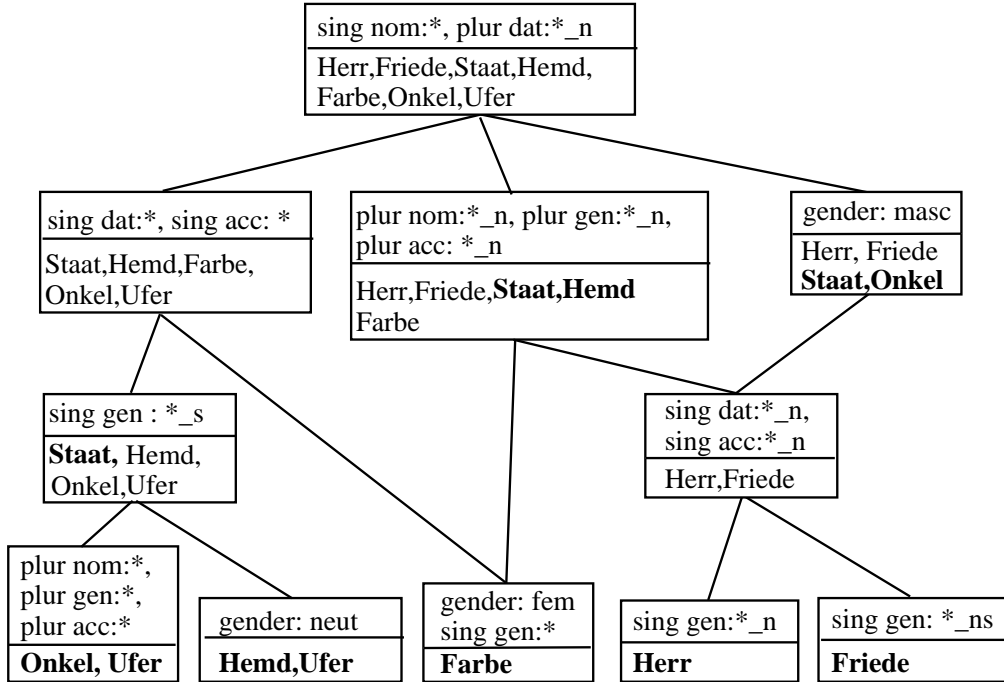
3

Fig. 1. hierarchy ordered by the attribute sets

Fig. 2. hierarchy ordered by the object sets

corresponding to the single objects constitutes a superset of another attribute set, this approach does not lead to any hierarchical relation at all. The outcome is the totally flat structure shown in figure 1; it shows the same range of redundancy as the input data.

The opposite way of structuring the data from table 2 by ordering the sets of objects which have an attribute in common leads to the hierarchy shown in figure 2 (object names are bold-printed if they cannot be inherited from lower nodes; the inheritance direction is bottom-up). This also is not free of redundancy, since the objects "Staat", "Hemd","Onkel" and "Ufer" have to be stated more than once in order to ensure that the attributes inherit the correct objects. This problem arises from the onesided extensional point of view that only takes the object sets into account.

A redundancy-free solution is obtained if one combines these two ways of structuring the data (see figure 3). The constructed network is a monotonic multiple-inheritance hierarchy. It is labeled in such a way that each object and
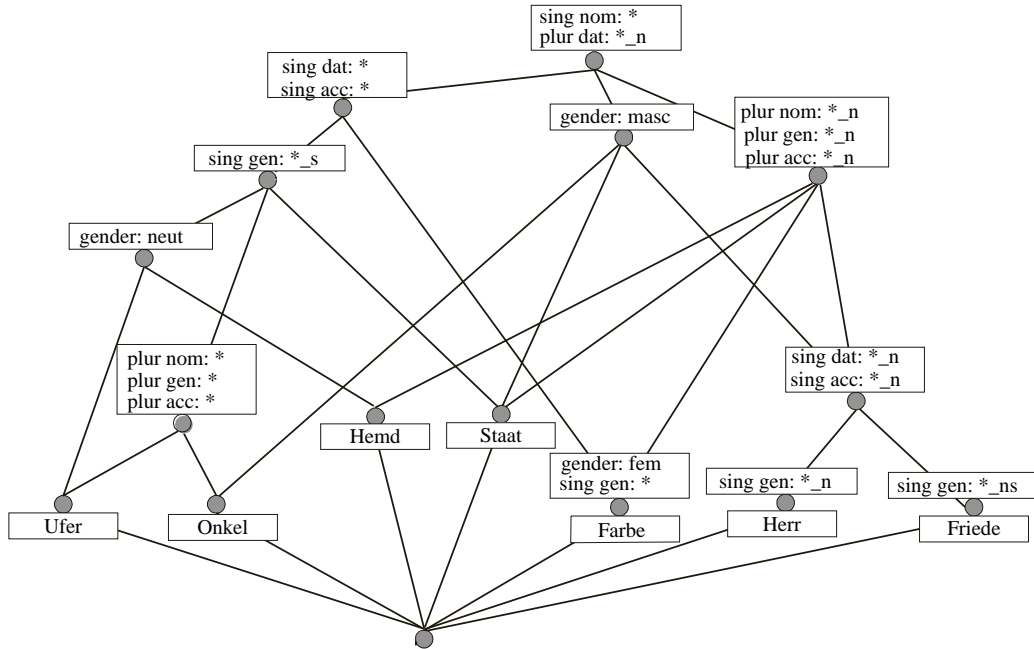
Fig. 3. partial order of the object and attribute concepts

each attribute name appears exactly once. This is achieved by attaching each attribute to the highest possible node and each object to the lowest possible; for more details see section 2.2. Inheritance hierarchies constructed in this way have the desirable property that the number of nodes extends the sum of the number of attributes and the number of objects maximally by two, and they are therefore very compact. Furthermore, they are data-consistent and complete, since a row of table 2 representing the properties of one object can be reconstructed by collecting all the attribute names labeling nodes above the node labeled with the object name. Analogously, a column corresponding to one attribute is given by the object names which can be found below the attribute.

Some theories require that inheritance networks be join semilattices, in which any two elements have a least upper bound (cf. Carpenter 1992 [3]). The partial order shown in figure 3 is not a semi-lattice, since for instance the nodes labeled "Hemd" and "Staat" do not have a unique least upper bound. The following section shows that even complete lattices can be induced by using Formal Concept Analysis, a set-theoretic approach which combines the extensional and intensional point of view in one theory.

2.2 Applying Formal Concept Analysis to induce inheritance lattices

Formal Concept Analysis is a mathematical theory which was especially designed to provide a formal model of knowledge as a tool for communication (cf. Zickwolf 1994 [22]). It aims at combining the advantages of a formal representation, like being machine-readable and processible, and a representation

5

that can be presented visually in such a way that it is readable for human beings. It has been applied to the following linguistic problems: meronymy (cf. Priß 1998 [17]), WordNet (cf. Priß 1998 [16]), semantics of speech-act-verbs (Großkopf & Harras 1999 [8]) and verb paradigms (Großkopf 1996 [7]). In the context of this paper it is only possible to list the main definitions of Formal Concept Analysis (FCA); for more details see Ganter & Wille 1999 [6].

FCA starts with the definition of a *formal context* K as a triple (G, M, I), consisting of a set of objects G, a set of attributes M and a binary *incidence relation* $I \subseteq G \times M$, where $(g, m) \in I$ means "the formal object g has the formal attribute m." Formal contexts are typically represented in cross tables (see table 2). For any subset of objects $A \subseteq G$, their set of common attributes is defined as $A' := \{m \in M | \forall g \in A : (g, m) \in I\}$. Analogously, the set of common objects for a subset $B \subseteq M$ of attributes is $B' := \{g \in G | \forall m \in B : (g, m) \in I\}$. A *formal concept* is a pair (A, B), with the properties $A = B'$ and $B = A'$, where A is called the *extent* and B the *intent* of the concept. The set of all formal concepts of a context is partially ordered by the subconcept-superconcept-relation: $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2$. It can be proved that the set of formal concepts together with this partial order form a complete lattice, called the *formal concept lattice* (see figure 4). In figure 4 only the attribute and the object concepts are labeled, where the *attribute concept* $\mu(m)$ associated with an attribute m is the greatest concept whose intent contains m, and analogously, the *object concept* $\gamma(g)$ associated with an object g is the smallest concept whose extent contains g. Labeled as in figure 4, a concept lattice can be seen as a monotonic multiple inheritance hierarchy. Since the hierarchy is constructed on the base of the subset relation, it is excluded that conflicting attributes are inherited from parent nodes. However, the inheritance is not orthogonal in the sense that parent nodes have disjoint sets of attributes. Since orthogonality is stipulated in some theories like HPSG (Pollard & Sag 1987 [15]), this requires special attention when FCA is used as a tool to construct hierarchies.

The inheritance network of figure 3 is the picture of the partial ordered set of the attribute and object concepts (*AOC-poset*); it is therefore a sub-poset of the concept lattice of figure 4. Compared with the AOC-poset, the number of nodes increases in the concept lattice, since nodes are inserted which do not introduce any new attributes (or objects) but represent greatest lower or least upper bounds.[5] If the induced hierarchy is to be used in a formalism which does not require it to be a lattice, it is probably sufficient to use the AOC-poset. However, it should be noted that lattices are mathematically well-known objects, which are easy to treat computationally, and for which many algorithms exist. Furthermore, since the concept lattice represents for every

---

[5] Under certain circumstances the number of nodes increases extremely; for instance, the number of nodes in the concept lattice capturing the derivational information of German lemmas contained in the lexical database CELEX is greater than 72.000; however, if the same information is represented in an AOC-poset the number of nodes is less than 4.000.
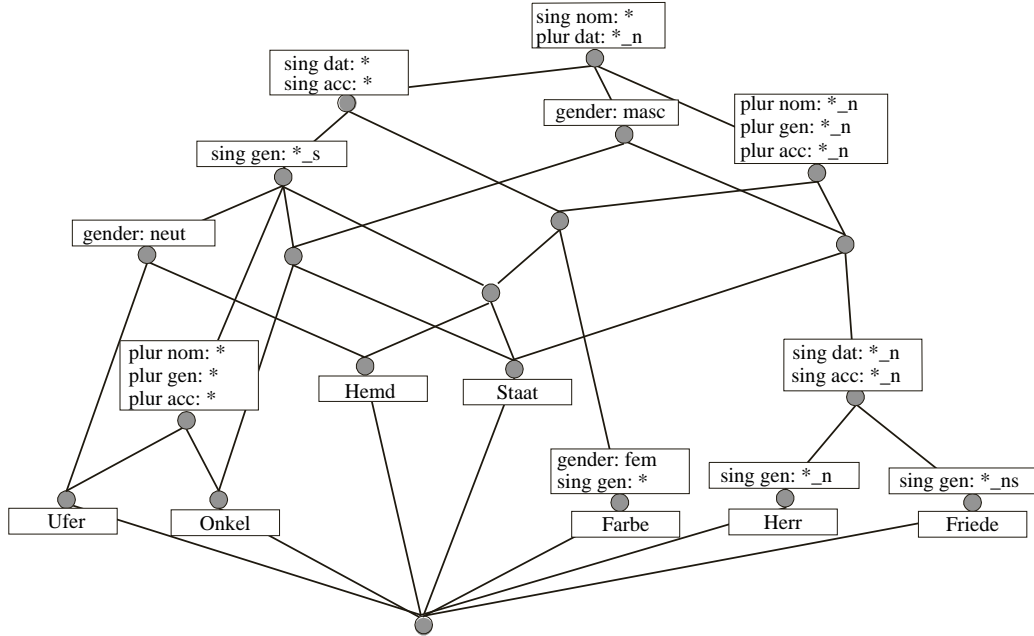
Fig. 4. concept lattice

set of objects their common attributes, in some cases inserting new objects in the hierarchy is simplified.

3 Notes on inducing regularities, subregularities and exceptions from flat data

3.1 General remarks

Since lexical information can be structured very well in terms of regular, sub-regular and exceptional forms, many theories allow elements of nonmonotonic inheritance, for instance default inheritance in DATR (cf. Evans & Gazdar 1996 [5]) or default unification theories based on (typed) feature structures (e.g. Copestake & Lascarides 1999 [4]). In nonmonotonic inheritance networks the dominance relation must be reinterpreted. In the framework of single inheritance networks this is normally done by stipulating that in the case of conflicting attributes those attributes attached to a node get precedence over those attributes which are inherited from parent nodes (for a formal semantics of nonmonotonic inheritance see for example the concept of "inferential distance ordering" in Touretzky 1986 [21]). In the case of multiple inheritance networks the interpretation of the inheritance relation gives rise to theoretical problems. One of the key questions is how to deal with mutually contradictory information inherited from two or more parent nodes. Several strategies have been applied to solve this problem, for example, excluding the problematic cases by stipulating orthogonal inheritance, marking explicitly which information is defeasible and may be overwritten, or ordering explicitly the parent

nodes. There are some new interpretations of default unification which try to avoid these stipulations (e.g. Copestake & Lascarides 1999 [4]).

Automatic induction of nonmonotonic structures leads to special problems, since the data do not reveal nonmonotonic relations as openly as monotonic ones. The set of possible representations using nonmonotonicity is much larger than the set of monotonic representations. Independently of the chosen semantics of nonmonotonic inheritance, it has to be decided which information may possibly be overwritten. In most approaches the idea is that the upper area of a network encodes the regular or default cases, while subregular and exceptional information is placed below these nodes. A node inherits the regular properties from its ancestors if they are not overwritten by subregular or exceptional properties.

To find a "good" representation using default information (defeasible information) a method has to be found to decide which information may be overwritten in which case. One way could be to investigate the similarity of attribute classes applicable to a set of objects. If two classes are very similar, one could become a subclass of the other; normally it is more favorable to make the class applying to a larger class of objects the upper (more general) class. For applying this method, a measure of similarity is needed; in the framework of Formal Concept Analysis similarity measures have been formalized (cf. Lengnink 1996 [11]; Leischner 1999 [10]). Another way to solve the task of deciding which information is to be marked as default is to investigate the set of attribute implications valid in the formal context.

3.2   Set of default information proposed by the set of implications

An *attribute implication* of a formal context is of the form "for every object for which the attributes $m_1, m_2, ..., m_k$ apply, the attributes $a_1, a_2, ..., a_i$ apply too" or $\{m_1, m_2, ..., m_k\} \rightarrow \{a_1, a_2, ..., a_i\}$ (cf. Ganter & Wille 1999). Let A and B be subsets of the attribute set M, the implication $A \rightarrow B$ is *valid* in the context if and only if $B \subseteq A''$.[6] In terms of the concept lattice this means that the concept (m', m'') is a superconcept of the concept (A', A'') for every $m \in B$. Since the union of every premise and the maximal corresponding conclusion forms an intent of one of the concepts of the context, the structure of the concept lattice is determined by the set of valid implications up to isomorphism. Some implications can be derived from others and so it is possible to define the notion of a *minimal base* of the valid implications (cf. Ganter & Wille 1999). A minimal implication base of the example context is given in table 3.[7]

To solve the task of deciding which set of attributes forms an acceptable set

---

[6] A'' is an abbreviation for $(A')'$

[7]   Implication 1 indicates that all objects of the example context have two attributes in common. Each of the implications 15-19 has zero support in the context, which means that there is no object to which all the attributes of the premise apply.

1	{ }	→	{sing nom:*, plur dat:*_n}
2	{ sing gen:* }	⇆	{ gender: fem }
3	{ gender: fem }	→	{ sing acc:*, plur nom:*_n }
4	{ plur nom:* }	⇆	{ plur gen:* } ⇆ {plur acc:* }
5	{ plur nom:* }	→	{ sing gen:*_s }
6	{ gender: neut }	→	{ sing gen:*_s }
7	{ sing gen:*_s }	→	{ sing acc:* }
8	{ sing acc:* }	⇆	{ sing dat:* }
9	{ sing gen:*_ns }	→	{ sing acc:*_n }
10	{ sing gen:*_n }	→	{ sing acc:*_n }
11	{ sing acc:*_n }	⇆	{ sing dat:*_n}
12	{ sing acc:*_n }	→	{ plur nom:*_n, gender: masc }
13	{ plur nom:*_n }	⇆	{ plur gen:*_n } ⇆ {plur acc:*_n }
14	{ gender: masc, sing acc:* }	→	{ sing gen:*_s }
15	{ plur gen:*_n, plur nom:* }	→	{ sing acc:*_n }
16	{ sing gen:*_s, gender: fem }	→	{ sing acc:*_n }
17	{ gender: masc, gender: neut }	→	{ sing acc:*_n }
18	{ sing acc:*_n, sing acc:* }	→	{ sing gen:*_n, sing gen:*_ns }
19	{ sing gen:*_n, sing gen:*_ns }	→	{ gender: neut, gender: fem, plur nom:* }

Table 3
minimal base of valid implications

of default information it must be considered what the qualities of well-chosen
default information are. First, default information should not be more specific
than non-default information in the sense that, if some default attributes imply
other attributes, the latter should belong to the set of default information
too. Another assumption is that using defaults in data representation should
decrease the size of the representation and make the structure more compact.
Finally, the default information must be consistent. In the framework of
formal concept lattices these assumptions lead to the following definition: An
acceptable set of default information D is a subset of the attribute set M
with the properties that $D = D''$ (all conclusions are included) and $D' \neq \emptyset$
(the default information is consistent). The last condition that there is at
least one object which has all the default attributes seems to be stronger than
the normal idea of consistency which only excludes conflicting values for a
single feature. But since in the present approach every feature-value pair is
represented by a single attribute and since by the first condition the default
information forms a concept intent, renunciation of the second condition would
entail that the set of all attributes constitutes an acceptable set of default
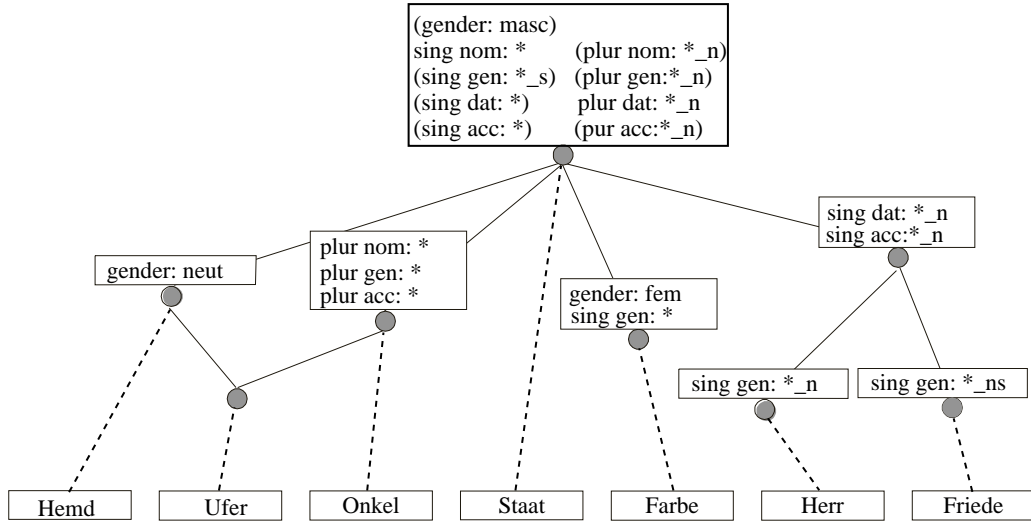
9

Fig. 5. example default inheritance hierarchy

information, possibly containing conflicting feature-value pairs.

The choice of the default case depends on the chosen representation formalism, the data to be represented and special wishes of the designer of the representation about its compactness and its hierarchical structure. Therefore it does not seem possible to find the best selection of default information automatically. The present approach proposes possible acceptable default sets, namely the intents of formal concepts with non-empty extent. The size of the extent, i. e. the number of objects having the default information in common and the size of the intent can be offered to support the decision of selecting a default set. With the methods presented above it is even possible to compute a nonmonotonic inheritance network corresponding to an acceptable default set. To avoid conflicts arising from multiply inherited conflicting information, the strategy of explicitly marking the default information as being defeasible is applied. The nonmonotonic network is obtained by computing the lattice corresponding to the subcontext consisting of the original context without those default attributes which do not apply to the set of all objects. Figure 5 shows the inheritance hierarchy corresponding to the acceptable default set {gender: masc, sing nom:*, sing gen:*_s, sing dat:*, sing acc:*, plur nom:*_n, plur gen:*_n, plur dat:*_n, plur acc:_n}, where the defeasible attributes are put in brackets. To model subregularities and exceptions it is possible to attach further default attribute sets to subnodes of the top node. In this case, one has to pay special attention to the correct inheritance of the attributes. Instead of looking at valid implications it is also possible to look at partial implications, which are valid except for a certain amount of exceptional objects (cf. Luxemburger 1994 [14]).

3.2.1 Some notes on attribute exploration

The example shows another interesting problem of data representation. As in most linguistic tasks concerning the lexicon, the universe of the objects is known, and the characteristic attributes which are to be used to classify the objects are fixed, but as usual the set of objects is too large to check each object-attribute pair. The linguist looks for a selection of objects which provide the same relations with respect to the attributes as the whole universe. Formal Concept Analysis offers a tool called *attribute exploration* to find such a *complete selection context* (cf. Stumme 1996 [19,20], Ganter & Wille 1999 [6]). It computes the minimal base of implications corresponding to the actual context and asks the user if each single implication is valid in the universe of objects or if a counterexample is known. The counterexamples are than added to the context and the implications are newly computed until all implications are accepted. In the present example the user would already reject implication 1 and add for example a noun like "Auto" which builds all its plural forms with '*_s'. In this way a selection context of representatives of all possible inflectional paradigms can be found.

4 Conclusions

In this paper I have presented three different ways of looking at the same data: in a tabular form (see tables 1 and 2), as a hierarchical network (see figures 4 and 3), and as a set of attribute implications (see table 3). Each of these views provides different advantages in the field of data processing. Data is often stored in tabular form since it allows easy data entry. Furthermore, a table directly exhibits the information available for a single object. Representing data in an hierarchical network helps to reveal more of the implicit structure of the stored information. Since generalizations are captured, inheritance hierarchies allow a compact and less redundant representation. In addition, networks are particularly suitable for classification tasks like classifying underspecified objects. Finally, the system of attribute implications explicitly reveals interdependencies between attributes. As shown in this paper, the set of implications can be used to support defining a default set and to construct a nonmonotonic inheritance hierarchy. Furthermore, attribute exploration makes it easier to efficiently explore a huge universe of objects in a guided way and to find a minimal selection of objects which represents the whole universe. I want to emphasize the remarkable property that the three data representations introduced are information equivalent in the sense that each of them can be obtained from each other. It turns out that applying ideas of Formal Concept Analysis to certain problems of knowledge representation and processing (e.g. linguistic lexicons) is a promising approach.

References

[1] Barg, P., *Automatic inference of DATR theories*, in: H.-H. Bock and W. Polasek, editors, *Data Analysis and Information Systems. Statistical and Conceptual Approaches (Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e. V. University of Basel, 1995-03-8/10)* (1996), pp. 506–515.

[2] Barg, P., "Automatischer Erwerb von linguistischem Wissen. Ein Ansatz zur Inferenz von DATR-Theorien," Niemeyer, Tübingen, 1996.

[3] Carpenter, B., "The Logic of Typed Feature Structures. With Applications to Unification Grammars, Logic Programs and Constraint Resolution." Cambridge University Press, Cambridge, 1992.

[4] Copestake, A. and A. Lascarides, *Default representation in constraint-based frameworks*, Computational Linguistics **25** (1999), pp. 55–105.

[5] Evans, R. and G. Gazdar, *Datr: A language for lexical knowledge representation*, Computational Linguistics **22** (1996), pp. 167–216.

[6] Ganter, B. and R. Wille, "Formal Concept Analysis. Mathematical Foundations," Springer, Berlin, 1999.

[7] Großkopf, A., *Formal concept analysis of verb paradigms in linguistics*, in: E. Diday, Y. Lechevallier and O. Opitz, editors, *Ordinal and Symbolic Data Analysis*, Springer, Berlin, 1996 pp. 70–79.

[8] Großkopf, A. and G. Harras, *Begriffliche Erkundung semantischer Strukturen von Sprechaktverben*, in: G. Stumme and R. Wille, editors, *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, Springer, Berlin, 1999 pp. 273–295.

[9] Kilbury, J., P. Naerger and I. Renz, *Simulation lexikalischen Erwerbs*, in: S. W. Felix, C. Habel and G. Rickheit, editors, *Kognitive Linguistik: Repräsentation und Prozesse*, Westdeutscher Verlag, Opladen, 1994 pp. 251–271.

[10] Leischner, L., *Additive similarity measures*, Technical Report FB4 preprint 2022, Technische Universität Darmstadt, Fachbereich Mathematik (1999).

[11] Lengnink, K., "Formalisierungen von Ähnlichkeiten aus der Sicht der Formalen Begriffsanalyse," Shaker Verlag, Aachen, 1996.

[12] Light, M., *Classification in feature-based default inheritance hierarchies*, in: H. Trost, editor, *Proceedings of KONVENS '94: Verarbeitung natürlicher Sprache. Österreichische Gesellschaft für Artificial Intelligence* (1994), pp. 220–229.

[13] Lüngen, H. and C. Sporleder, *Automatic induction of lexical inheritance hierarchies*, in: J. Gippert, editor, *Proceedings of 11th Annual Meeting of GLDV: Multilinguale Corpora. Codierung, Strukturierung, Analyse*, Prague, 1999, pp. 42–52.

[14] Luxenburger, M., *Partial implications*, Technical Report FB4 preprint 1612, Technische Universität Darmstadt, Fachbereich Mathematik (1994).

[15] Pollard, C. and I. Sag, *Information-based syntax and semantics*, Technical Report 13, Center for the Study of Language and Information (1987).

[16] Priß, U., *The formalization of WordNet by methods of relational concept analysis*, in: C. Fellbaum, editor, *WordNet: An Electronic Lexical Database and Some of its Applications*, MIT-Press, 1998 pp. 179–196.

[17] Priß, U., "Relational concept analysis: semantic structures in dictionary and lexical databases," Shaker Verlag, Aachen, 1998.

[18] Sporleder, C., "Learning Lexical Generalisations. An Operational Evaluation of Current Machine Learning Methods," Master's thesis, Universität Bielefeld (1999).

[19] Stumme, G., *Attribute exploration with background implications and exceptions*, in: H.-H. Bock and W. Polasek, editors, *Data Analysis and Information Systems. Statistical and Conceptual Approaches (Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e. V. ,University of Basel, 1995-03-8/10)* (1996), pp. 457–469.

[20] Stumme, G., *Exploration tools in formal concept analysis*, in: E. Diday, Y. Lechevallier and O. Opitz, editors, *Ordinal and Symbolic Data Analysis*, Springer, Berlin, 1996 pp. 70–79.

[21] Touretzky, D. S., "The Mathematics of Inheritance Systems," Research Notes in Artificial Intelligence, Morgan Kaufmann Publishers, Los Altos, 1986.

[22] Zickwolff, M., *Zur Rolle der formalen Begriffsanalyse in der Wissensakquisition*, in: R. Wille and M. Zickwolff, editors, *Begriffliche Wissensverarbeitung: Grundlagen und Aufgaben*, BI-Wissenschaftsverlag, Mannheim, 1994 pp. 173–190.