

Efficient Implementation of Semantic Relations in Lexical Databases

Uta Priss

School of Library and Information Science,
Indiana University, Bloomington, IN 47405.

Abstract

In this paper we develop a formalization of semantic relations that facilitates efficient implementations of relations in lexical databases or knowledge representation systems using bases. The formalization of relations is based on a modeling of hierarchical relations in Formal Concept Analysis. Further, relations are analysed according to Relational Concept Analysis which allows a representation of semantic relations consisting of relational components and quantificational tags. This representation utilizes mathematical properties of semantic relations. The quantificational tags imply inheritance rules among semantic relations that can be used to check the consistency of relations and to reduce the redundancy in implementations by storing only the basis elements of semantic relations. The research presented in this paper is an example of an application of Relational Concept Analysis to lexical databases and knowledge representation systems (compare Priss (1996)) which is part of a larger framework of research on natural language analysis and formalization.

Introduction

In this paper we define a formalization of semantic relations that occur in lexical databases. In semantic networks (for example in Evans & Gazdar (1989)), semantic relations are often implemented as relations between nodes that do not have strong mathematical properties. Some semantic relations are defined as symmetric; some are more or less transitive (compare the discussion on the transitivity of the meronymy (part/whole) relation in Winston et al. (1987)). Fischer (1991) formalizes basic properties of thesauri¹, inverse relations, symmetry, strong intransitivity, acyclicity, and uses them to examine structural consistencies. But his research does not include more specific formal properties of semantic relations. There should neither be too many nor too few formal properties of semantic relations. Too many or too strict properties limit the semantic network to applications which do not contain natural language expressions. Too few properties (for example, no transitivity) make it impossible to use inheritance of relations or properties and require that much redundant information has to be implemented. We claim that Relational Concept Analysis provides a tool for analysing the formal properties of semantic relations very carefully (compare Priss (1996)). The meronymy relation, for example,

¹Thesauri can be seen as highly formalized types of semantic networks.

can be classified into different formal types, some of which are transitive, some are intransitive. Each relational type can then be defined with an appropriate number of formal properties to avoid redundancy and over-formalism.

The second important factor that facilitates Relational Concept Analysis is the distinction between lexical and conceptual entities. The building blocks of a semantic network should be concepts and not words. For example, the word ‘cherry’ corresponds to (at least) three different concepts: ‘cherry1 - the fruit’, ‘cherry2 - the tree’, and ‘cherry3 - the wood’. This may seem to lead to redundancy since several concepts have to be implemented for one word, but disk space has become so cheap that conceptual clarity should be valued higher than storage space. Implementing one concept/word ‘cherry’ with three different sets of rules would increase the computational complexity involved in every query and update of ‘cherry’. Another aspect of the conceptual versus lexical modeling is the use of non-lexicalized concepts. If it improves the structure of a lexical database, it should be allowed to implement concepts, such as ‘wheeled vehicle’ and ‘wheel of vehicle’, which are usually non-lexicalized, i.e. they do not usually occur as words or entries in a dictionary.

We claim that following these principles (conceptual instead of lexical modeling, defining formal properties of semantic relations where possible) improves the efficiency of lexical databases or semantic networks. A further advantage of Relational Concept Analysis is that a formal *basis*² can be defined for semantic relations. From the basis, which usually consists of a relatively small set of representative elements of the relation, the complete relation can be reconstructed. Only the relational bases need to be implemented in a lexical database, the other relations follow from inheritance rules. This significantly reduces redundancy in the implementation of semantic relations. And it helps with the maintenance of the lexical database: changes of the semantic relation are made at the basis level. If at any point inconsistencies occur, the semantic relation is simply split into smaller consistent semantic relations with appropriate formal properties. Before we define a basis for semantic relations and demonstrate our theory with some examples, we give a short introduction to Formal Concept Analysis and Relational Concept Analysis.

Formal Concept Analysis

Formal Concept Analysis (Ganter & Wille, 1996) starts with the definition of a *formal context* \mathcal{K} as a triple (G, M, I) consisting of two sets G and M and a relation I between G and M (i.e. $I \subseteq G \times M$). The elements of G and M are called *formal objects* (Gegenstände) and *formal attributes* (Merkmale), respectively. The relationship is written as gIm or $(g, m) \in I$ and is read as ‘the formal object g has the formal attribute m ’. A formal context can be represented by a cross table which has a row for each formal object g , a column for each formal attribute m and a cross in the row of g and the column of m

²The term ‘basis’ is here not used in the sense of a ‘canonical basis’ as defined in the context of Sowa’s conceptual graphs, but in the mathematical sense of a generating system.

	juvenile	grown-up	female	male
person				
adult		×		
female person			×	
male person				×
child	×			
woman		×	×	
man		×		×
girl	×		×	
boy	×			×

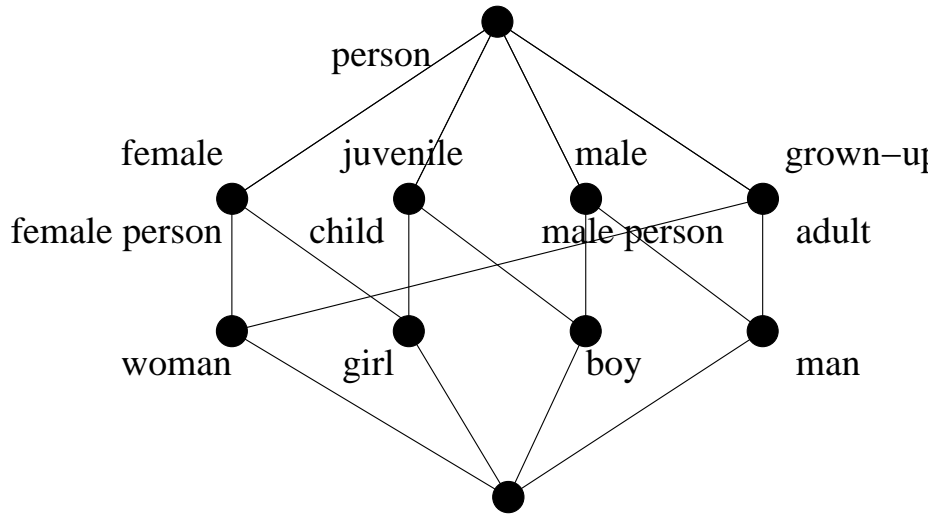


Figure 1: A formal context and a line diagram of its concept lattice

if gIm . The upper half of Figure 1 shows an example of a formal context. It has ‘person’, ‘adult’, and so on as formal objects, and ‘juvenile’, ‘grown-up’, ‘female’, and ‘male’ as formal attributes. It should be noted that this use of ‘context’ must be distinguished from the linguistic use of ‘context’. In a context (G, M, I) the set of all common formal attributes of a set $A \subseteq G$ of formal objects is denoted by $\iota A := \{m \in M \mid gIm \text{ for all } g \in A\}$ and, analogously, the set of all common formal objects of a set $B \subseteq M$ of formal attributes is $\varepsilon B := \{g \in G \mid gIm \text{ for all } m \in B\}$. For example, in the formal context in Figure 1, $\iota\{\text{man}\} = \{\text{grown-up, male}\}$ and $\varepsilon\{\text{grown-up}\} = \{\text{adult, woman, man}\}$ hold.

A pair (A, B) is said to be a *formal concept* of the formal context (G, M, I) if $A \subseteq G, B \subseteq M, A = \varepsilon B$, and $B = \iota A$. In this paper formal concepts are denoted by c, c_1, c_i and so on. For a formal concept $c := (A, B)$, A is called the *extent*

(denoted by $Ext(c)$) and B is called the *intent* (denoted by $Int(c)$) of the formal concept. In the example of Figure 1, $(\{\text{adult, woman, man}\}, \{\text{grown-up}\})$ is a formal concept, because $\iota\{\text{adult, woman, man}\} = \{\text{grown-up}\}$ and $\varepsilon\{\text{grown-up}\} = \{\text{adult, woman, man}\}$. The set of all formal concepts of (G, M, I) is denoted by $\mathcal{B}(G, M, I)$. The most important structure on $\mathcal{B}(G, M, I)$ is given by the formal subconcept-superconcept relation that is defined as follows: the formal concept c_1 is a *formal subconcept* of the formal concept c_2 (denoted by $c_1 \leq c_2$) if $Ext(c_1) \subseteq Ext(c_2)$, which is equivalent to $Int(c_2) \subseteq Int(c_1)$; c_2 is then a *formal superconcept* of c_1 (denoted by $c_1 \geq c_2$). For example, $(\{\text{adult, woman, man}\}, \{\text{grown-up}\})$ as a formal superconcept of $(\{\text{woman}\}, \{\text{grown-up, female}\})$ includes all the formal objects of the subconcept, but contains only a subset of the formal attributes of $(\{\text{woman}\}, \{\text{grown-up, female}\})$. It follows from this definition that each formal concept is a formal subconcept of itself in contrast to the natural language use of ‘subconcept’ which excludes a concept from being a subconcept of itself. The relation ‘ \leq ’ is a mathematical order relation called *formal conceptual ordering* on $\mathcal{B}(G, M, I)$ with which the set of all formal concepts forms a mathematical lattice denoted by $\underline{\mathcal{B}}(G, M, I)$. (This means the greatest common subconcept and the least common superconcept exist for all sets of concepts.)

Graphically, mathematical lattices can be visualized by line diagrams which represent a formal concept by a small circle. For each formal object g the smallest formal concept to whose extent g belongs is denoted by γg . And for each formal attribute m the largest formal concept to whose intent m belongs is denoted by μm . The concepts γg and μm are called the *object concept* of g and the *attribute concept* of m , respectively. In the line diagram it is not necessary to write the full extent and intent for each concept, instead the name (verbal form) of each formal object g is written slightly below the circle of γg and the name of each formal attribute m is written slightly above the circle of μm . The lower half of Figure 1 shows the line diagram of the concept lattice of the formal context in Figure 1. To read the line diagram, the extent of a formal concept consists of all formal objects which are retrieved by starting with the formal concept and then collecting all formal objects that are written at formal subconcepts of that formal concept. Analogously, the intent is retrieved by collecting all formal attributes that are written at formal superconcepts of the formal concept. More details on Formal Concept Analysis can be found in Ganter & Wille (1997).

Relational Concept Analysis

Relational Concept Analysis is the extension of Formal Concept Analysis—which provides a conceptual hierarchy—to a more general theory that includes other relations among objects or attributes. It is also an extension of Woods’ (1990) quantificational tags and inheritances. In what follows, only binary relations $r \subseteq G \times G$ are considered. These relations are transferred to relations among concepts, i. e., $R \subseteq \mathcal{B}(G, M, I) \times \mathcal{B}(G, M, I)$, according to the following definitions. The quantifiers that are used in the definitions can be natural lan-

guage quantifiers or mathematical expressions, such as $\|\text{all}\|$, $\|\text{at least } 1\|$ (or $\|\geq 1\|$), $\|\text{exactly } 1\|$ (or $\|\equiv 1\|$), or $\|\text{at least all but } 3\|$ (or $\|\gt (\text{all} - 3)\|$) (for more details on natural language quantifiers see Westerstahl (1989)).

Definition 1:

For a context (G, M, I) , concepts $c_1, c_2 \in \mathcal{B}(G, M, I)$, a relation $r \subseteq G \times G$, and quantifiers Q^i , $1 \leq i \leq 4$, we define

$$c_1 R^r[Q^1, Q^2;] c_2 :\iff Q_{g_1 \in \text{Ext}(c_1)}^1 Q_{g_2 \in \text{Ext}(c_2)}^2 : g_1 r g_2 \quad (1)$$

$$c_1 R^r[; Q^3, Q^4] c_2 :\iff Q_{g_2 \in \text{Ext}(c_2)}^3 Q_{g_1 \in \text{Ext}(c_1)}^4 : g_1 r g_2 \quad (2)$$

$$c_1 R^r[Q^1, Q^2; Q^3, Q^4] c_2 :\iff c_1 R^r[Q^1, Q^2;] c_2 \text{ and } c_1 R^r[; Q^3, Q^4] c_2 \quad (3)$$

r is called the *relational component* and $[Q^1, Q^2;]$, $[; Q^3, Q^4]$, or $[Q^1, Q^2; Q^3, Q^4]$ are called the *quantificational tag* of a relation. If no ambiguities are possible, relational component and quantificational tag can be omitted in the notation of the relation.

Depending on the quantifiers, each relation r therefore leads to several different relations R^r among concepts. The terms ‘quantificational tag’ and ‘relational component’ are taken from Woods’ terminology. The formalization can best be understood through an example: ‘all door-handles are parts of doors’ states a meronymy relation between door-handles and doors. More precisely it means that all objects that belong to the extent of the concept ‘door-handle’ have an object in the extent of the concept ‘door’ such that the meronymy relation holds between them. The variables in equivalence (1) are for this example $Q^1 := \|\text{all}\|$, $Q^2 := \|\geq 1\|$, c_1 is the concept ‘door-handle’, c_2 is the concept ‘door’, and r is the relation ‘is part of’. Equivalence (2) could be ‘there is at least one door which has a handle’, because ‘all doors have to have handles’ is not true. Equivalence (3) is the conjunction of the first two. For the door-handle example the quantifiers are $Q^1 := \|\text{all}\|$, $Q^2 := \|\geq 1\|$, $Q^3 := \|\geq 1\|$ and $Q^4 := \|\geq 1\|$. Abbreviations are used for the more frequently used types of relations:

Definition 2:

‘ $R^r[\|\geq 1\|, \|\geq 1\|; \|\geq 1\|, \|\geq 1\|]$ ’ is abbreviated as R_0^r .
‘ $R^r[\|\text{all}\|, Q^2; \|\text{all}\|, Q^4]$ ’ is abbreviated as $R_{(Q^4; Q^2)}^r$. The vertical lines ‘|’ can be left out for Q^4 and Q^2 in the subscript of $R_{(Q^4; Q^2)}^r$

Besides its applications to the modeling of lexical databases, this formalization can be used to describe functions $R_{(\geq 0; 1)}^r$, bijections $R_{(1; 1)}^r$, or Cartesian products $R_{(\text{all}; \text{all})}^r$. It is useful to characterize a conceptual relation by considering the concepts only and not the objects. This leads to the definition of *characteristics* of the relations.

Definition 3:

Q^1	Q^2	Q^5	Q^6
$\ \text{all}\ $	$\ \text{all}\ $	$\ \text{all}\ $	$\ \text{all}\ $
$\ \geq n\ $	$\ \geq m\ $	$\ \geq 1\ $	$\ \geq 1\ $
$\ \text{all}\ $	$\ \geq n\ $	$\ \text{all}\ $	$\ \geq 1\ $
$\ \geq n\ $	$\ \text{all}\ $	$\ \geq 1\ $	$\ \text{all}\ $

Table 1: Different quantifiers for equivalence (4)

A relation $R \subseteq \mathcal{B}(G, M, I) \times \mathcal{B}(G, M, I)$ for which there exist quantifiers Q^5 , Q^6 so that for all $c_1, c_2 \in \mathcal{B}(G, M, I)$

$$c_1 R c_2 \iff Q^5_{c_{11} \leq c_1} Q^6_{c_{21} \leq c_2} : c_{11} R c_{21} \quad (4)$$

holds is called *of characteristic* $[Q^5, Q^6;]$. Relations of characteristic $[; Q^5, Q^6]$ are defined analogously.

It is not known whether all relations have such a characteristic. But if the set of quantifiers for Q^1 and Q^2 in (1) is limited to $\{\|\> (\text{all} - n)\|, \|\geq n\|, \|\leq (\text{all} - n)\|, \|\leq n\| \mid n \geq 1\}$ then $Q^5, Q^6 \in \{\|\text{all}\|, \|\geq 1\|\}$ in equivalence (4) (see Priss (1996)). Table 1 shows a subset of these cases ($n \geq 1, m \geq 1$). An interpretation of Table 1 is that if a specific number occurs on the object level, for example, all hands have five fingers, it does not occur on the conceptual level. For a concept ‘hand’ there is one concept ‘finger’ so that each object of ‘hand’ has five parts among the objects of ‘finger’; and not: for a concept ‘hand’ exist five concepts ‘finger’ with that property. A linguistic example where this is even reflected in the language is that ‘having two shoes’ can also be expressed as ‘having a pair of shoes’. From equivalence (4) follows equivalence (5) if $R^r[Q^1, Q^2;]$ is of characteristic $[Q^5, Q^6;]$.

$$\gamma g_1 R^r[Q^1, Q^2;] \gamma g_2 \iff Q^5_{g_{11} \in \text{Ext}(\gamma g_1)} Q^6_{g_{21} \in \text{Ext}(\gamma g_2)} : \gamma g_{11} R^r[Q^1, Q^2;] \gamma g_{21} \quad (5)$$

It is therefore enough to consider object concepts in order to determine the characteristic of a relation.

Additional properties of the relation r have consequences for the relations R^r . For example, if r is irreflexive and transitive (and thus, by implication, antisymmetric) and the sets of objects of the given contexts are finite, then $R^r_{(\geq 0; \geq 1)}$, $R^r_{(\geq 1; \geq 0)}$, and $R^r_{(\geq 1; \geq 1)}$ are also irreflexive, antisymmetric, and transitive. If r is the equality relation (=) then $R^r_{(\geq 0; \geq 1)}$ is an order relation, $R^r_{(\geq 1; \geq 0)}$ is the dual order, and $R^r_{(\geq 1; \geq 1)}$ is an equivalence relation, and the following equivalences hold

$$c_1 R^r_{(\geq 0; \geq 1)} c_2 \iff c_1 \leq c_2 \quad (6)$$

$$c_1 R^r_{(\geq 1; \geq 0)} c_2 \iff c_1 \geq c_2 \quad (7)$$

$$c_1 R^r_0 c_2 \iff \text{Ext}(c_1) \cap \text{Ext}(c_2) \neq \emptyset \quad (8)$$

$$c_1 R^r_{(\geq 1; \geq 1)} c_2 \iff c_1 = c_2 \quad (9)$$

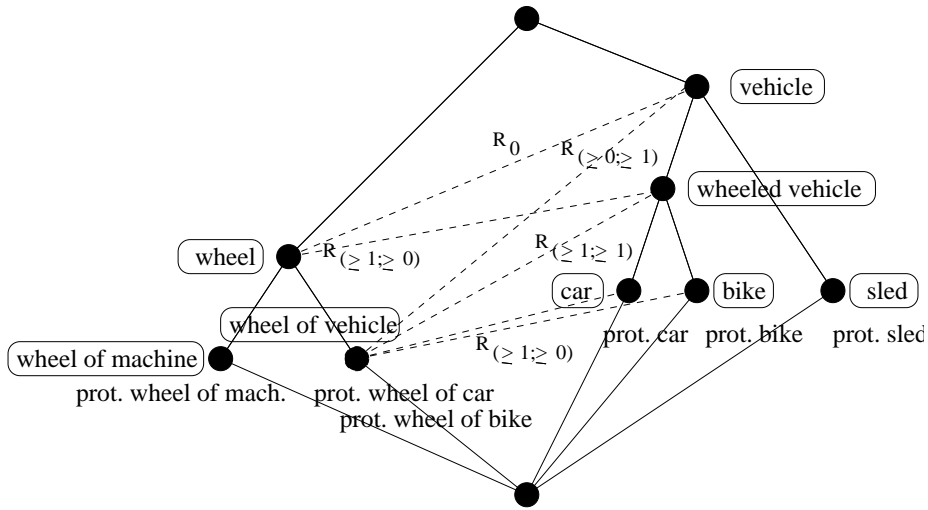


Figure 2: Different meronymy relations

Therefore the conceptual ordering itself results from a relation between objects. For further details and the proofs of the statements above see Priss (1996).

Bases of Semantic Relations

Figure 2 shows a part of a lexical database. The formal objects for this lexical database are prototypical objects, such as ‘prototypical car’. In constructing lexical databases it is sometimes difficult to select the distinguishing attributes. For example, what are the exact attributes of ‘car’ which are shared by all cars among the objects, but not by other objects? A simple solution for this problem is to construct a concept hierarchy based on the objects and then to determine the attributes afterwards or select default attributes, such as the defining attribute of a car is ‘to be a car’. For these reasons we omit the formal attributes in Figures 2 and 3. The words in the ellipses of Figures 2 and 3 denote names of the concepts.

The dotted lines in Figure 2 denote a meronymy (or part/whole) relation. We will refer to R and r both as ‘meronymy’ relation although one is a concept relation whereas the other is a relation among objects. The object relation r has only the two pairs (prototypical wheel of car, prototypical car) and (prototypical wheel of bike, prototypical bike). The relation $R_{(\geq 1; \geq 1)}^r$ holds between ‘wheel of vehicle’ and ‘wheeled vehicle’ because all wheels of vehicles are part of a wheeled vehicle and all wheeled vehicles have wheels of vehicles. It is the ‘strongest’ relation in the sense that it implies all other relations in Figure 2. The ‘weakest’ relation (R_0^r) holds, for example, between ‘wheel’ and ‘vehicle’. There is at least one wheel (for example, the prototypical wheel of bike) and at least one vehicle (for example, the prototypical bike) so that the wheel is part of the vehicle. But

not all wheels are part of vehicles, nor do all vehicles have wheels.

In the rest of this paper we will define a *basis* for semantic relations and demonstrate that using a basis only one pair of the meronymy relation in Figure 2 has to be implemented so that the complete relation is implied by the basis pair. An $||\text{all}||$ -quantifier for Q^5 (and Q^6 respectively) in equivalence (4) obviously causes inheritance of the relation to all subconcepts, whereas an $||\geq 1||$ -quantifier causes inheritance to all superconcepts. For example, if all bird feathers are parts of birds then all sparrow feathers (as special bird feathers) are parts of birds and all bird feathers are parts of animals (a generalization of bird). In lattice terms, relations that use $||\text{all}||$ - or $||\geq 1||$ -quantifiers for Q^5 (or Q^6) in equivalence (4) always hold for the corresponding ideals and filters, respectively. The relations between the smallest elements of the filters and the largest elements of the ideals form a basis. This is formalized in the next definition. Theorem 1 (for the proof of this theorem see Priss (1996)) shows that bases exist and are unique for the concept relations that have characteristics as in Table 1. Relations of characteristic $[\cdot; Q^5, Q^6]$ can be treated analogously.

Definition 4:

A *basis* \mathcal{R} of a relation R of characteristic $[\cdot; ||\text{all}||, ||\geq 1||; \cdot]$ is defined as a relation $\mathcal{R} \subseteq \mathcal{B}(G, M, I) \times \mathcal{B}(G, M, I)$ satisfying, for all $c_1, c_2 \in \mathcal{B}(G, M, I)$,

$$\text{a) } c_1 R c_2 \iff \exists_{(c_1^\circ, c_2^\circ) \in \mathcal{R}} : (c_1 \leq c_1^\circ \text{ and } c_2 \geq c_2^\circ) \quad (10)$$

and b) \mathcal{R} has the minimal number of elements among all relations that fulfill equivalence (10). Bases of characteristics $[\cdot; ||\text{all}||, ||\text{all}||; \cdot]$, $[\cdot; ||\geq 1||, ||\geq 1||; \cdot]$ and $[\cdot; ||\geq 1||, ||\text{all}||; \cdot]$ are defined respectively using $(c_1 \leq c_1^\circ \text{ and } c_2 \leq c_2^\circ)$, $(c_1 \geq c_1^\circ \text{ and } c_2 \geq c_2^\circ)$, and $(c_1 \geq c_1^\circ \text{ and } c_2 \leq c_2^\circ)$.

Theorem 1:

Bases as defined in Definition 4 are unique. A relation $R \subseteq \mathcal{B}(G, M, I) \times \mathcal{B}(G, M, I)$ is of characteristic $[\cdot; ||\text{all}||, ||\geq 1||; \cdot]$, $[\cdot; ||\text{all}||, ||\text{all}||; \cdot]$, $[\cdot; ||\geq 1||, ||\geq 1||; \cdot]$ or $[\cdot; ||\geq 1||, ||\text{all}||; \cdot]$ if and only if it has a basis according to Definition 4.

Equivalence (10) uses the fact that $||\text{all}||$ -quantifiers cause inheritance to subconcepts (therefore $c_1 \leq c_1^\circ$) and $||\geq 1||$ -quantifiers cause inheritance to superconcepts (therefore $c_2 \geq c_2^\circ$). A basis can thus be used to define a characteristic of a concept relation that is defined on the concepts and not derived from a relation among objects or attributes.

Figure 3 shows the same example as Figure 2. The boldface dotted line is the basis for the relation $R_{(\geq 0; \geq 1)}$, which is represented by the dotted lines in the figure. In this case $R_{(\geq 1; \geq 0)}$ which is not shown in the figure has the same basis as $R_{(\geq 0; \geq 1)}$. There is not usually a basis for a relation $R_{(\geq 1; \geq 1)}$ which is usually represented as an intersection of relations $R_{(\geq 0; \geq 1)}$ and $R_{(\geq 1; \geq 0)}$. All the concepts in the lower ellipsis are in relation $R_{(\geq 0; \geq 1)}$ to all concepts in the upper ellipsis. ‘Wheel of vehicle’ is the most general part of ‘wheeled vehicle’ which is itself the most specific whole of ‘wheel of vehicle’. A basis can consist

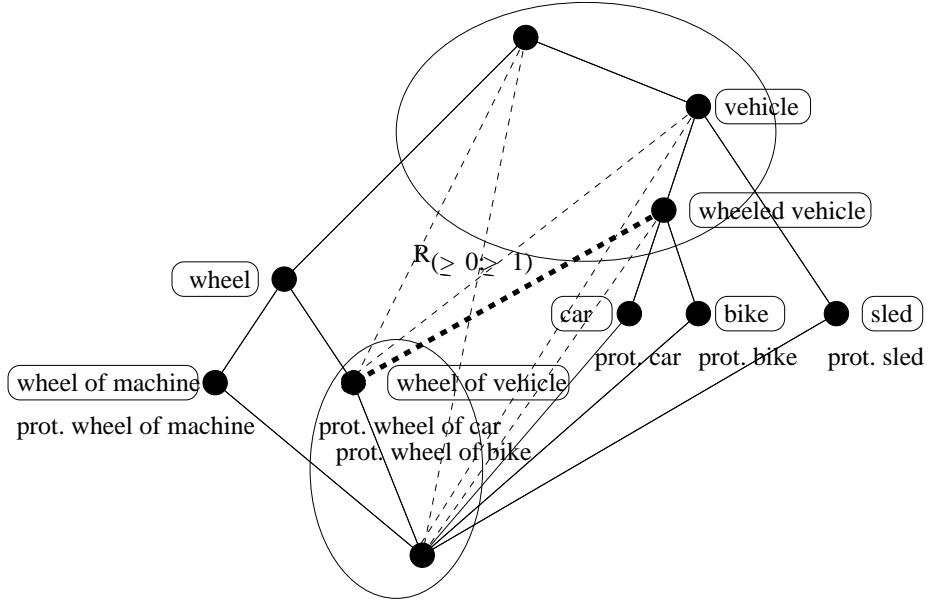


Figure 3: A basis for meronymy relations

of several elements, which means a concept can have several most general parts or most specific wholes. For example, ‘wheeled vehicle’ could also have ‘engine of wheeled vehicle’ as another most general part. Of course, then there could be a concept ‘parts of wheeled vehicles’ generated or identified (by adding an attribute ‘is part of wheeled vehicle’ to the context and assigning it to each object that is in ‘part of’ relation to ‘wheeled vehicle’). This new concept would be a hypernym of ‘engine of wheeled vehicle’ and ‘wheel of vehicle’. ‘Parts of wheeled vehicles’ would be in $R_{(\geq 0; \geq 1)}$ relation to ‘wheeled vehicle’. This pair could then be used as a basis element. On the other hand, the pair (parts of wheeled vehicles, wheeled vehicle) cannot replace the pairs (engine of wheeled vehicle, wheeled vehicle) and (wheel of vehicle, wheeled vehicle) as basis element for the $R_{(\geq 1; \geq 0)}$ relation. It should be obvious by now that, although the mathematical properties of bases of semantic relations are explained in this paper, there are still open questions for actual implementations, especially since neither ‘engine of wheeled vehicle’ nor ‘wheel of vehicle’ nor ‘parts of wheeled vehicle’ are usually lexicalized.

There is also a limit based on the mathematical theory to the kind of concepts that can be generated or identified by adding appropriate attributes to the context. For example, if $c_1 R_{(\geq 0; \geq 1)} c_2$ holds then a concept c_1^* with $c_1 R_{(\geq 1; \geq 1)} c_1^*$ can be generated by adding an attribute m to the context with, for all $g \in G$, $gIm : \iff (g \in Ext(c_2) \text{ and } \exists_{g_1 \in Ext(c_1)} : g_1 r g)$. But for $c_1 R_{(\geq 0; \geq 2)} c_2$ it is only possible to generate $c_1 R_{(\geq 1; \geq 2)} c_1^*$ with $Ext(c_1^*) := \{g_2 \in Ext(c_2) \mid \exists_{g_1 \in Ext(c_1)} : g_1 r g_2\}$ because $c_1 R_{(\geq 2; \geq 2)} c_1^*$ may not be true for any c_1^* . For the implementation

of meronymy it is helpful if these concepts exist, therefore the question arises whether and how non-lexicalized concepts should be added to a lexical database to ease the implementation of semantic relations — especially, since a verbal description of these non-lexicalized concepts is easy to obtain using ‘part of...’, ‘most general part of ...’, and so on.

Conclusion

The discussion in the last paragraph shows that there are several questions left open concerning the implementation of natural language in lexical databases. These questions consider not so much the mathematical details as the problems of adding non-lexicalized concepts. As stated in the introduction, non-lexicalized concepts can be helpful to reduce redundancies and, as explained in the other sections of this paper, they can be useful for keeping the bases of semantic relations small. On the other hand, adding too many non-lexicalized concepts may have a negative impact on the usability of the lexical database. These questions require further empirical research.

The more important result of this paper is the fact that a basis implementation of semantic relations facilitates consistency checks. According to the characteristic of a semantic relation, the relation is inherited by certain sub- and superconcepts. The implementation of this feature should be straight forward. Each time a new pair of an existing relation is added to the lexical database, it has to be checked whether the new pair is already implied by the existing basis (in which case the new pair is not added), whether it needs to be added to the basis, or whether it contradicts the basis (in which case a closer investigation of the involved concepts is required). Adding a new complete semantic relation requires the determination of its characteristic and its basis. This could be achieved interactively: the user adds the relation, the computer checks the consistency and prints a list of further pairs for which the relation should also hold. The user either accepts these pairs or modifies the characteristic of the relation. Deleting a concept from the database does not cause problems, unless a basis of a semantic relation is implemented using the concept, in which case the computer suggests alternative concepts that can be used for the basis. If concepts are added, again the user will have to determine interactively with the computer whether existing relations can (or must) be extended to the new concepts.

References

- Evans, R; Gazdar, G. (1989). *The semantics of DATR*. In: Cohn, A. G. (ed.). Proc. Seventh Conf. of the AISB. Falmer, Sussex, pp. 79-87.
- Fischer, Dietrich H. (1991). *Consistency Rules and Triggers for Thesauri*. Int. Classif., 18, No. 4.

- Ganter, Bernhard; Wille, Rudolf (1997). *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg. (Translation of: *Formale Begriffsanalyse: Mathematische Grundlagen*. Springer, Heidelberg 1996.)
- Priss, Uta (1996). *Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases*. Dissertation, TH-Darmstadt.
- Westerstahl, Dag (1989). *Quantifiers in Formal and Natural Languages*. In: Gabbay, D.; Guenther F. (ed.). *Handbook of Philosophical Logic*. Vol. 4, Kluwer, Dordrecht.
- Winston, Morton E.; Chaffin, Roger; Herrmann, Douglas (1987). *A Taxonomy of Part-Whole Relations*. Cogn. Science. Vol. 11, pp. 417-444.
- Woods, William A. (1990). *Understanding Subsumption and Taxonomy: A Framework for Progress*. In: Sowa, John (ed.). *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, M. Kaufmann, California.