

The Concept Classification of a Terminology Extended by Conjunction and Disjunction

Gerd Stumme

Technische Hochschule Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D-64289 Darmstadt, stumme@mathematik.th-darmstadt.de

© Springer-Verlag Berlin-Heidelberg 1996

1 Introduction

At the two conferences KRUSE '95 ([5]) and ICCS '95 ([6], [7]) held at Santa Cruz in August 1995 researchers on description logics, conceptual graphs, and formal concept analysis came together and discovered common interests and tasks. A fruitful discussion revealed that these three disciplines should integrate their research. Therefore common developments were considered. In one of the presented papers ([2]), for instance, F. Baader demonstrated how a classification algorithm providing more information can be built by combining a subsumption algorithm of description logics with a knowledge acquisition tool of formal concept analysis. In this paper we show how a classification algorithm providing still more information can be obtained by choosing another acquisition tool of formal concept analysis.

Much work has been done to develop algorithms for computing the subsumption hierarchy for knowledge representation systems based on description logics (also called KL-ONE like systems, terminological knowledge representation systems; cf. [1]). In [2], F. Baader describes how this computation can be extended to all conjunctions of concepts given in a terminology (TBox). He applies *attribute exploration* [8], an exploration tool of formal concept analysis (cf. [22], [9]) which is usually used as an interactive procedure to interview efficiently a human expert about a certain domain of knowledge. Instead of computing only the partially ordered set of the concepts in the TBox with the subsumption ordering, he obtains the complete semi-lattice of all possible conjunctions of concepts in the TBox. Since every complete semi-lattice is in fact a complete lattice, the existence of suprema (i. e., least common superconcepts) is asserted. However they generally differ from the disjunction – unlike the infima which are always equal to the conjunction of concepts. This paper describes, how the complete lattice of all possible combinations of conjunctions and disjunctions (and negations) of concepts in the TBox can be computed by applying another exploration tool of formal concept analysis, namely *distributive concept exploration* [17], instead of attribute exploration.

As in [2] we restrict ourselves to the description logic language \mathcal{ALC} , but the results can be generalized to other languages. The basic notions of \mathcal{ALC} are recalled in the next section. There we also give a short introduction into formal concept analysis.

2 Description Logic and Formal Concept Analysis

In this section we briefly recall the basic notations of the description logic \mathcal{ALC} and of formal concept analysis. For a more detailed introduction we refer to [14] and [1] for \mathcal{ALC} , and to [9] and [22] for formal concept analysis.

2.1 The Description Logic \mathcal{ALC}

The *syntax* of \mathcal{ALC} is built from a set of *concept names* and a set of *role names*. *Concept descriptions* are defined recursively:

- The concept names (which are assumed to contain two particular names \top and \perp for the *top* and the *bottom concept*) are concept descriptions.
- If C and D are concept descriptions and R is a role name, then $C \sqcap D$ (*conjunction*), $C \sqcup D$ (*disjunction*), $\neg C$ (*negation*), $\exists R.C$ (*existential restriction*), and $\forall R.C$ (*value restriction*) are concept descriptions.

A *terminological axiom* is a pair $A=D$ where A is a concept name different from \top and \perp and D is a concept description. A *terminology* (*TBox*) is a finite set \mathcal{T} of terminological axioms such that there are no cyclic and no multiple definitions. The concepts A appearing in an axiom $A = D$ on the left side are called *defined concepts*, otherwise they are called *primitive concepts*.

Next we describe the *semantics* of \mathcal{ALC} : An *interpretation* \mathcal{I} consists of a set $\text{dom}(\mathcal{I})$ and of a function $(\)^{\mathcal{I}}$ which maps every concept name to a subset of $\text{dom}(\mathcal{I})$ (\top has to be mapped to $\text{dom}(\mathcal{I})$ and \perp to the empty set) and every role name to a binary relation on $\text{dom}(\mathcal{I})$. This mapping is recursively extended to concept descriptions by

- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$,
- $(\neg C)^{\mathcal{I}} := \text{dom}(\mathcal{I}) \setminus C^{\mathcal{I}}$,
- $(\exists R.C)^{\mathcal{I}} := \{x \in \text{dom}(\mathcal{I}) \mid \exists y \in \text{dom}(\mathcal{I}): (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$,
- $(\forall R.C)^{\mathcal{I}} := \{x \in \text{dom}(\mathcal{I}) \mid \forall y \in \text{dom}(\mathcal{I}): (x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$.

A *model* of a TBox \mathcal{T} is an interpretation \mathcal{I} which satisfies the equality $A^{\mathcal{I}} = D^{\mathcal{I}}$ for all terminological axioms $A=D$ in the TBox \mathcal{T} .

We say that a concept description D *subsumes* a concept description C *with respect to* a TBox \mathcal{T} ($C \sqsubseteq_{\mathcal{T}} D$), if the inequality $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models \mathcal{I} of \mathcal{T} .

In [14], a *subsumption algorithm* is described which computes for given concept descriptions C and D whether C is subsumed by D with respect to a TBox \mathcal{T} . In [2], it is shown that, if C is not subsumed by D then the algorithm can provide a “counterexample”, i. e. a model \mathcal{I} of \mathcal{T} and an individual $c \in \text{dom}(\mathcal{I})$ with $c \in C^{\mathcal{I}} \setminus D^{\mathcal{I}}$.

2.2 Formal Concept Analysis

Formal concept analysis is based on the philosophical understanding of a concept as a unit of thought consisting of two parts: the extension contains all objects belonging to the concept and the intension contains all attributes valid for all these objects (cf. [21]). Formal concept analysis starts with a *formal context* (G, M, I) which consists of two sets G and M and a relation $I \subseteq G \times M$. The elements of G and M are called *objects* and *attributes*, respectively, and $(g, m) \in I$ is read as “the object g has the attribute m ”.

Now, the *formal concepts* of the context (G, M, I) are all pairs (A, B) with $A \subseteq G$ and $B \subseteq M$ such that (A, B) is maximal with the property $A \times B \subseteq I$. The set A is called the *extent* and the set B is called the *intent* of the formal concept (A, B) . The set $\mathfrak{B}(G, M, I)$ of all formal concepts of a formal context with the ordering $(A_1, B_1) \leq (A_2, B_2) : \iff A_1 \subseteq A_2$ is always a complete lattice which is called the *concept lattice* of the context (G, M, I) (cf. [22]). The ordering reflects the subconcept-superconcept-relation.

Next we introduce the two derivations $A' := \{m \in M \mid \forall g \in A: (g, m) \in I\}$ for $A \subseteq G$, and $B' := \{g \in G \mid \forall m \in B: (g, m) \in I\}$ for $B \subseteq M$. The fact that (A, B) with $A \subseteq G$ and $B \subseteq M$ is a formal concept is equivalent to $A' = B$ and $A = B'$. The smallest formal concept having an object g in its extent is $\gamma g := (\{g\}'', \{g\}')$, the largest formal concept having an attribute m in its intent is $\mu m := (\{m\}', \{m\}'')$. In the concept lattice, infima and suprema are calculated as follows:

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)'' \right), \quad \bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)'', \bigcap_{t \in T} B_t \right)$$

Every complete lattice can be viewed as a concept lattice: The Basic Theorem of Formal Concept Analysis (cf. [22]) shows that a complete lattice L is isomorphic to the concept lattice $\mathfrak{B}(L, L, \leq)$. We say that a complete lattice L is *represented* by a formal context (G, M, I) if $L \cong \mathfrak{B}(G, M, I)$. If L is a finite lattice then it is also isomorphic to the concept lattice $\mathfrak{B}(J(L), M(L), \leq)$ where $J(L)$ is the set of all join-irreducible elements and $M(L)$ is the set of all meet-irreducible elements of L . The context $(J(L), M(L), \leq)$ is said to be *reduced*. It is (up to isomorphism) the unique minimal context which represents L .

Since description logics and formal concept analysis have been developed independently, the notations are slightly different (see [27] for an extensive discussion): The concepts in description logics are understood as unary predicates. Hence they correspond more to the attributes in formal concept analysis than to the formal concepts, which have no direct counterpart in description logics. The conjunction of concepts in description logics correspond directly to the infimum of attribute concepts in formal concept analysis. In [11] and [15], concept formations like negation and disjunction are discussed for formal concept analysis, since they are important for the handling of incomplete knowledge (cf. [4], [11], [25], [26]) in *conceptual knowledge systems* [26]. For her dissertation, U. Priß is working on adding existential and value restriction (cf. also [12]).

Description logics have a strict distinction between the TBox containing purely intensional definitions of concepts and roles, and the ABox providing information about individuals. In formal concept analysis, extension and intension are understood as two aspects of a concept which cannot be treated separately.

3 Extending the Concept Classification of a Terminology

It is efficient to provide the subsumption relationships of the concepts in a terminology explicitly as a partially ordered set for further computations. The computation of the ordering, called *classification*, is done by repeatedly applying a subsumption algorithm. For two given concepts C and D the subsumption algorithm computes whether C is subsumed by D with respect to a terminology. In [14], the first sound and complete subsumption algorithm for \mathcal{ALC} is given.

Although the classification gives important information about a terminology, there are hierarchical dependencies between the concepts that cannot be described. In [2] (where also the subsumption algorithm of [14] is described), F. Baader gives as example the terminology

$$\begin{aligned} \text{Male} &= \neg \text{Female}, & \text{Human} &= \text{Male} \sqcup \text{Female}, & \text{Parent} &= \exists \text{child. Human}, \\ \text{NoDaughter} &= \forall \text{child. Male}, & \text{NoSon} &= \forall \text{child. Female}, \\ \text{and NoSmallChild} &= \forall \text{child. } \neg \text{Small} \end{aligned}$$

where **Small** and **Female** are primitive concepts. In the ordering resulting from the classification, the three concepts **NoDaughter**, **NoSon** and **NoSmallChild** are incomparable. The subsumption $\text{NoDaughter} \sqcap \text{NoSon} \sqsubseteq \text{NoSmallChild}$ cannot be deduced from the partially ordered set.

For including information about the subsumption-relationship between conjunctions, the classification can be extended with all conjunctions of the concepts of the terminology. Instead of testing all pairs of conjunctions for subsumption (which would not be effective, since in the worst case the number of concepts built by conjunction is exponential in the size of the terminology), Baader applies *attribute exploration* ([8], see also [9], [3]), an exploration tool of formal concept analysis. Attribute exploration produces questions of the kind “Is $C_1 \sqcap \dots \sqcap C_n$ subsumed by $D_1 \sqcap \dots \sqcap D_m$?” which are answered by the subsumption algorithm. The set of all suggested subsumptions being accepted by the subsumption algorithm is a minimal representation (called *Duquenne-Guigues-Basis*) of the semi-lattice of all possible conjunctions of the concepts in the TBox. Additionally this algorithm provides a list of “counterexamples” (\mathcal{I}, c) for all subsumptions that do not hold with respect to the terminology: For every pair $C_1 \sqcap \dots \sqcap C_n, D_1 \sqcap \dots \sqcap D_m$ of conjunctions of concepts of the TBox with $C_1 \sqcap \dots \sqcap C_n \not\sqsubseteq_{\mathcal{T}} D_1 \sqcap \dots \sqcap D_m$ there is a pair (\mathcal{I}, c) in the list such that $c \in (C_1 \sqcap \dots \sqcap C_n)^{\mathcal{I}} \setminus (D_1 \sqcap \dots \sqcap D_m)^{\mathcal{I}}$.

Since every complete semi-lattice is also a complete lattice, we can compute suprema in the resulting ordering. For instance, the supremum of **Male** and **Female** is **Human** in our example. Unfortunately, this does not imply $\text{Human} = \text{Male} \sqcup \text{Female}$ (but only $\text{Human} \sqsupseteq \text{Male} \sqcup \text{Female}$), since the supremum

in general does not correspond to the disjunction.¹ The subsumption $\text{Human} \sqsubseteq \text{Male} \sqcup \text{Female}$ cannot be deduced from the classification of all conjunctions alone, although it follows directly from the definition of Human in the TBox.

By replacing attribute exploration by *distributive concept exploration* ([17], [10]), the classification algorithm computes the complete lattice of all combinations of conjunctions and disjunctions of the concepts in the TBox. In particular, the supremum in the resulting lattice will correspond to the disjunction. The lattice will be represented by a minimal formal context (which can be stored for further computations). As in the previous case, the algorithm provides a list of counterexamples for all non valid subsumptions.

4 Computing the Conjunction-Disjunction-Lattice

The algorithm for the computation of the conjunction-disjunction-lattice generated by the concepts of a terminology uses the fact that this lattice is isomorphic to a suitable quotient lattice of the free bounded distributive lattice generated by the concepts. Hence the main task is to determine the corresponding congruence relation. Since free bounded distributive lattices grow exponentially, the algorithm does not calculate in this lattice, but splits up the task of determining the congruence relation. Therefore the tensor product for complete lattices ([23], see definition below) which is the coproduct in the category of completely distributive complete lattices is used. The equation $\text{FBD}(\{x_1, \dots, x_i\}) = \text{FBD}(\{x_1, \dots, x_{i-1}\}) \otimes \text{FBD}(\{x_i\})$ allows an iterative computation.

Starting with $i = 1$ the algorithm determines a lattice L_i that is isomorphic to the conjunction-disjunction-lattice generated by the first i concepts C_1, \dots, C_i of the terminology. The lattice L_i results from L_{i-1} by $L_i := (L_{i-1} \otimes \text{FBD}(\{C_i\})) / \Theta_i$, where L_0 is the two element lattice $\perp < \top$. The congruence relation Θ_i is determined by applying the subsumption algorithm. The lattice $L_{i-1} \otimes \text{FBD}(\{C_i\})$ is the lattice which respects all hierarchical dependencies between the first $i-1$ concepts, but no relationships to the concept C_i . The congruence Θ_i is then describing these relationships. Both congruence relations and tensor products can be defined by formal contexts representing the lattices. This allows an effective computation.

4.1 Tensor Products and Congruence Relations of Complete Lattices

The *tensor product* of two complete lattices L_1 and L_2 is defined to be the concept lattice $L_1 \otimes L_2 := \mathfrak{B}(L_1 \times L_2, L_1 \times L_2, \nabla)$ with $(x_1, x_2) \nabla (y_1, y_2) : \iff (x_1 \leq y_1 \text{ or } x_2 \leq y_2)$. We define the *direct product* of two contexts $\mathbb{K}_1 := (G_1, M_1, I_1)$ and $\mathbb{K}_2 := (G_2, M_2, I_2)$ to be the context $\mathbb{K}_1 \times \mathbb{K}_2 := (G_1 \times G_2, M_1 \times M_2, \nabla)$ with the incidence $(g_1, g_2) \nabla (m_1, m_2) : \iff ((g_1, m_1) \in I_1 \text{ or } (g_2, m_2) \in I_2)$.

¹ The supremum always subsumes the disjunction; in general the inverse does not hold.

The tensor product of two concept lattices is (up to isomorphism) just the concept lattice of the direct product of their contexts: $\mathfrak{B}(\mathbb{K}_1) \otimes \mathfrak{B}(\mathbb{K}_2) \cong \mathfrak{B}(\mathbb{K}_1 \times \mathbb{K}_2)$ (cf. [23]).

We say that a context is *distributive* if its concept lattice is distributive. All contexts in the following will be distributive reduced finite contexts. The direct product of distributive reduced contexts is again a distributive reduced context.

In reduced finite contexts every *congruence relation* corresponds to a *compatible subcontext*: A context (H, N, J) is called a *subcontext* of a context (G, M, I) if $H \subseteq G$, $N \subseteq M$ and $J = I \cap (H \times N)$. It is called *compatible* if for every concept (A, B) of (G, M, I) the pair $(A \cap H, B \cap N)$ is also a concept of the subcontext. Every compatible subcontext of a distributive reduced context is again a distributive reduced context (cf. [9]).

Factorizing a concept lattice is equivalent to deleting suitable rows and columns in the context (which generates a compatible subcontext). The rows and columns that have to be deleted can be described with the \swarrow -relation: For $g \in G$ and $m \in M$ we write $g \swarrow m$ if γg is minimal in γG with $\gamma g \not\leq \mu m$ and μm is maximal in μM with $\gamma g \not\leq \mu m$. In a distributive reduced finite context the \swarrow -relation is a bijection between the set of objects and the set of attributes, and the compatible subcontexts are exactly those of the form $(H, N, I \cap (H \times N))$ where $g \swarrow m$ implies $g \in H \iff m \in N$. The following theorem (cf. [17]) describes the correspondence between the compatible subcontexts and the congruence relations:

Theorem 1. *Let (G, M, I) be a distributive reduced finite context, $g \in G$ and $m \in M$ with $g \swarrow m$. Then $(A_1, B_1)\Theta(A_2, B_2) : \iff A_1 \setminus \{g\} = A_2 \setminus \{g\}$ ($\iff B_1 \setminus \{m\} = B_2 \setminus \{m\}$) defines the congruence relation on $\mathfrak{B}(G, M, I)$ that is generated by the pair $(\gamma g, \gamma g \wedge \mu m)$ (i. e., by forcing $\gamma g \leq \mu m$). The corresponding compatible subcontext is*

$$(G \setminus \{g\}, M \setminus \{m\}, I \cap ((G \setminus \{g\}) \times (M \setminus \{m\}))) .$$

For determining the congruence relation we have thus to compute for every pair $g \swarrow m$ if the subsumption $\gamma g \leq \mu m$ holds. For the computation of the \swarrow -relation, the algorithm uses the fact that the relation is inherited to compatible subcontexts, and that for every direct product of contexts the equivalence $(g_1, g_2) \swarrow (m_1, m_2) \iff (g_1 \swarrow m_1 \text{ and } g_2 \swarrow m_2)$ holds.

4.2 Classifying with Distributive Concept Exploration

In this subsection we explain the algorithm via the example given above. First we list the concepts of the terminology: $C_1 := \text{Female}$, $C_2 := \text{Male}$, $C_3 := \text{Human}$, \dots , $C_8 := \text{NoSmallChild}$. The concepts \top and \perp are considered in the first step of the computation.

The algorithm starts with the free bounded distributive lattice $\text{FBD}(\{C_1\})$ which is the three element chain shown in Fig. 1. For the two \swarrow in the context the subsumptions $\top \sqsubseteq \text{Female}$ and $\text{Female} \sqsubseteq \perp$ are tested with the subsumption

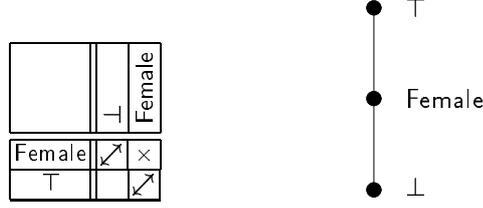


Fig. 1. The free bounded distributive lattice $\text{FBD}(\{C_1\})$ and its context representation

algorithm. The algorithm denies both and provides the two counterexamples (\mathcal{I}_1, c_1) and (\mathcal{I}_2, c_2) :

(\mathcal{I}_1, c_1)
$\text{dom}(\mathcal{I}_1) := \{\text{Otto}\}$
$\text{Female}^{\mathcal{I}_1} := \emptyset$
$\text{Small}^{\mathcal{I}_1} := \emptyset$
$\text{child}^{\mathcal{I}_1} := \emptyset$
$c_1 := \text{Otto}$

(\mathcal{I}_2, c_2)
$\text{dom}(\mathcal{I}_2) := \{\text{Tina}, \text{Tom}\}$
$\text{Female}^{\mathcal{I}_2} := \{\text{Tina}\}$
$\text{Small}^{\mathcal{I}_2} := \emptyset$
$\text{child}^{\mathcal{I}_2} := \emptyset$
$c_2 := \text{Tina}$

Hence there are no rows or columns to delete. We obtain the lattice L_1 describing the subsumption relationships between the three concepts \perp , \top , and **Female**. The lattice and the representing context \mathbb{K}_1 are shown in the upper left of Fig. 2. At the left of the context the counterexamples are listed.

Now the tensor-product of L_1 with $\text{FBD}(\{C_2\})$ is computed (see Fig. 2). The computation is only done on the context level, the line diagrams are only displayed for a better understanding. For the two counterexamples (\mathcal{I}_1, c_1) and (\mathcal{I}_2, c_2) , now the algorithm tests (by finite model-checking) whether $c_1 \in \text{Male}^{\mathcal{I}_1}$ and $c_2 \in \text{Male}^{\mathcal{I}_2}$. The answers “Yes” and “No”, resp., determine the place to put the counterexamples in the context \mathbb{K}'_1 . In the context we write $\text{Female} \sqcap \text{Male}$ for the object (Female, Male) and $\text{Female} \sqcup \text{Male}$ for the attribute (Female, Male), since this is exactly the interpretation of the relation ∇ in the definition of the direct product.

Next the congruence relation that describes the subsumption relationships of the concept **Male** to the three already computed concepts \perp , \top , and **Female** is computed. For two of the four \swarrow there are already counterexamples. For the other two \swarrow , the subsumption algorithm is asked the questions “Does $\text{Female} \sqcap \text{Male} \sqsubseteq \perp$ hold?” and “Does $\top \sqsubseteq \text{Female} \sqcup \text{Male}$ hold?”. This time both subsumptions are accepted, since the subsumption algorithm is not able to provide a counterexample. Hence the corresponding two lines and two columns have to be deleted. The resulting lattice is shown at the bottom of Fig. 3.

In this way the classification continues. The next step, for instance, with $C_3 = \text{Human}$, discovers that $\top = \text{Human}$, since the subsumption algorithm accepts the two subsumptions $\text{Male} \sqsubseteq \text{Female} \sqcup \text{Human}$ and $\text{Female} \sqsubseteq \text{Male} \sqcup \text{Human}$.

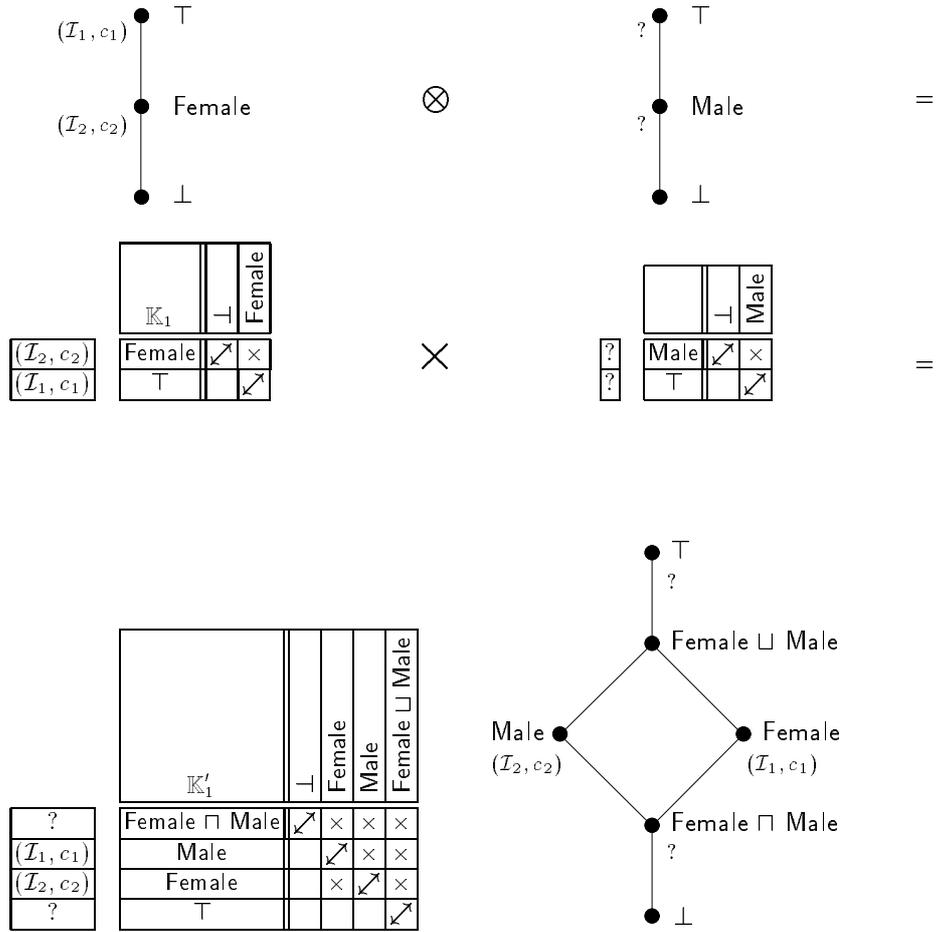


Fig. 2. The tensor-product

Hence the fact that every individual of a model of the terminology is a **Human** can directly be read from the result of the classification. The computation for our example ends with the eighth concept **NoSmallChild**. The result is a formal context with 44 objects and 44 attributes, and a list of 44 counterexamples.

5 Outlook

The algorithm can easily be modified such that it computes the Boolean lattice of all combinations of conjunctions, disjunctions, and negations of the concepts in the terminology, since the tensor-product is also the coproduct in the category of completely distributive complete Boolean algebras. In that case the free bounded

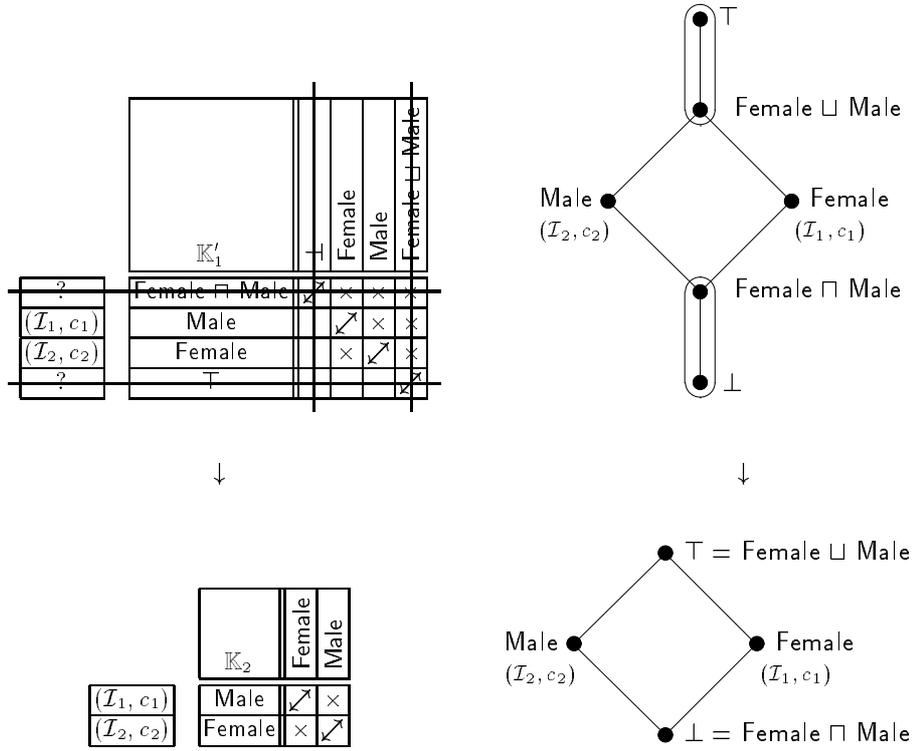


Fig. 3. Factorization of the tensor-product

distributive lattice $\text{FBD}(\{C_i\})$ has to be replaced by the free Boolean algebra $\text{FBA}(\{C_i\})$ (see Fig. 4). An interesting question is whether the classification can be extended further by existential and value restriction. There one encounters with new problems: The free algebra is infinite, and hence the desired result may be infinite, too. This could be overcome by restricting the length of the concept descriptions to be considered. Secondly these algebras have less algebraic structure than semi-lattices, lattices or Boolean algebras; and the quantifiers are less related to the subsumption ordering than conjunction, disjunction and negation.

The inference mechanisms presented in the last section and the one described in [1] show that combining techniques of description logic and formal concept analysis can provide interesting results. A further extension of these combinations seems desirable, especially for the development of conceptual knowledge systems. While description logics are more sophisticated in knowledge representation and inference, tools of formal concept analysis focus more on knowledge acquisition (cf. [16], [24]) and communication (cf. [20]). All four aspects

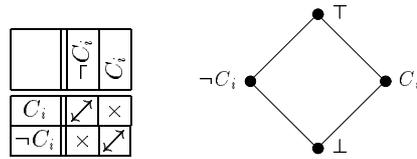


Fig. 4. The free Boolean algebra $FBA(\{C_i\})$ and its context representation

play a crucial role for conceptual knowledge systems. The management system TOSCANA ([20]) for *conceptual data systems* ([13], [19]) provides techniques for knowledge representation and communication. It is promising to examine how this system can be extended with a terminology in order to increase expressiveness and to treat incomplete knowledge.

References

1. F. Baader: Logische Methoden in der Wissensrepräsentation am Beispiel terminologischer Repräsentationssprachen. Preprint
2. F. Baader: Computing a minimal representation of the subsumption lattice of all conjunctions of concepts defined in a terminology. In: G. Ellis, R. A. Levinson, A. Fall, V. Dahl (eds.): *Proceedings of the International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency*. Santa Cruz, CA, USA, August 11–13, 1995, 168–178
3. P. Burmeister: ConImp – Programm zur formalen Begriffsanalyse einwertiger Kontexte. TH Darmstadt 1987 (latest version 1995)
4. P. Burmeister: Merkmalsimplikationen bei unvollständigem Wissen. In: W. Lex (ed.): *Arbeitstagung Begriffsanalyse und Künstliche Intelligenz*, Informatik-Bericht 89/3, Universität Clausthal-Zellerfeld, 1988, 15–46
5. G. Ellis, R. A. Levinson, A. Fall, V. Dahl (eds.): *Proceedings of the International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency*. Santa Cruz, CA, USA, August 11–13, 1995
6. G. Ellis, R. A. Levinson, W. Rich, J. F. Sowa (eds.): *Conceptual Structures: Applications, Implementation and Theory*. Third International Conference on Conceptual Structures, Santa Cruz, CA, USA, August 14–18, 1995, Springer, Berlin – Heidelberg – New York 1995
7. G. Ellis, R. A. Levinson, W. Rich, J. F. Sowa (eds.): *Supplementary Proceedings of the Third International Conference on Conceptual Structures*, Santa Cruz, CA, USA, August 14–18, 1995
8. B. Ganter: Algorithmen zur Begriffsanalyse. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse*. B. I.-Wissenschaftsverlag, Mannheim, Wien, Zürich 1987. 241–254
9. B. Ganter, R. Wille: *Formale Begriffsanalyse: Mathematische Grundlagen*. Springer, Heidelberg 1996
10. B. Groh: *Distributive Concept Exploration for Windows*, TH Darmstadt 1995

11. P. Luksch, R. Wille: A mathematical model for conceptual knowledge systems. In: H.-H. Bock, P. Ihm (eds.): *Classification, data analysis and knowledge organization*. Springer, Berlin 1991, 156–162
12. U. Priß: The formalization of WordNet by methods of relational concept analysis. In: C. Fellbaum (ed.): *WordNet: An electronic lexical database and some of its applications*. MITpress 1996 (to appear)
13. P. Scheich, M. Skorsky, F. Vogt, C. Wachter, R. Wille: Conceptual data systems. In: O. Opitz, B. Lausen, R. Klar (eds.): *Information and classification*. Springer, Heidelberg 1993, 72–84
14. M. Schmidt-Schauß, G. Smolka: Attribute concept descriptions with complements. *Artificial Intelligence* **48**, 1991, 1–26
15. G. Stumme: Boolesche Begriffe. Diplomarbeit, TH Darmstadt 1994
16. G. Stumme: Exploration tools in formal concept analysis. In: *Proceedings of the International Conference on Ordinal and Symbolic Data Analysis*, Paris, June 20–23, 1995, Springer, Heidelberg 1996, 31–44
17. G. Stumme: Knowledge acquisition by distributive concept exploration. In: G. Ellis, R. A. Levinson, W. Rich, J. F. Sowa (eds.): *Supplementary Proceedings of the Third International Conference on Conceptual Structures*, Santa Cruz, CA, USA, August 14–18, 1995, 98–111
18. G. Stumme: Attribute exploration with background implications and exceptions. In: H. H. Bock, W. Polasek (eds.): *Data analysis and information systems. Statistical and conceptual approaches*. Studies in classification, data analysis, and knowledge organization **7**, Springer, Heidelberg 1996, 457–469
19. F. Vogt, C. Wachter, R. Wille: Data analysis based on a conceptual file. In: H.-H. Bock, P. Ihm (eds.): *Classification, data analysis, and knowledge organization*. Springer, Heidelberg 1991, 131–140
20. F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. In: R. Tamassia, I. G. Tollis (eds.): *Graph Drawing '94*, Lecture Notes in Computer Sciences **894**, Springer, Heidelberg 1995, 226–233
21. H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe*. Kösel Verlag, München 1973, 191–209
22. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets*. Reidel, Dordrecht–Boston 1982, 445–470
23. R. Wille: Tensorial decomposition of concept lattices. In: *Order* **2**, 1985, 81–95
24. R. Wille: Knowledge acquisition by methods of formal concept analysis. In: E. Diday (ed.): *Data analysis, learning symbolic and numeric knowledge*. Nova Science Publisher, New York, Budapest 1989, 365–380
25. R. Wille: Local completeness of conceptual knowledge systems. In: E. Diday, Y. Lechevallier (eds.): *Symbolic-numeric data analysis and learning*. Nova Science Publisher, New York–Budapest 1991, 347–356
26. R. Wille: Concept lattices and conceptual knowledge systems. In: *Computers & Mathematics with applications* **23**, 1992, 493–515
27. M. Zickwolff: Begriffliche Wissenssysteme in der künstlichen Intelligenz. FB4-Preprint 1506, TH Darmstadt 1992