Contextual Boolean Logic: How did it develop?

Julia Klinger and Björn Vormbrock

Technische Universität Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D-64289 Darmstadt,
{jklinger,vormbrock}@mathematik.tu-darmstadt.de

Abstract. The aim of this paper is to explain the fundamental notions of Boolean Concept Logic and, based on this theory, to compare various approaches to Contextual Judgment Logic. First we motivate the basic definitions of semiconcept algebras and show how semiconcept algebras generalize to protoconcept algebras and double Boolean algebras. This enables us to give a brief presentation of the theories of concept graphs, semiconcept graphs, protoconcept graphs and concept graphs with cuts as specific approaches to a mathematical judgment logic. In addition, differences and common grounds of these contributions are discussed.

1 Introduction

In this paper, the development of Contextual Boolean Logic, in particular of Boolean Concept Logic and Boolean Judgment Logic is described. In the part dealing with Boolean Concept Logic it is shown how the basic definitions of algebras of semiconcept arise naturally from problems in practice. Based on this it is studied how and to what extent various approaches to Contextual Judgment Logic were developed. This paper consists of three more sections. In the second section an example of practical interest is used to illustrate the fundamental notions of semiconcept algebras. Furthermore, protoconcept algebras and double Boolean algebras are discussed. In the third section the logic systems of concept graphs, semiconcept graphs, protoconcept graphs and concept graphs with cuts are summarized. Finally, we give some perspectives for further research in the last section.

2 Boolean Concept Logic

In this section the expressiveness of Boolean Concept Logic as introduced in [Wi00] is demonstrated. By considering the problem of hotel selection, the fundamental definitions of semiconcept algebras are motivated and defined. This enables us to explain some advantages of Boolean Concept Logic in contrast to Formal Concept Analysis.

	guest house	hotel, standard	First Class	Luxus	telephone	sauna	hotel car park	covered parking space	price =< 50	price >= 50	price >= 75	price >=150	price >=200
Kempinski		×	×	×	×	×		×		×	×	×	×
Hilton		×	×		×	×	×	×		×	×	×	
Astron		×	×		×	×		×		×	×		
ibis Bastei		×			×					×			
ISSY	×						×	×	×				
Pension Reiche	×				×		×		×				

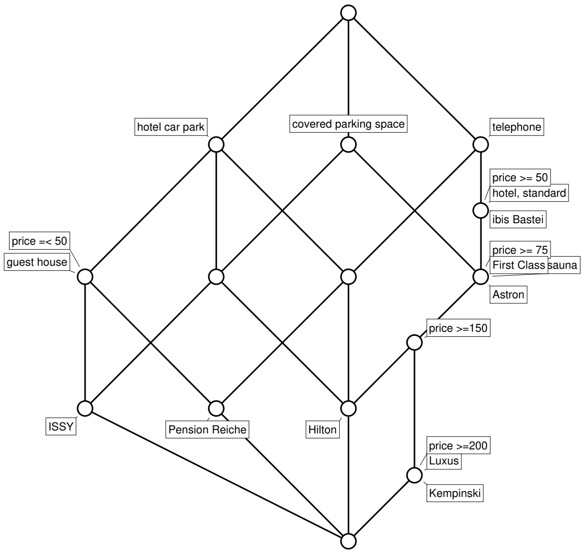


Fig. 1. The context $\mathbb{K}_0$ of hotels and guest houses and its corresponding concept lattice

2.1 The problem of choosing a suitable hotel

We assume that we want to stay in Dresden for a few days and look for accommodation that meets our needs. One approach to compare and to find suitable hotels would be to encode the information we obtain in a formal context and to use Formal Concept Analysis. The context in Figure 1 is based on a brochure of the Dresden tourist information office. Its objects are some of the hotels and guest houses in Dresden. The attributes 'guest house' and 'hotel' refer to the type of the accommodation. In particular, 'standard', 'First Class' and 'Luxus' are different categories for hotels. The attribute 'telephone' states that there are telephones in the rooms. If an object is incident to 'sauna', 'hotel car park' or 'covered parking space' then the corresponding services are offered. Prices in Euro are per night and for single rooms.

Now suppose we want to find the set H_1 of all hotels that are at least of category 'standard' and have a sauna. The concept lattice in Figure 1 shows that $H_1 = \{$Hilton, Kempinski and Astron$\}$, as these hotels form the extension of the formal concept generated by the attributes 'hotel' and 'sauna'. In addition we can derive that a night in any of these hotels costs at least 75 Euro, because the intent of this concept consists of 'hotel', 'sauna', 'telephone', 'covered parking space', 'price $\geq$ 50', and 'price $\geq$ 75'. However, we might as well want to exclude properties, for instance a price of 150 Euro or more. If we want to find the set H_2 of all hotels that are at least of category 'standard' and cost less than 150 Euro per night, it is natural to calculate the intersection of the derivation of 'standard' and the complement of the derivation of 'price $\geq$ 150':

$$H_2 = \text{standard}^I \cap (G \setminus (\text{price } \geq \text{ } 150)^I) = \{\text{Ibis Bastei, Astron}\}.$$

Hence, we find that the hotels 'Ibis Bastei' and 'Astron' are the only hotels in the context that satisfy our request. As before, the derivation $\{\text{Ibis Bastei, Astron}\}^I$ $= \{\text{telephone, hotel, standard, price} \geq 50\}$ yields that every hotel with the properties 'category standard' and 'price $\leq$ 150' also has telephone in each room and costs at least 50 Euro per night.

The set {Ibis Bastei, Astron} is not the extension of a formal concept of $\mathbb{K}_0$. In order to obtain this set by means of Formal Concept Analysis the original multi-valued context would have to be scaled inter-ordinally. The following two examples show reasonable questions where there is no natural way of scaling the original multi-valued context such that the answer may be expressed as the intent or the extent of a formal concept. In Section 2.2 it is shown how Boolean Concept Logic provides a language and the means to answer these questions. Therefore the introduction of negation and opposition to Formal Concept Analysis increases its expressiveness.

A different question could be: 'Which attributes has the Kempinski that distinguish it from the Hilton?'. Let H_3 denote the set of these attributes. In order to compute H_3, the derivation of 'Kempinski' is intersected with the complement of the derivation of 'Hilton':

$$H_3 = \text{Kempinski}^I \cap (M \setminus \text{Hilton}^I) = \{\text{luxus, price} \geq 200\}.$$

As a third example we take the following problem: If in a first step the set H_4 of all hotels and guest houses incident with 'telephone' is considered and in a second step H_5 is defined as the set of all elements of H_4 with a price of less than 50 Euro then we could ask which additional attributes are implied in the second step. Let H_6 denote this set of attributes. As above, the intersection of the derivation of 'telephone' and the complement of the derivation of 'price $\geq$ 50' is calculated:

$$\text{telephone}^I \cap (G \setminus (\text{price } \geq \text{ } 50)^I) = \{\text{Pension Reiche}\}.$$

Then we obtain H_6 as the intersection of the derivation of 'Pension Reiche' and the complement of the derivation of 'telephone':

$$H_6 = \text{Pension Reiche}^I \cap (M \setminus \text{ telephone}^I) = \{\text{guest house, hotel car park,} \\ \text{price} \leq 50\}.$$

2.2 The contributions of Boolean Concept Logic

In [Wi00] the set $\mathfrak{H}(\mathbb{K})$ of semiconcepts of a context $\mathbb{K} = (G, M, I)$ is defined as the set of all pairs (A, B) with $A \subseteq G$ and $B \subseteq M$ satisfying $A^I = B$ or $B^I = A$.

On this set the following operations are introduced:

$$\begin{aligned}
(A_1,B_1)\sqcap(A_2,B_2) &:= (A_1\cap A_2,(A_1\cap A_2)')\\
(A_1,B_1)\sqcup(A_2,B_2) &:= ((B_1\cap B_2)',B_1\cap B_2)\\
\neg(A_1,B_1) &:= (G\setminus A_1,(G\setminus A_1)')\\
\lrcorner(A_1,B_1) &:= ((M\setminus B_1)',M\setminus B_1)\\
\bot &:= (\emptyset,M)\\
\top &:= (G,\emptyset)
\end{aligned}$$

The set $\mathfrak{H}(\mathbb{K})$ with the operations $\sqcap,\sqcup,\neg,\lrcorner,\top,\bot$ is called the semiconcept algebra of $\mathbb{K}$ and denoted by $\underline{\mathfrak{H}}(\mathbb{K})$. Note that for a semiconcept x holds $x\sqcap x=x$ or $x\sqcup x=x$. Semiconcepts having the first property are called $\sqcap$-semiconcepts, those fulfilling the second are called $\sqcup$-semiconcepts. A semiconcept is a concept if and only if it is both a $\sqcup$- and a $\sqcap$-semiconcept. On semiconcept algebras an order relation $\sqsubseteq$ is defined by
$(A_1,B_1)\sqsubseteq(A_2,B_2) :\Leftrightarrow A_1\subseteq A_2$ and $B_1\supseteq B_2$.

The ideas used above to address our requests can easily be expressed in terms of semiconcepts. The set H_2 from 2.1 was obtained as the extension of the following semiconcept:

$$\begin{gathered}
(\{\text{standard}\}^I,\{\text{standard}\})\sqcap\neg(\{\text{price }\geq 150\}^I,\{\text{price }\geq 150\})=\\
(\{\text{Ibis Bastei, Astron}\},\{\text{hotel, standard, telephone, price }\geq 50\})
\end{gathered}$$

The sets H_3 and H_6 are the respective intents of the semiconcepts given by:

$$\begin{gathered}
(\{\text{Kempinski}\},\{\text{Kempinski}\}^I\sqcup\lrcorner(\{\text{Hilton}\},\{\text{Hilton}\}^I)=\\
(\{\text{Kempinski}\},\{\text{luxus, price }\geq 200\})
\end{gathered}$$

and

$$\begin{gathered}
((\text{telephone}^I,\text{telephone})\sqcap\neg(\text{price }\geq 150^I,\text{price }\geq 150))\\
\sqcup\lrcorner(\text{telephone}^I,\text{telephone})\\
=(\{\text{ISSY, Pension Reiche}\},\{\text{guest house, hotel car park, price }\leq 50\})
\end{gathered}$$

Thus Boolean Concept Logic provides an adequate language to express and to formalize the intuitive approach we used to deal with these problems. A development of this theory and an improvement of its methods will thereby enable us to understand and answer more complex problems as well.

Since semiconcept algebras are ordered sets, they can be represented as Hasse diagrams. Unfortunately, the number of semiconcepts of a concept with n objects and m attributes can be up to 2^{n+m}. In our example the semiconcept algebra consists of $2^{18}-14$ elements. As a smaller example, Figure 2 depicts the diagram of the subcontext consisting of all objects of $\mathbb{K}_0$ and the attributes 'guest house', 'hotel, standard', 'First Class' and 'Luxus'. The elements represented as filled circles are the concepts. The circles with the upper half filled stand for $\sqcup$-semiconcepts, those with filled lower half represent $\sqcap$-semiconcepts. The semiconcepts generated by single objects or attributes are labelled as well as their negations or oppositions.

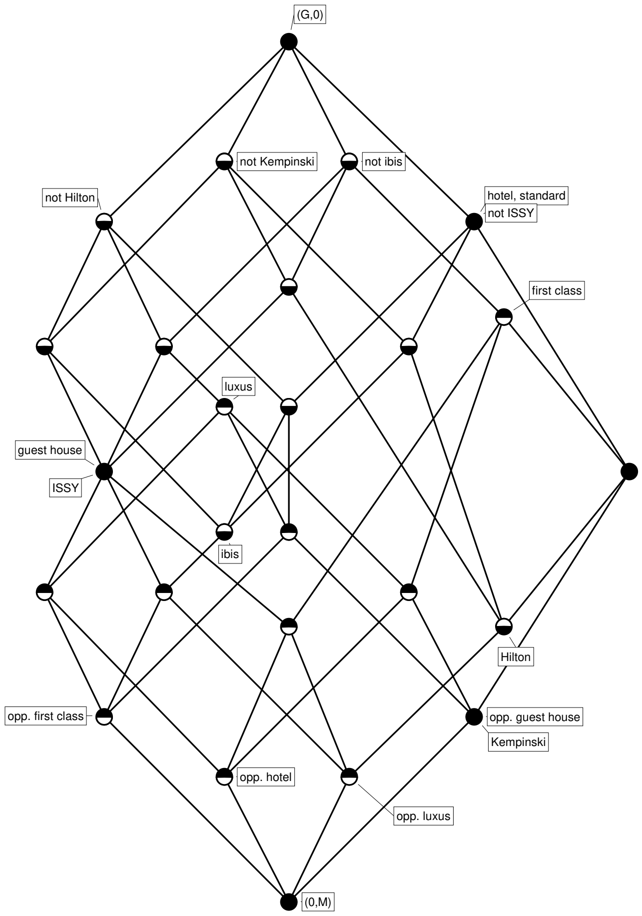


Fig. 2. Semiconcepts of the context restricted to the attributes 'guest house', 'hotel, standard', 'First Class' and 'Luxus'

2.3 Protoconcept algebras and double Boolean algebras

In order to develop the logic of semiconcept algebras, the variety generated by these algebras is needed. This variety must contain all direct products of semiconcept algebras. However, a direct product of semiconcept algebras is not necessarily a semiconcept algebra itself: Let (x, y) be an element of the product $(\mathfrak{H}(\mathbb{K}_1), \mathfrak{H}(\mathbb{K}_2))$ of the semiconcept algebras $\mathfrak{H}(\mathbb{K}_1)$ and $\mathfrak{H}(\mathbb{K}_2)$, and let x be a $\sqcap$-semiconcept of $\mathfrak{H}(\mathbb{K}_1)$ and y a $\sqcup$-semiconcept of $\mathfrak{H}(\mathbb{K}_2)$. Then (x, y) does not satisfy the condition $(x, y) \sqcap (x, y) = (x, y)$ or $(x, y) \sqcup (x, y) = (x, y)$ which holds in every semiconcept algebra. For two semiconcept algebras $\mathfrak{H}(\mathbb{K}_1)$ and $\mathfrak{H}(\mathbb{K}_2)$ the elements of the product $\mathfrak{H}(\mathbb{K}_1) \times \mathfrak{H}(\mathbb{K}_2)$ can be understood as pairs (A, B) consisting of a set of objects and a set of attributes of the context $\mathbb{K} := \mathbb{K}_1 + \mathbb{K}_2$ satisfying $A^{II} = B^I$. This leads to the definition of protoconcept algebras. In [Wi00] the set $\mathfrak{P}(\mathbb{K})$ of protoconcepts of the context $\mathbb{K} := (G, M, I)$ is introduced as the set of all pairs (A,B) with $A \subseteq G$ and $B \subseteq M$ satisfying $A^{II} = B^{I}$. On these sets the same operations as on semiconcept algebras are defined.

Nevertheless, protoconcept algebras do not form a variety either. The equational class generated by these algebras is the class of double Boolean algebras (cf. [Wi00]). These are defined as algebras $\underline{D} := (D, \sqcap, \sqcup, \neg, \lrcorner, \bot, \top)$ of type (2,2,1,1,0,0) satisfying the equations

1a) $(x \sqcap x) \sqcap y = x \sqcap y$	1b) $(x \sqcup x) \sqcup y = x \sqcup y$
2a) $x \sqcap y = y \sqcap x$	2b) $x \sqcup y = y \sqcup x$
3a) $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$	3b) $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
4a) $x \sqcap (x \sqcup y) = x \sqcap x$	4b) $x \sqcup (x \sqcap y) = x \sqcup x$
5a) $x \sqcap (x \underline{\sqcup} y) = x \sqcap x$	5b) $x \sqcup (x \overline{\sqcap} y) = x \sqcup x$
6a) $x \sqcap (y \underline{\sqcup} z) = (x \sqcap y) \underline{\sqcup} (x \sqcap z)$	6b) $x \sqcup (y \overline{\sqcap} z) = (x \sqcup y) \overline{\sqcap} (x \sqcup z)$
7a) $\neg\neg(x \sqcap y) = x \sqcap y$	7b) $\lrcorner\lrcorner(x \sqcup y) = x \sqcup y$
8a) $\neg(x \sqcap x) = \neg x$	8b) $\lrcorner(x \sqcup x) = \lrcorner x$
9a) $x \sqcap \neg x = \bot$	9b) $x \sqcup \lrcorner x = \top$
10a) $\neg\bot = \top \sqcap \top$	10b) $\lrcorner\top = \bot \sqcup \bot$
11a) $\neg\top = \bot$	11b) $\lrcorner\bot = \top$
12) $(x \sqcap x) \sqcup (x \sqcap x) = (x \sqcup x) \sqcap (x \sqcup x)$	

with the operations $\underline{\sqcup}, \overline{\sqcap}, \overline{\top}, \underline{\bot}$ defined as $x \underline{\sqcup} y := \neg(\neg x \sqcap \neg y)$, $x \overline{\sqcap} y := \lrcorner(\lrcorner x \sqcup \lrcorner y)$, $\overline{\top} := \neg\bot$ and $\underline{\bot} := \lrcorner\top$.

Information on the structure of double Boolean algebras can be found in [HLSW00] and [Wi00]. Congruence relations on these were investigated in [Vo02].

3 Contextual Judgment Logic

Taking the traditional philosophical understanding into account, we regard judgments as valid propositions, hence as meaningful combinations of concepts. This shall be reflected in the theory of concept graphs as our formalization of the doctrine of judgments: We demand that the graphs are satisfiable, and therefore that the definition does not allow the construction of a self-contradictory graph. Three of the theories summarized in this section support this understanding, namely the logic systems of concept graphs, semiconcept graphs and protoconcept graphs. Moreover, these theories extend each other: the system of concept graphs is the most fundamental and the best developed theory, based on it is the theory of semiconcept graphs, and finally protoconcept graphs are the most general entities. The theory of concept graphs with cuts, however, has a different focus. The aim is equivalence to first order predicate logic, hence the introduction of a global negation is needed. Obviously, one of the consequences of this approach is that the construction of self-contradictory graphs is possible.

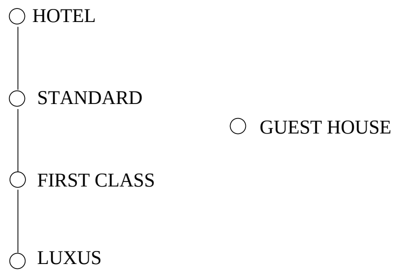
The following four approaches are, different to Wille's view of graphs as semantical structures, based on the separation of syntax and semantics as it is common in mathematical logic. This is reflected by the manner the theories are developed: First an alphabet is defined, then syntactical graphs are introduced

$\mathcal{G}$:= {kempinski, hilton, astron, bastei, ibis bastei, issy, pension reiche}

$\mathcal{C}$:= {HOTEL, STANDARD, FIRST CLASS,LUXUS, GUEST HOUSE}

$\mathcal{R}$:= {closer to airport, closer to train station}

$\leq_{\mathcal{C}}$:=



$\leq_{\mathcal{R}} := \mathrm{id}_{\mathcal{R}}$

Fig. 3. Example for a basic alphabet

	closer to Dresden Airport	closer to a train station
(Kempinski, Hilton)	×	
(Kempinski, Ibis Bastei)	×	
(Kempinski, ISSY)		×
(Kempinski, Pension Reiche)		×
(Hilton, ISSY)		×
(Hilton, Pension Reiche)	×	×
(Astron, Kempinski)	×	×
(Astron, Hilton)	×	×
(Astron, Ibis Bastei)	×	
(Astron, ISSY)	×	×
(Astron, Pension Reiche)	×	×
(Ibis Bastei, Kempinski)		×
(Ibis Bastei, Hilton)		×
(Ibis Bastei, Astron)		×
(Ibis Bastei, ISSY)		×
(Ibis Bastei, Pension Reiche)	×	×
(ISSY, Kempinski)	×	
(ISSY, Hilton)	×	
(ISSY, Ibis Bastei)	×	
(ISSY, Pension Reiche)	×	
(Pension Reiche, ISSY)		×

Fig. 4. The context $\mathbb{K}_2$

as mathematical structures and afterwards the semantics is defined via context-interpretations of alphabets and the notion of validity. In each of the following four subsections, we will briefly summarize one theory, however, we will restrict ourselves to the non-existential cases.

There are several structures occurring in each of the theories. In order to stay as informal as possible in the following paragraphs, we name some of them here: First of all, the underlying structure for syntactical concept graphs and its two extensions is that of a relational graph, which is a triple (V, E, ν) consisting of two disjoint sets V and E whose elements are called vertices and edges, respectively, and a function $\nu\colon E \to \bigcup_{k\in\mathbb{N}} V^k$ which maps each edge to the ordered tuple of its adjacent vertices. Moreover, the notion of a power context family is basic for all four approaches: A power context family $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \mathbb{K}_2, \dots)$ is a family of contexts $\mathbb{K}_k := (G_k, M_k, I_k)$ with $G_0 \neq \emptyset$ and $G_k \subseteq (G_0)^k$ for each $k \in \mathbb{N}$.

In order to explain the different approaches to judgment logic, we continue the example from Section 2. To distinguish the syntactical names from the elements of the power context family used for the interpretation, we employ different capitalizations.

3.1 Concept Graphs

The approach in [Wi97] was adopted and both modified and extended by Prediger. In [Pr98b], she introduced concept graphs as *syntactical* constructs and equipped them with an explicit contextual semantics instead of the traditional implicit semantics via a translation into predicate logic (see for instance [So84]).

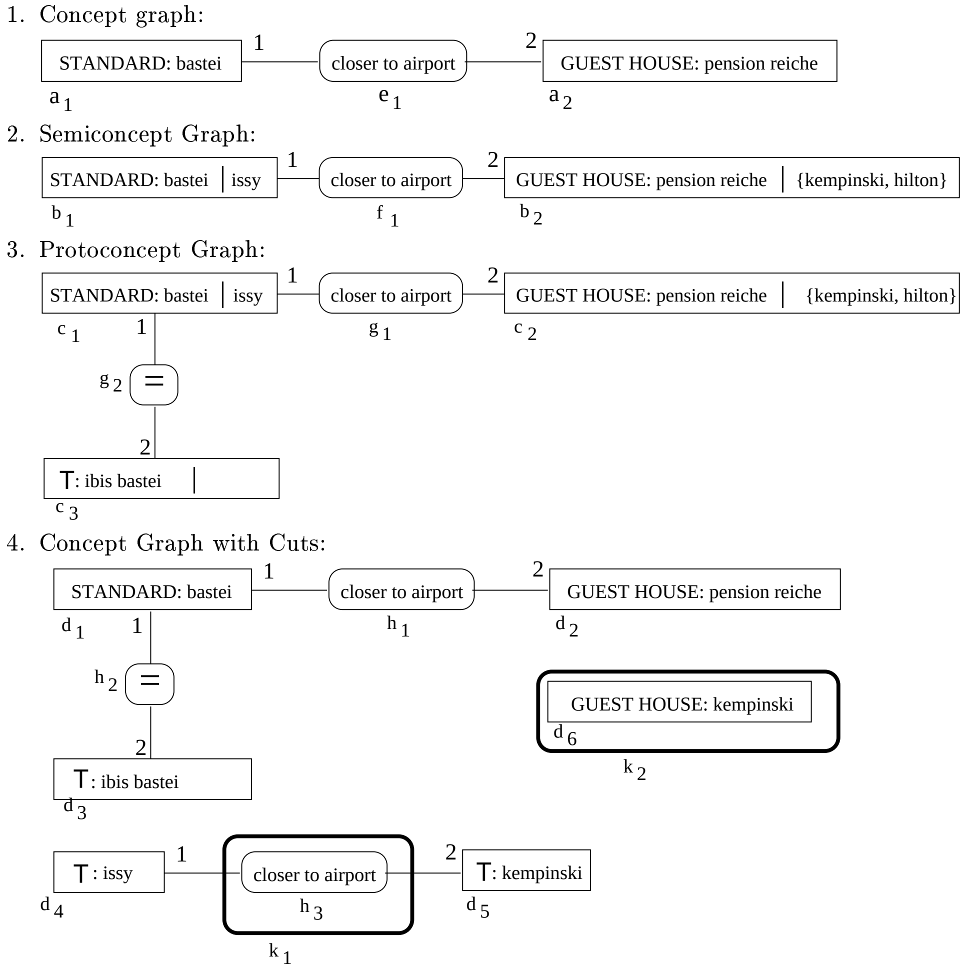


Fig. 5. Examples for graphs

The first step towards a syntactical implementation of concept graphs was the definition of an alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ consisting of a set $\mathcal{G}$ of object names, an ordered set $\mathcal{C}$ of concept names and a family $\mathcal{R}$ of ordered sets of relation names. In Figure 3 a sample for such an alphabet is given. Syntactical concept graphs over an alphabet were then introduced as mathematical structures of the form $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$, consisting of a relational graph, a function κ assigning concept names to vertices and relation names to edges, and a function ρ which assigns non-empty sets of object names to the vertices (as references). Figure 5.1 shows an example for a syntactical concept graph over the alphabet depicted in Figure 3.

For the semantics, the names of the alphabet are interpreted in a given power context family $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \dots)$ via a so called interpretation λ: This interpretation specifies how the syntactical elements of the alphabet are related to elements of $\overrightarrow{\mathbb{K}}$: object names are mapped to objects of $\mathbb{K}_0$, concept names to concepts of $\mathbb{K}_0$ and relation names to elements of $\mathfrak{B}(\mathbb{K}_k)$. Moreover, the orders specified in the alphabet are preserved. The resulting structures $(\overrightarrow{\mathbb{K}}, \lambda)$ are called context-interpretations. As an example consider the power context family $(\mathbb{K}_0, \mathbb{K}_2)$ consisting of the contexts in Figure 1 and Figure 4. An interpretation of the alphabet shown in Figure 3 may be defined as follows: the object names except for 'bastei' are mapped to the objects with the same name, 'bastei' is mapped to the the object 'Ibis Bastei'. Each concept name is interpreted as the attribute concept generated (neglecting capitalization) by the same name. For instance, the name 'STANDARD' is mapped to μ(standard). The relation name 'closer to airport' is interpreted as the attribute concept μ(closer to Dresden Airport), and 'closer to train station' as μ(closer to a train station). Now we say that a concept graph is valid in a context-interpretation $(\overrightarrow{\mathbb{K}}, \lambda)$ (and call $(\overrightarrow{\mathbb{K}}, \lambda)$ a model for the graph) if the so called vertex- and edge condition for the vertices respectively edges are both satisfied. The vertex condition for a vertex v is fulfilled if the interpreted object names of v belong to the extent of the interpreted concept name of that vertex. Similarly, the edge condition for an edge e holds if the objects along e are in the relation concept assigned to that edge. It is easy to see that the concept graph in Figure 5.1 is valid in $((\mathbb{K}_0, \mathbb{K}_2), \lambda)$.

The definition of concept graphs yields their satisfiability as well as decidability: For each two graphs over the same alphabet one can decide whether one entails the other or not. Moreover, Prediger discussed several extensions of simple concept graphs. She included generic markers and nestings into the theory (see [Pr98b], [Pr00]; for the semantic approach see [Wi98]). With respect to conclusion logic a sound and complete calculus for non-existential concept graphs and each of the extensions was provided. Moreover, for each concept graph a standard model was defined which codes exactly the same information as the graph, and, vice versa, the so called standard graph of a power context family was introduced.

3.2 Semiconcept Graphs

Semiconcept graphs were proposed in [Wi01]. They extend the theory of concept graphs by including negation on the level of concepts and relations. Again, the graphs were defined as semantical structures. Similarly to Prediger's approach, in [Kl01a] a strict separation in syntax and semantics was introduced. First the notion of an alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{S}, \mathcal{R})$ is provided, which extends the definition of alphabets from the previous section by replacing the set $\mathcal{C}$ of concept names by a set $\mathcal{S}$ of semiconcept names. Setting $\mathcal{S} := \mathcal{C}$, the alphabet in Figure 3 may serve as an example for an alphabet for semiconcept graphs as well. Then a syntactical semiconcept graph is defined in accordance to concept graphs as a tuple $(V, E, \nu, \kappa, \rho)$ such that (V, E, ν) is a relational graph and κ maps semiconcept names and relation names to the vertices and edges, respectively. Now ρ maps to each vertex *two* sets of object names, namely one set of positive and one set of negative references. There are four conditions for ρ, ensuring that for both vertices and edges there is at least one reference and that the graph is not self-contradictory. Hence, in comparison with concept graphs the main difference lies in the definition of ρ. This is resembled by the rule of how these graphs are read: The semiconcept graph in Figure 5.2 represents that 'bastei' is STANDARD and 'issy' is not, 'pension reiche' is 'GUEST HOUSE' while both 'kempinski' and 'hilton' are not, and 'bastei' is 'closer to airport' than 'pension reiche' while 'issy' is not 'closer to airport' than both 'kempinski' and 'hilton'.

For the semantics, the alphabet is interpreted in a power context family $\vec{\mathbb{K}}$ by an interpretation λ. As before, object names are mapped to objects of $\mathbb{K}_0$, however, semiconcept names are interpreted as $\sqcap$-semiconcepts of $\mathbb{K}_0$ instead of concepts. Moreover, k-ary relation names are interpreted as $\sqcap$-semiconcepts of $\mathbb{K}_k$. Since each concept is a $\sqcap$-semiconcept, the interpretation defined in the previous section may serve as an example for semiconcept graphs as well. For defining the validity of a semiconcept graph in a context-interpretation $(\vec{\mathbb{K}}, \lambda)$, we have to modify the vertex and the edge condition in order to capture the negation. Therefore, instead of just checking if the interpreted positive references of a vertex are in the interpreted semiconcept name, we additionally check whether the objects assigned to the negative references are in the negation of that semiconcept. Similarly, the edge condition consists of a positive and a negative part. We may now check if the sample graph in Figure 5.2 is valid in $((\mathbb{K}_0, \mathbb{K}_2), \lambda)$: It is easy to see that the positive vertex- and edge condition are satisfied. The negative conditions are satisfied as well. Consider for instance the vertex b_1: We have to check that the image of the object name 'issy' (thus the object ISSY) is in the extension of the semiconcept $\neg\mu(\text{Standard})$, which is equal to $\{\text{Pension Reiche, ISSY}\}$.

The definition of semiconcept graphs yields their satisfiability. Furthermore, it was shown via so called standard power context families that decidability is guaranteed. As an extension, variables were introduced as referents, yielding existential quantification (cf. [Kl01b],[Kl02]). Again, satisfiability and decidability were shown. Finally, for each power context family the graph representing the same information was provided.

3.3 Protoconcept Graphs

The goal was to develop a theory of concept graphs with negation comprising concept graphs, semiconcept graphs and suitable syntactical elements of concept graphs with cuts in such a way that our understanding of judgments as valid propositions remains supported. As semantical structures, protoconcepts graphs were introduced in [Wi02]. Based on this, the separation of syntax from semantics was described: Similar to the previous two approaches, first an alphabet $\mathcal{A} := (\mathcal{G}, \mathcal{P}, \mathcal{R})$ is defined. This time, however, the set $\mathcal{P}$ of protoconcept names contains a well-distinguished smallest element $\perp$ and a greatest element $\top$. Moreover, $\mathcal{R}$ includes a special binary relation name $=$. Therefore, the alphabet in Figure 3 has to be modified in order to obtain a suitable example: We take $\mathcal{A}' := (\mathcal{G}, \mathcal{P}, \mathcal{R}')$ with $\mathcal{P} := \mathcal{C} \cup \{\perp, \top\}$ and $\mathcal{R}' := \mathcal{R} \cup \{=\}$. Moreover, the orders on $\mathcal{C}$ and $\mathcal{R}$ are extended to orders on $\mathcal{P}$ and $\mathcal{R}'$ in a natural way. Syntactical protoconcept graphs are defined essentially in the same way as semiconcept graphs, though the use of the relation name $=$ gives rise to different and more complex conditions for ρ in order to ensure satisfiability. Figure 5.3 shows a syntactical protoconcept graph, which can be read in the same way as the graph in 5.2. In addition, it states that 'bastei' is equal to 'ibis bastei'. The semantics follows the same principles as in the previous sections. First an interpretation λ maps object names to objects, protoconcept names to protoconcepts of $\mathbb{K}_0$ and relation names to protoconcepts of contexts $\mathbb{K}_k$. Moreover, the special syntactical elements of the alphabet have to be respected by the interpretation: $\perp$ and $\top$ are mapped to the smallest and the greatest element of $\mathfrak{P}(\mathbb{K}_0)$, respectively, and the relation name $=$ is mapped to a protoconcept whose extension is $\{(g,g) \mid g \in G_0\}$. Then validity of a protoconcept graph is checked in the same way as for semiconcept graphs. Again, the graph depicted in 5.3 is a protoconcept graph of $((\mathbb{K}_0, \mathbb{K}_2), \lambda)$.

Via so called standard power context families, satisfiability and decidability were shown. Moreover, a sound and complete calculus was introduced. Up till now, no syntactical extensions for protoconcept graphs have been studied. However, in [Wi02] (semantical) existential protoconcept graphs were defined as protoconcept graphs of free X-extensions of power context families (where X is a set of variables).

3.4 Concept Graphs with Cuts

Concept graphs with cuts were introduced in [Da02] with the goal to obtain a logic system of concept graphs whose expressiveness matches first order predicate logic. This requires negation to be treated as a logical operator on the level of propositions, which was done by adopting so called cuts from Peirce's theory of existential graphs. After introducing non-existential concept graphs with cuts, Dau extended the theory by including existential quantifiers and was then able to show the desired equivalence (see [Da01],[Da02]).

In a similar way as for concept graphs, an alphabet is defined as a triple $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ such that there is a special concept name $\top$ which is the greatest element of $\mathcal{C}$ and a well-distinguished relation name $=$ of arity 2. Hence, the alphabet in Figure 3, modified by including $\top$ in $\mathcal{C}$, the binary relation name $=$ in $\mathcal{R}$, and extending the orders on the sets of concept and of relation names accordingly, yields an example of an alphabet for concept graphs with cuts. Next, a relational graph with cuts is defined as a tuple $(V, E, \nu, \top, \mathit{Cut}, \mathit{area})$ where (V, E, ν) is a relational graph, $\top$ is a single element called the sheet of assertion (which, thinking diagrammatically, can be understood as the sheet on which the graph is drawn), *Cut* is the set of cuts and *area* is a map which defines for each cut c the elements of the graph (thus the vertices, edges and other cuts) which are 'within' c. When picturing a concept graph, the cuts of the concept graph are usually drawn as bold ovals. Hence, if we consider the graph depicted in Figure 5.4 as an example, we have $\mathit{Cut} := \{k_1, k_2\}$, $\mathit{area}(k_1) = \{h_3\}$ and $\mathit{area}(k_2) = \{d_6\}$. Relational graphs with cuts satisfy several conditions, which can be reread in [Da02]. Most importantly, cuts do not intersect. A syntactical concept graph with cuts is then defined as a relational graph with cuts equipped with two functions κ and ρ. Again, the function κ maps concept names to the vertices and relation names to the edges, however, ρ does not map sets of object names to the vertices but single elements of $\mathcal{G}$. As already mentioned, Figure 5.4 depicts a sample graph. The subgraphs not containing cuts are read in the same way as concept graphs. However, each cut negates everything within it, so the bottommost graph is read: 'kempinski' and 'issy' are '$\top$' and 'kempinski' is not 'closer to airport' than 'issy'.

For the semantics, interpretations of alphabets in power context families are defined in the same way as in Section 3.1, except for the special elements $\top$ and $=$. Similar to the approach in 3.3, λ maps $\top$ to the greatest element of $\underline{\mathfrak{B}}(\mathbb{K}_0)$ and $=$ is mapped to a concept whose extension is $\{(g,g) \mid g \in G_0\}$. Next, validity is defined: In contrast to the graphs discussed in the previous sections, a concept graph $(V, E, \nu, \top, \mathit{Cut}, \mathit{area}, \kappa, \rho)$ is evaluated *inductively* in a context-interpretation. The induction is carried out on the tree $\mathit{Cut} \cup \{\top\}$. The main idea of this approach, which is closely related to the endoporeutic method of Peirce, is that in order for everything enclosed by a cut to be valid in a power context family, all cuts directly enclosed by it are not allowed to be valid (this is called the cut condition). Therefore, for each cut, the vertex, the edge and the cut condition are checked. If the sheet of assertion is valid, then the graph is, too. It is easy to check that the graph of Figure 5.4 is indeed valid in $(\mathbb{K}_0, \mathbb{K}_2)$.

A sound and complete calculus based on Peirce's work was defined for concept graphs with cuts both with and without generic markers. Moreover, in [Da01], translations of existential concept graphs with cuts to first order predicate logic and vice versa were given, yielding the equivalence of these two logic systems (see also [Da02]). Obviously, this implies that we have neither satisfiability nor decidability.

3.5 Discussion

First we note that the theories presented in the Sections 3.1 to 3.3 are indeed successively extending each other[1]:

Each non-existential concept graph $\mathfrak{G} = (V, E, \nu, \kappa, \rho)$ over the alphabet $\mathcal{A} = (\mathcal{G}, \mathcal{C}, \mathcal{R})$ can be understood as a semiconcept graph $\mathfrak{G} = (V, E, \nu, \kappa, \rho')$ with $\rho'(v) = (\rho(v), \emptyset)$ over the alphabet $\mathcal{A}' = (\mathcal{G}, \mathcal{S}, \mathcal{R})$ with $\mathcal{S} := \mathcal{C}$. Every interpretation of $\mathcal{A}$ yields an interpretation of $\mathcal{A}'$ simply by extending the ranges of the interpretation mapping λ for $\mathcal{C}$ and $\mathcal{R}$ from the concept lattices to the semiconcept algebras of the corresponding contexts. This interpretation of $\mathcal{A}'$ is denoted by λ'. Thus, whenever $\mathfrak{G}$ is valid in a context-interpretation $(\vec{\mathbb{K}}, \lambda)$, the graph $\mathfrak{G}'$ is valid in $(\vec{\mathbb{K}}, \lambda')$.

[1] We say that a theory is extended by another one if each graph of the first theory can be translated to a graph of the extension, such that the new graph represents the same relationships, and whenever the first graph is valid in a context-interpretation $(\vec{\mathbb{K}}, \lambda)$, the graph in the extension is valid in $(\vec{\mathbb{K}}, \lambda')$ (where λ' is derived from λ in a natural way).

Similarly, we obtain that each non-existential semiconcept graph $\mathfrak{G}$ over an alphabet $\mathcal{A} = (\mathcal{G}, \mathcal{S}, \mathcal{R})$ corresponds to a protoconcept graph: We set $\mathcal{A}' := (\mathcal{G}, \mathcal{P}, \mathcal{R}')$ with $\mathcal{P} := \mathcal{S} \cup \{\top, \bot\}$ and $\mathcal{R}' := \mathcal{R} \cup \{=\}$, $\mathfrak{G}' := \mathfrak{G}$ and extend the ranges of the interpretation functions accordingly while mapping the new syntactical elements to the corresponding protoconcepts.

The theory of concept graphs with cuts, however, does not generalize the theory of protoconcept graphs: Syntactically, for a given protoconcept graph we can construct a concept graph with cuts expressing the same information. In general, it will have more vertices and edges, because Dau's theory allows only single objects names as reference, whereas sets of object names are assigned to vertices of protoconcept graphs. Semantically, however, the fact that context-interpretations for protoconcept graphs map protoconcept names to protoconcepts and interpretations for concept graphs with cuts assign concepts to the corresponding elements of the alphabet poses a problem. Consider for instance the protoconcept graph $[\mathtt{P}\colon \mathtt{g} \mid \mathtt{h}]$ and a context $\mathbb{K}_0 := (G, M, I)$ with $\{g, h\} \subseteq G$ and $g^I = h^I$. We now define the interpretation λ such that P is mapped to the protoconcept (g, g^I), and both g and h are mapped to the object with the same name. Then the protoconcept graph is valid in $(\mathbb{K}_0, \lambda)$. However, every concept whose extension includes g automatically contains h as well. Hence, there cannot be a context-interpretation for a concept graph with cuts which expresses the same as the graph above. Thus, though the protoconcept graph is a graph of $(\mathbb{K}_0, \lambda)$, the corresponding concept graph with cuts would not be.

Since the existential quantifiers are formalized differently in these approaches (either by introducing the generic marker $*$ or by allowing variables as references), this discussion was restricted to non-existential graphs.

4 Work in Progress

Presently, methods for the diagrammatic representation of semiconcept algebras are developed and the algebraic theory of double Boolean algebras is investigated further. Moreover, there are several extensions for concept graphs which are currently worked on. In [DH03a] and [DH03b], Dau and Hereth Correia discuss nested concept graphs with cuts as query graphs. These graphs can serve as queries for relational databases and have the whole expressiveness of SQL, including set-functions like `AVG` and `COUNT`. Moreover, protoconcept graphs are about to be extended by introducing variables as referents in order to obtain existential quantification. The next step would then be to introduce nestings. Finally, in [SW03], a semantic approach for concept graphs with subdivision (generalizing nested concept graphs) is introduced.

References

[Da01] F. Dau: Concept Graphs and Predicate Logic. In: H. S. Delugach, G. Stumme (Eds.): Conceptual Structures: Broadening the Base, Springer Verlag, Berlin–New York 2001, 72-86.

[Da02] F. Dau: The Logic System of Concept Graphs with Negations (and its Relationship to Predicate Logic), Dissertation. FB Mathematik, TU Darmstadt 2002. For the submitted version see `http://www.mathematik.tu-darmstadt.de/~dau`

[Da03] F. Dau: Concept Graphs without Negations: Standardmodels and Standardgraphs. FB4-Preprint, TU Darmstadt, 2003.

[DH03a] F. Dau, J. Hereth Correia: Nested Concept Graphs: Mathematical Foundations. FB4-Preprint, TU Darmstadt, 2003.

[DH03b] F. Dau, J. Hereth Correia: Nested Concept Graphs: Applications for Databases. FB4-Preprint, TU Darmstadt, 2003.

[HLSW00] C. Herrmann, P. Luksch, M. Skorsky, R. Wille: Algebras of Semiconcepts and Double Boolean Algebras. Contributions to General Algebra 13, 2000.

[Kl01a] J. Klinger: Simple Semiconcept Graphs: a Boolean Logic Approach. In: H. S. Delugach, G. Stumme (Eds.): Conceptual Structures: Broadening the Base, Springer Verlag, Berlin–New York 2001, 115-128.

[Kl01b] J. Klinger: Semiconcept Graphs: Syntax and Semantics, Diplomarbeit, FB Mathematik, TU Darmstadt 2001.

[Kl02] J. Klinger: Semiconcept Graphs with Variables. In: U. Priss, D. Corbett, G. Angelova (Eds.): Conceptual Structures: Integration and Interfaces, Springer Verlag, Berlin–New York 2002, 382-396.

[Pr98a] S. Prediger: Kontextuelle Urteilslogik mit Begriffsgraphen, Shaker Verlag, Aachen 1998.

[Pr98b] S. Prediger: Simple Concept Graphs: A Logic Approach. In: M.-L. Mugnier, M. Chein (Eds): Conceptual Structures: Theory, Tools and Application, Springer Verlag, Berlin–Heidelberg 1998, 225-239.

[Pr00] S. Prediger: Nested Concept Graphs and Triadic Power Context Families: A Situation-Based Contextual Approach. In: B. Ganter, G. W. Mineau (Eds.): Conceptual Structures: Logical, Linguistic, and Computational Issues, Springer Verlag, Berlin–New York 2000, 249-262.

[SW03] L. Schoolmann, R. Wille: Concept Graphs with Subdivision: a Semantic Approach. FB4-Preprint, TU Darmstadt, 2003.

[So84] J. F. Sowa: Conceptual Structures: Information Processing in Mind and Machine. Adison-Wesley, Reading 1984.

[Vo02] B. Vormbrock: Kongruenzrelationen auf doppelt-booleschen Algebren. Diplomarbeit, FB Mathematik, TU Darmstadt 2002.

[Wi97] R. Wille: Conceptual Graphs and Formal Concept Analysis. In D. Lukose, H. Delugach, M. Keeler, L. Searle, J. Sowa (eds.): Conceptual Structures: Fullfilling Peirce's Dream. Springer, Berlin - Heidelberg - New York 1997, 290 - 303.

[Wi98] R. Wille: Triadic Concept Graphs. In: M.-L. Mugnier, M. Chein (Eds): Conceptual Structures: Theory, Tools and Application, Springer Verlag, Berlin–Heidelberg 1998, 194-208.

[Wi00] R. Wille: Boolean Concept Logic. In: B. Ganter, G.W. Mineau (Eds.): Conceptual Structures: Logical, Linguistic, and Computational Issues, Springer Verlag, Berlin–New York 2000, 317-331.

[Wi01] R. Wille: Boolean Judgment Logic. In: H. S. Delugach, G. Stumme (Eds.): Conceptual Structures: Broadening the Base, Springer Verlag, Berlin–New York 2001, 115-128.

[Wi02] R. Wille: Existential Graphs of Power Context Families. In: U. Priss, D. Corbett, G. Angelova (Eds.): Conceptual Structures: Integration and Interfaces, Springer Verlag, Berlin–New York 2002, 382-396.