Background Knowledge in Concept Graphs

Frithjof Dau

Technische Universität Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D-64289 Darmstadt, dau@mathematik.tu-darmstadt.de

Abstract. Traditional logic can be understood as the investigation of the three main essential functions of thinking – concepts, judgements and conclusions. In the last years, in a new research field termed *Contextual Logic*, a mathematical theory of this logic is elaborated. Concepts have already been mathematically elaborated by Formal Concept Analysis. Judgements and Conclusions can be expressed by so-called *Concept Graphs*, which are built upon families of formal contexts.

There are two approaches to concept graphs: A semantical approach, which investigates the theory of concept graphs in an algebraic manner, and a logical approach, which focuses on derivation rules for concept graphs, relying on a separation between syntax and semantics. In [26], Wille introduced two forms of complex implications (object implications and concept implications) to the semantical approach. In this paper it is investigated how these implications can be incorporated into the logical approach.

1 Introduction and Basic Definitions

The traditional philosophical goal of logic is the investigation of the forms of thinking. I. Kant explained this understanding of logic as "the theory of the three main essential functions of thinking – concepts, judgements and conclusions". In the last years, a new research field termed *Contextual Logic (CL)* came up, where a mathematical theory of traditional logic is elaborated. The understanding of concepts in the line of traditional logic has already been adopted and successfully mathematized by Wille's *Formal Concept Analysis (FCA)*, so the question of how to proceed with judgements and conclusions arose.

John Sowa developed on the basis of Peirce's *Existential Graphs* and the semantic networks of artificial intelligence the theory of *Conceptual Graphs (CGs)*. These graphs are a diagrammatic system of logic whose purpose is 'to express meaning in a form that is logically precise, humanly readable, and computationally tractable' (see [18]). Thus, conceptual graphs can considered to be formal representations of judgements. Moreover, Sowa incorporated deduction rules into his system which allow to obtain new graphs from given ones. Hence, this theory offers a formalization of conclusions as well.

Both FCA and CGs have been used for knowledge representation and processing. They have a common philosophical background, based on Peirce's pragmatism. Particularly, a main goal of both systems is the support of human, rational communication. Thus, for a mathematization of judgements and conclusions, associating FCA and CGs turned out to be a promising approach.

The combination of FCA and CGs yields mathematically defined structures termed *Concept Graphs*. The system of CGs is a very comprehensive and complex system, not mathematically defined, and without sharp borders, thus it has been impossible to develop a single system of Concept Graphs covering all features of CGs. Instead of that, since Concept Graphs were introduced for the first time in [22], different forms of concept graphs have been elaborated.

Most important for this paper is to distinguish two different accesses for a development of concept graphs, namely *semantical approaches*, and those based on a separation of syntax and semantics, termed *syntactical approaches*. In order to discuss this distinction accordingly, we first need some definitions. We start with the underlying structures for (both approaches to) concept graphs.

Definition 1 (Relational Graphs with Cuts).
A structure $(V, E, \nu, \top, Cut, area)$ *is called a* relational graph with cuts *if*

- V, E *and* Cut *are pairwise disjoint, finite sets whose elements are called* vertices, edges *and* cuts, *respectively,*
- $\nu : E \to \bigcup_{k\in\mathbb{N}} V^k$ *is a mapping,*[1]
- $\top \notin V \cup E \cup Cut$ *is a single element called the* sheet of assertion, *and*
- $area : Cut \cup \{\top\} \to \mathfrak{P}(V \cup E \cup Cut)$ *is a mapping such that*
  - *a)* $c_1 \neq c_2 \Rightarrow area(c_1) \cap area(c_2) = \emptyset$,
  - *b)* $V \cup E \cup Cut = \bigcup_{d\in Cut\cup\{\top\}} area(d)$,
  - *c)* $c \notin area^n(c)$ *for each* $c \in Cut \cup \{\top\}$ *and* $n \in \mathbb{N}$ *(with* $area^0(c) := \{c\}$ *and* $area^{n+1}(c) := \bigcup\{area(d) \mid d \in area^n(c)\}$*).*

For an edge $e \in E$ *with* $\nu(e) = (v_1, \ldots, v_k)$ *we set* $|e| := k$ *and* $\nu(e)\big|_i := v_i$. *Sometimes, we will write* $e\big|_i$ *instead of* $\nu(e)\big|_i$, *and* $e = (v_1, \ldots, v_k)$ *instead of* $\nu(e) = (v_1, \ldots, v_k)$. *We set* $E^{(0)} := V$ *and* $E^{(k)} := \{e \in E \mid |e| = k\}$ *for* $k > 0$.
As for every $x \in V \cup E \cup Cut$ *we have exactly one context* $c \in Cut \cup \{\top\}$ *with* $x \in area(c)$, *we can write* $c = area^{-1}(x)$ *for every* $x \in area(c)$, *or even more simple and suggestive:* $c = cut(x)$.
If $Cut = \emptyset$, *we will speak simply of* relational graphs, *which can be identified with the more simple structure* $\mathfrak{G} := (V, E, \nu)$.

The contextual structures for both approaches to concept graphs are so-called *power context families (PCFs)*, i.e., families $\vec{\mathbb{K}} := (\mathbb{K}_0, \mathbb{K}_1, \mathbb{K}_2, \ldots, \mathbb{K}_n)$ 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 $k = 1, \ldots, n$. The formal concepts of $\mathbb{K}_k$ with $k = 1, \ldots, n$ are called *relation concepts* because their extents are $k-$ary relations on the object set G_0.
Semantical theories deal with the elaboration of a mathematical structure theory of concept graphs of a given power context family in an *algebraic* manner. They are defined as as follows:

Definition 2 (Semantical Concept Graphs).
A semantical concept graph of a power context family *is a structure* $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$ *such that*

1. (V, E, ν) *is a relational graph without cuts,*
2. $\kappa : V \dot\cup E \to \bigcup_{k\in\mathbb{N}_0} \mathfrak{B}(\mathbb{K}_k)$ *is a mapping with* $\kappa(u) \in \mathfrak{B}(\mathbb{K}_k)$ *for* $u \in E^{(k)}$, *and*
3. $\rho : V \to G_0$ *is a mapping such that* $\rho(v) \in Ext(\kappa(v))$ *for each* $v \in V$, *and* $(\rho(e_1), \ldots, \rho(e_k)) \in Ext(\kappa(e))$ *for each* $e = (v_1, \ldots, v_k) \in E$.

Semantical concept graphs are mathematical structures defined over a given PCF. Then the forms and relations of those concept graphs are investigated. This includes *operations* on those graphs and a thorough study of the properties of the corresponding algebra of concept graphs of a given power context family. This approach was proposed by Wille in [22], and since then, it was further elaborated by himself, Pollandt and Schoolmann (see for instance [12, 13, 16]).
In the *syntactical* approach to concept graphs, the graphs are investigated with methods known from *mathematical logic*. This approach has been investigated by Prediger, Klinger, and Dau (see for instance [14, 15, 7, 8, 2]). In this paper, we will focus on the approach of the author.

[1] We set $\mathbb{N} := \{1, 2, 3, \ldots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

First of all, a separation between syntax and semantics is introduced. In order to do this, first the notion of an alphabet is provided: An *alphabet* is a triple $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ of disjoint sets $\mathcal{G}$, $\mathcal{C}$, $\mathcal{R}$ such that $\mathcal{G}$ is a finite set whose elements are called *object names*, $(\mathcal{C}, \leq_{\mathcal{C}})$ is a finite ordered set with a greatest element $\top$ whose elements are called *concept names*, and $(\mathcal{R}, \leq_{\mathcal{R}})$ is a family of finite ordered sets $(\mathcal{R}_k, \leq_{\mathcal{R}_k})$, $k = 1, \ldots, n$ (for an $n \in \mathbb{N}$) whose elements are called *relation names*. We suppose that we have a special name $\doteq \in \mathcal{R}_2$ which is called *identity*. Finally, let $\leq_{\mathcal{G}}$ be the order on $\mathcal{G} \dot\cup \{*\}$ such that $*$ is the greatest element of $\mathcal{G} \dot\cup \{*\}$, and all elements of $\mathcal{G}$ are incomparable.
Now we can obtain the concept with cuts graphs from relational graphs by labelling the vertices and edges with names from our alphabet. For our purpose, we will consider particularly *(syntactical) concept graphs with cuts (CGwCs)* (see [2]).

Definition 3 (Syntactical Concept Graphs with Cuts).
A structure $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ *is called* syntactical concept graph with cuts over the alphabet $\mathcal{A}$*, if*

- $(V, E, \nu, \top, Cut, area)$ *is a relational graph with cuts,*[2]
- $\kappa : V \cup E \to \mathcal{C} \cup \mathcal{R}$ *is a mapping such that* $\kappa(V) \subseteq \mathcal{C}$*,* $\kappa(E) \subseteq \mathcal{R}$*, and all* $e \in E$ *with* $|e| = k$ *satisfy* $\kappa(e) \in \mathcal{R}_k$*, and*
- $\rho : V \to \mathcal{G} \dot\cup \{*\}$ *is a mapping.*

In contrast to semantical concept graphs, the vertices and edges of a syntactical concept graph are now labelled with *names* for objects and concepts. Thus, we have no direct connection between graphs and PCFs. As usual in mathematical logic, the connection is established by interpreting the names of the alphabet in models. Therefore, in addition to PCFs, we need an interpretation. The resulting structures are the models for the logical approach to concept graphs. The are given as follows:

Definition 4 (Contextual Models).
Let $\mathcal{A} := (\mathcal{G}, \mathcal{C}, \mathcal{R})$ *be an alphabet and* $\vec{\mathbb{K}}$ *be a power context family. Then we call the disjoint union* $\lambda := \lambda_{\mathcal{G}} \dot\cup \lambda_{\mathcal{C}} \dot\cup \lambda_{\mathcal{R}}$ *of the mappings* $\lambda_{\mathcal{G}}: \mathcal{G} \to G_0$*,* $\lambda_{\mathcal{C}}: \mathcal{C} \to \underline{\mathfrak{B}}(\mathbb{K}_0)$ *and* $\lambda_{\mathcal{R}}: \mathcal{R} \to \mathfrak{R}_{\vec{\mathbb{K}}}$ *a* $\vec{\mathbb{K}}$-interpretation *of* $\mathcal{A}$ *if* $\lambda_{\mathcal{C}}$ *and* $\lambda_{\mathcal{R}}$ *are order-preserving, and* $\lambda_{\mathcal{C}}, \lambda_{\mathcal{R}}$ *satisfy* $\lambda_{\mathcal{C}}(\top) = \top$*,* $\lambda_{\mathcal{R}}(\mathcal{R}_k) \subseteq \underline{\mathfrak{B}}(\mathbb{K}_k)$ *for all* $k = 1, \ldots, n$*, and finally,* $(g_1, g_2) \in Ext(\lambda_{\mathcal{R}}(\doteq)) \Leftrightarrow g_1 = g_2$ *for all* $g_1, g_2 \in G_0$*. The pair* $(\vec{\mathbb{K}}, \lambda)$ *is called* contextual model *over* $\mathcal{A}$ *or* contextual structure over $\mathcal{A}$.[3]

The focus of a logical investigation of concept graphs are *derivation rules* instead of operations. Particularly for CGwCs, a sound and complete calculus is provided in [2], to which we will refer in the remaining paper.
These two approaches are two different viewpoints on the same thing, thus they are not competing, but complementary to each other.

2 Background Knowledge

We have already seen that the syntactical concept graphs are based on an alphabet of which the names are ordered. This idea is adopted from the theory of Conceptual Graphs, where this order is usually called *type hierarchy*. This order encodes pre-knowledge on the concepts and relations we consider, thus, the type-hierarchy is how

[2] An additional restriction is needed: $(V, E, \nu, \top, Cut, area)$ must have so-called *dominating nodes*. Due to space limitations, a discussion of this is omitted here. The interested reader is referred to [2].

[3] The name 'contextual structure' is chosen according to the term 'relational structure', which is common in first order logic.

background knowledge is incorporated into the system of Conceptual Graphs. The type-hierarchy yields specific restrictions for models: The interpretation functions λ have to respect the type-hierarchy. As a simple example, let us consider two concept names C_1, C_2 with $C_1 \leq C_2$. If these concept names are interpreted in a contextual structure $(\vec{\mathbb{K}}, \lambda)$, we demand that $\lambda_{\mathcal{C}}(C_1) \leq \lambda_{\mathcal{C}}(C_2)$ is satisfied. That is, $\forall g \in G_0 : g \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_1)) \Rightarrow g \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_2))$. Thus, $C_1 \leq C_2$ can be understood as an implication $C_1 \to C_2$ as well.
In [25, 26], Rudolf Wille – inspired by Brandom's elaboration of 'material implications' in [1] – introduced more complex implications as background knowledge to semantical concept graphs. First of all, he allowed *conjunctions* of concepts in the premises and conclusions of the implications (particularly, this approach is a generalization of the type-hierarchy). Furthermore, he used this approach for implications between objects as well. He considered the following implications:
For two given sets of objects(tuples) $A, B \subseteq G_k$ in a PCF $\vec{\mathbb{K}} := (\mathbb{K}_0, \ldots, \mathbb{K}_n)$ of contexts $\mathbb{K}_k = (G_k, M_k, I_k)$, a *object implication* $A \to B$ is given if we have $A^{I_k} \subseteq B^{I_k}$. For two given sets of concepts $\mathfrak{C}, \mathfrak{D} \subseteq \mathfrak{B}(\mathbb{K}_k)$, a *concept implication* $\mathfrak{C} \to \mathfrak{D}$ is given if we have $\bigwedge \mathfrak{C} \leq \bigwedge \mathfrak{D}$. In the following, it will be investigated how these kinds of implications can be integrated into the logical approach of concept graphs.

	state capital	administrative centre	state	territory	population >6 mill.	population >4.5 mill.	population >3 mill.	population >1.5 mill.	population >1 mill.	population >0.3 mill.	population >0.15 mill.	population >0.08 mill.
Sydney	☒	☒					☒	☒	☒	☒	☒	☒
Melbourne	☒	☒					☒	☒	☒	☒	☒	☒
Brisbane	☒	☒						☒	☒	☒	☒	☒
Perth	☒	☒							☒	☒	☒	☒
Adelaide	☒	☒							☒	☒	☒	☒
Canberra		☒								☒	☒	☒
Hobart	☒	☒									☒	☒
Darwin		☒										☒
ACT				☒						☒	☒	☒
NSW			☒		☒	☒	☒	☒	☒	☒	☒	☒
Vic			☒			☒	☒	☒	☒	☒	☒	☒
Qld			☒				☒	☒	☒	☒	☒	☒
SA			☒						☒	☒	☒	☒
WA			☒					☒	☒	☒	☒	☒
Tas			☒							☒	☒	☒
NT				☒							☒	☒

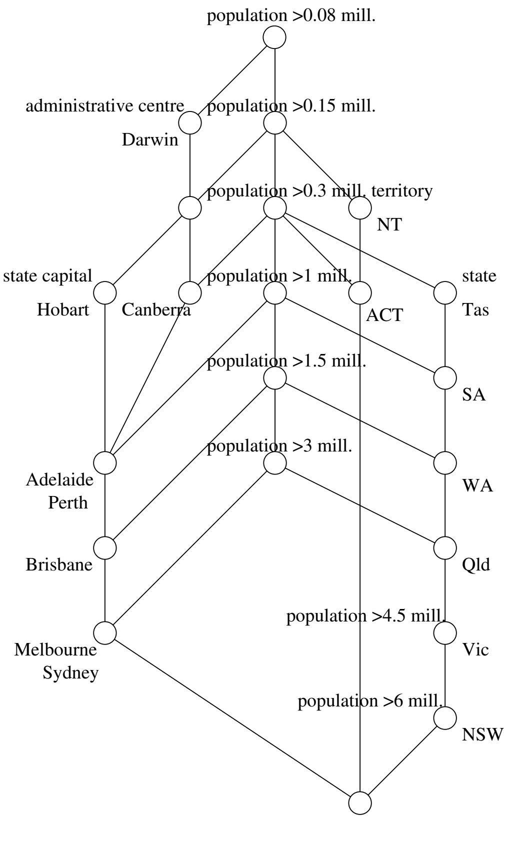


Fig. 1. A PCF of Australian cities and states and its concept lattice

In order to exemplify object- and concept-implications, we adopt the example of [25]. Consider the context of Australian cities and states and its concept lattice, both depicted in Fig. 1. The state names are abbreviated by ACT (Australian Capital

Territory), NSW (New South Wales), Vic (Victoria), Qld (Queensland), SA (South Australia), WA (Western Australia), Tas (Tasmania), and NT (Northern Territory) (see [11]).
The following two implications are valid object implications in our example:

$$Tas \rightarrow SA \qquad \text{and} \qquad SA \wedge ACT \rightarrow Tas \quad .$$

Let C denote the concept generated by the attributes *administrative center* and *population* > 0.3 *mill*. The next two implications are valid concept implications:

$$\mu(\textit{state capital}) \rightarrow \mu(\textit{administrative center}) \qquad \text{and}$$

$$\mu(\textit{state capital}) \wedge C \rightarrow \mu(\textit{population} > 1\ \textit{mill}) \quad .$$

In [25], Wille defines the *informational content* of a semantical concept graph, and he defines a graph $\mathfrak{G}_a$ to be *more specific* than a graph $\mathfrak{G}_b$ (which is denoted by $\mathfrak{G}_a \lesssim \mathfrak{G}_b$), if and only if the informational content of $\mathfrak{G}_b$ is contained in the information content of $\mathfrak{G}_a$. In the following picture, due to the second object implication, the right concept graph is more specific than the left one (where TERRITORY stands for the concept $\mu(\textit{territory})$).

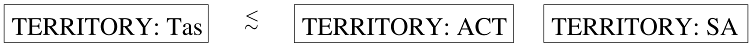


Fig. 2. An example for dependencies of CGs due to an object implication

Analogously, due to the second concept implication, we have the following dependency:

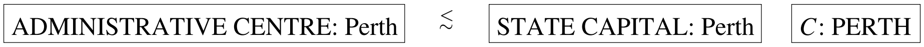


Fig. 3. An example for dependencies of CGs due to an concept implication

It should be briefly discussed why we focus on concept implications instead of attribute implications, as they are known in FCA. As said above, concept graphs are a main part of the program of Contextual Logic, of which the purpose is to develop and elaborate a formal theory of the elementary logic, based on the doctrines of concepts, judgements, and conclusions. Concept graphs are a formalization of judgements. The most elementary judgements are of the form "an object belongs to the extension of a concept", and these judgements correspond to the "atomar" items of concept graphs, namely concept boxes and relation ovals (relation ovals represent elementary judgements where *tuples* of objects belong to the extension of a *relation*-concept). For this reason, in the framework of Contextual Logic, it is quite natural to elaborate concept implications instead of attribute implications. Nonetheless, as each concept is generated by attributes, i.e., as each concept is the meet of attribute concepts, concept implications can be in some sense reduced to attribute implications. We will come back to this point in Sec. 5, where the results of this paper are discussed.
As said above, Wille incorporated object- and concept-implications into his algebraic understanding of concept graphs. The goal of this paper is to incorporate these implications into the logical approach to concept graphs. In the remaining part of this section, some basic ideas for doing this shall be described.

Obviously, an (object or concept) implication $X \to Y$ holds if each implication $X \to y$, $y \in Y$ holds (the above examples are already of this type). Thus, it is sufficient to consider only implications with a simple conclusion. Next, due to the separation of syntax and semantics, we have to consider implication between *names* for objects and concepts. Similar to the type hierarchy from which we obtained specific restrictions of our models, these implications correspond to (further) restrictions of our models as well. The implications can be now restated as follows: If $g_1, \ldots, g_n \in \mathcal{G}$ are object names, then $g_1 \wedge g_2 \wedge \ldots \wedge g_{n-1} \to g_n$ is called a *(syntactical) object implication.* If $C_1, \ldots, C_n \in \mathcal{C}$ are concept names, then $C_1 \wedge C_2 \wedge \ldots \wedge C_{n-1} \to C_n$ is called a *(syntactical) concept implication.* Note that we consider object implications only for objects, not for tuples, and concept implications only for concept names, not for relation names. The other cases can be handled analogously and are omitted for sake of simplicity.

If an object implication $g_1 \wedge g_2 \wedge \ldots \wedge g_{n-1} \to g_n$ is given, we consider only models $\mathcal{M} := (\vec{\mathbb{K}}, \lambda)$ which respect this implication, i.e., models which satisfy $\{\lambda_{\mathcal{G}}(g_1), \ldots, \lambda_{\mathcal{G}}(g_{n-1})\}^{I_0} \subseteq \{\lambda_{\mathcal{G}}(g_n)\}^{I_0}$. In other words:

$$\mathcal{M} \text{ respects } g_1 \wedge \ldots \wedge g_{n-1} \to g_n \quad \text{iff} \quad \gamma(\lambda_{\mathcal{G}}(g_1)) \vee \ldots \vee \gamma(\lambda_{\mathcal{G}}(g_{n-1})) \geq \gamma(\lambda_{\mathcal{G}}(g_n)) \quad (1)$$

Analogously, if a concept implication $C_1 \wedge C_2 \wedge \ldots \wedge C_{n-1} \to C_n$ is given, we have:

$$\mathcal{M} \text{ respects } C_1 \wedge \ldots \wedge C_{n-1} \to C_n \quad \text{iff} \quad \lambda_{\mathcal{C}}(C_1) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{n-1}) \leq \lambda_{\mathcal{C}}(C_n) \quad (2)$$

If we understand a subsumption relation $C_1 \leq C_2$ in a type hierarchy as an implication $C_1 \to C_2$, Eqn. (2) extends the conditions for models which respect a type-hierarchy.

We have already seen that background knowledge yields specific restrictions for the models. These restrictions have to be reflected by derivation rules, i.e., we have to provide rules for the calculus which reflect exactly the background knowledge. For example, the calculus should allow us to derive the left graph of Fig. 3 from the right one. Analogously, the left graph of Fig. 2 should be derivable from the right one.

3 General Logical Background

As already mentioned, the graphs in [2] are based on an alphabet of which the names are ordered. Moreover, in [2], a sound and complete calculus for CGwCs is provided. Particularly, this calculus has two rules (named 'specialization' and 'generalization') which reflect the order of the names.

Object- and concept implications are more complex than an order on the set of object- or concept names, that is, an order on the set of names can be reflected by object- or concept implications, but not vice versa. Our goal is to incorporate these kinds of implications into the theory of CGwCs. To be more specific: Instead of an ordered alphabet of which the order is reflected by rules in the calculus, we will now have an unordered alphabet, but a set of implications which has to be reflected by the calculus. Of course, this calculus has not to be developed 'from the scratch': We already have a sound and complete calculus for CGwCs, but without background-knowledge in the form of object- or concept implications. These implications will be incorporated by adding further rules to the calculus. It has to be checked whether these additional rules correspond to the implications of our background knowledge. In this section, a method for doing this will be described.

Our starting point are CGwCs which are based on a (non-ordered) alphabet. These concept graphs are evaluated in contextual structures via the relation $\models$. Let us denote the class of models for this kind of concept graphs by $\mathfrak{M}_1$. In [2], a calculus

for these graphs is provided. Let us denote the set of rules by $\vdash_1$. (This use of the symbol '$\vdash$' is a little bit sloppy: Usually, the symbol denotes the syntactical entailment relation between formulas of a given logic, which is *derived* from a set of rules. We will use '$\vdash$' in this sense as well, but the set of rules shall also be denoted with '$\vdash$'. It will be clear from the context which use of '$\vdash$' is intended.)
The basic idea is that further background knowledge corresponds to a restriction of $\mathfrak{M}_1$ (see the discussion above). So we get class of models $\mathfrak{M}_2 \subseteq \mathfrak{M}_1$. On the other hand, we are looking for an extension of the set of rules which captures exactly the background knowledge, i.e., we are looking for a calculus $\vdash_2 \supseteq \vdash_1$.
Similar to the distinction between $\vdash_1$ and $\vdash_2$, we will distinguish between $\models_1$ and $\models_2$ as follows: For two graphs $\mathfrak{G}_a, \mathfrak{G}_b$, we will write $\mathfrak{G}_a \models_i \mathfrak{G}_b$ if and only if $\mathcal{M} \models \mathfrak{G}_b$ for all models $\mathcal{M} \in \mathfrak{M}_i$ with $\mathcal{M} \models \mathfrak{G}_a$.
Analogously, we define $\mathfrak{H} \vdash_i \mathfrak{G}$ and $\mathfrak{H} \models_i \mathfrak{G}$ for a graph $\mathfrak{G}$ and a set of graphs $\mathfrak{H}$.
The soundness and completeness of $\vdash_1$ can be now stated as follows:

$$\text{Let } \mathfrak{G}_1, \mathfrak{G}_2 \text{ two CGwCs. We have: } \mathfrak{G}_1 \vdash_1 \mathfrak{G}_2 \quad \Longleftrightarrow \quad \mathfrak{G}_1 \models_1 \mathfrak{G}_2 \tag{3}$$

We assume that both $\vdash_1, \vdash_2$ are Peirce-style-calculi, that is:

1. Each rule can be applied either in arbitrary positive contexts, or in arbitrary negative contexts, or in each context of a graph, and
2. For each rule which can be applied in arbitrary positive contexts, we have a rule in the opposite direction which can be applied in arbitrary negative contexts. Vice versa: For each rule which can be applied in arbitrary negative contexts, we have a rule in the opposite direction which can be applied in arbitrary positive contexts. The rules which can be applied in each contexts can be applied in both directions.

The calculus $\vdash_1$, and hence $\vdash_2$ as well, encompasses the 5 basic-rules of Peirce. This yields a first simple lemma:

Lemma 1 (Syntactical and Semantical Deduction Theorem).
Let $\mathfrak{G}_a$, $\mathfrak{G}_b$ be two CGwCs. Then we have

$$\mathfrak{G}_a \vdash_i \mathfrak{G}_b \iff \vdash_i \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}} \tag{4}$$

$$\mathfrak{G}_a \models_i \mathfrak{G}_b \iff \models_i \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}} \tag{5}$$

Proof: We show both directions separately and start with '$\Longrightarrow$':

$$\overset{\text{dc.}}{\vdash} \boxed{\quad \boxed{\quad}} \overset{\text{ins.}}{\vdash} \boxed{\mathfrak{G}_a \boxed{\quad}} \overset{\text{it.}}{\vdash} \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_a}} \overset{\text{Lm. 6.4. in [2]}}{\vdash} \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}}$$

The direction '$\Longleftarrow$' is done as follows:

$$\mathfrak{G}_a \overset{\text{Lm. 6.4. in [2]}}{\vdash} \mathfrak{G}_a \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}} \overset{\text{deit.}}{\vdash} \mathfrak{G}_a \boxed{\quad \boxed{\mathfrak{G}_b}} \overset{\text{dc}}{\vdash} \mathfrak{G}_a \quad \mathfrak{G}_b \overset{\text{era.}}{\vdash} \mathfrak{G}_b$$

□

By definition of $\vdash$, we have: If $\mathfrak{H} := \{\mathfrak{G}_i \mid i \in I\}$ is a (possibly empty) set of CGwCs, then a graph $\mathfrak{G}$ can be derived from $\mathfrak{H}$ if and only if there is a finite subset $\{\mathfrak{G}_1, \ldots, \mathfrak{G}_n\} \subseteq \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \vdash \mathfrak{G}$ (where $\mathfrak{G}_1 \ldots \mathfrak{G}_n$ is the juxtaposition of $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$).
It is easy to check that $\mathfrak{M}_1$ satisfies the compactness-theorem. Thus, for the semantical entailment relation, we have a similar property, i.e. we have: If $\mathfrak{H} := \{\mathfrak{G}_i \mid i \in I\}$

is a (possibly empty) set of nonexistential concept graphs, then $\mathfrak{H} \models \mathfrak{G}$ if and only if there is a finite subset $\{\mathfrak{G}_1, \ldots, \mathfrak{G}_n\} \subseteq \mathfrak{H}$ with $\mathfrak{G}_1 \ldots \mathfrak{G}_n \models \mathfrak{G}$. From this we immediately conclude for a CGwC $\mathfrak{G}$ and set $\mathfrak{H}$ of CGwCs:

$$\mathfrak{H} \vdash_1 \mathfrak{G} \quad \Longleftrightarrow \mathfrak{H} \models_1 \mathfrak{G} \tag{6}$$

Now the main idea for the extension of $\vdash_1$ to $\vdash_2$ is the following: The models in $\mathfrak{M}_2$ have to be described properly by the additional rules in $\vdash_2 \setminus \vdash_1$. That is, on the one hand, the rules in $\vdash_2$ have to be sound, i.e. for two CGwCs $\mathfrak{G}_1, \mathfrak{G}_2$ we have

$$\mathfrak{G}_1 \vdash_2 \mathfrak{G}_2 \quad \Longrightarrow \mathfrak{G}_1 \models_1 \mathfrak{G}_2 \tag{7}$$

On the other hand, let us assume that for each $\mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2$, there exists a graph $\mathfrak{G}_M$ with

$$\vdash_2 \mathfrak{G}_M \text{ and } \mathcal{M} \not\models \mathfrak{G}_M \tag{8}$$

If the last two assumptions (7) and (8) hold, we obtain that $\vdash_2$ is an adequate calculus, as the following theorem shows.

Theorem 1 (Completeness of $\vdash_2$).
$\vdash_2$ *is complete, i.e.* $\mathfrak{G}_a \models_2 \mathfrak{G}_b \Longrightarrow \mathfrak{G}_a \vdash_2 \mathfrak{G}_b$.

Proof: Let $\mathfrak{H} := \{\mathfrak{G}_M \,|\, \mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2\}$. From (7) we conclude: $\models_2 \mathfrak{G}_M$ for all $\mathfrak{G}_M \in \mathfrak{H}$. Now (8) yields:

$$\mathfrak{M}_2 = \{\mathcal{M} \in \mathfrak{M}_1 \,|\, \mathcal{M} \models \mathfrak{G} \text{ for all } \mathfrak{G} \in \mathfrak{H}\} \tag{9}$$

Thus, we get:

$$\begin{aligned}
\mathfrak{G}_a \models_2 \mathfrak{G}_b &\overset{\text{Def}}{\Longleftrightarrow} \text{f.a. } \mathcal{M} \in \mathfrak{M}_2 : \text{ if } M \models \mathfrak{G}_a, \text{ then } \mathcal{M} \models \mathfrak{G}_b \\
&\overset{(9)}{\Longleftrightarrow} \text{f.a. } \mathcal{M} \in \mathfrak{M}_1 : \text{ if } M \models \mathfrak{G} \text{ for all } \mathfrak{G} \in \mathfrak{H} \text{ and } M \models \mathfrak{G}_a, \text{ then } M \models \mathfrak{G}_b \\
&\Longleftrightarrow \mathfrak{H} \mathbin{\dot\cup} \{\mathfrak{G}_a\} \models_1 \mathfrak{G}_b \\
&\overset{(6)}{\Longleftrightarrow} \mathfrak{H} \mathbin{\dot\cup} \{\mathfrak{G}_a\} \vdash_1 \mathfrak{G}_b \\
&\Longleftrightarrow \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \mathfrak{G}_1 \quad \mathfrak{G}_2 \ldots \quad \mathfrak{G}_n \quad \mathfrak{G}_a \vdash_1 \mathfrak{G}_b \\
&\overset{(4)}{\Longleftrightarrow} \text{ there are } \mathfrak{G}_1, \ldots, \mathfrak{G}_n \in \mathfrak{H} \text{ with } \vdash_i \boxed{\mathfrak{G}_1 \, \mathfrak{G}_2 \ldots \mathfrak{G}_n \, \mathfrak{G}_a \boxed{\mathfrak{G}_b}}
\end{aligned}$$

From this result, $\vdash_2 \supseteq \vdash_1$ and (8) we obtain:

$$\vdash_2 \mathfrak{G}_1 \ \ldots \ \mathfrak{G}_n \boxed{\mathfrak{G}_1 \ \ldots \ \mathfrak{G}_n \, \mathfrak{G}_a \boxed{\mathfrak{G}_b}} \overset{\text{deit.}}{\vdash_2} \mathfrak{G}_1 \ \ldots \ \mathfrak{G}_n \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}} \overset{\text{era.}}{\vdash_2} \boxed{\mathfrak{G}_a \boxed{\mathfrak{G}_b}}$$

Now (4) yields $\mathfrak{G}_a \vdash_2 \mathfrak{G}_b$. □

4 Incorporating Background Knowledge into the Calculus

As the preceding section shows, we have to find additional and sound rules which describe properly the restriction of $\mathfrak{M}_1$ to $\mathfrak{M}_2$. This is done separately for object- and concept-implications. Concept implications are easier to handle, so we start with them.

4.1 Incorporating Concept Implications

We have already seen that concept implications are a generalization of the order on the concept names in a type-hierarchy. A subsumption relation $C_1 \leq C_2$ is reflected in reflected in $\vdash_1$ by the rules 'generalization' and 'specialization', which –particularly for $C_1 \leq C_2$– allow to replace an evenly enclosed vertex $\boxed{C_1 : g}$ (with $g \in \mathcal{G} \dot\cup \{*\}$) by $\boxed{C_2 : g}$, and, vice versa, to replace an oddly enclosed vertex $\boxed{C_2 : g}$ by $\boxed{C_1 : g}$. These rules will be canonically extended.

Let a concept implication $C_1 \wedge \ldots \wedge C_{n-1} \to C_n$ for concept names $C_1, \ldots, C_n \in \mathcal{C}$ be given. As discussed in Sec. 2, this implication yields the following restriction for models: $\mathcal{M} \in \mathfrak{M}_2 \iff \mathcal{M} \in \mathfrak{M}_1$ and $\lambda_{\mathcal{C}}(C_1) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{n-1}) \leq \lambda_{\mathcal{C}}(C_n)$. In other words: For all $g \in G_0$ we have

$$g \in \lambda_{\mathcal{C}}(C_1) \wedge \ldots g \in \lambda_{\mathcal{C}}(C_{n-1}) \Longrightarrow g \in \lambda_{\mathcal{C}}(C_n) \tag{10}$$

Before we provide a rule which encompasses this implication, we need a simple definition.

Definition 5 ($\boldsymbol{\theta_{\mathfrak{G}}}$).
Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be a concept graph over $\mathcal{A}$. Let $\theta_{\mathfrak{G}}$ be the smallest equivalence relation on V such that if $e \in E$ is an edge with $\nu(e) = (v_1, v_2)$ and $\kappa(e) \leq \doteq$, then $v_1 \theta_{\mathfrak{G}} v_2$.

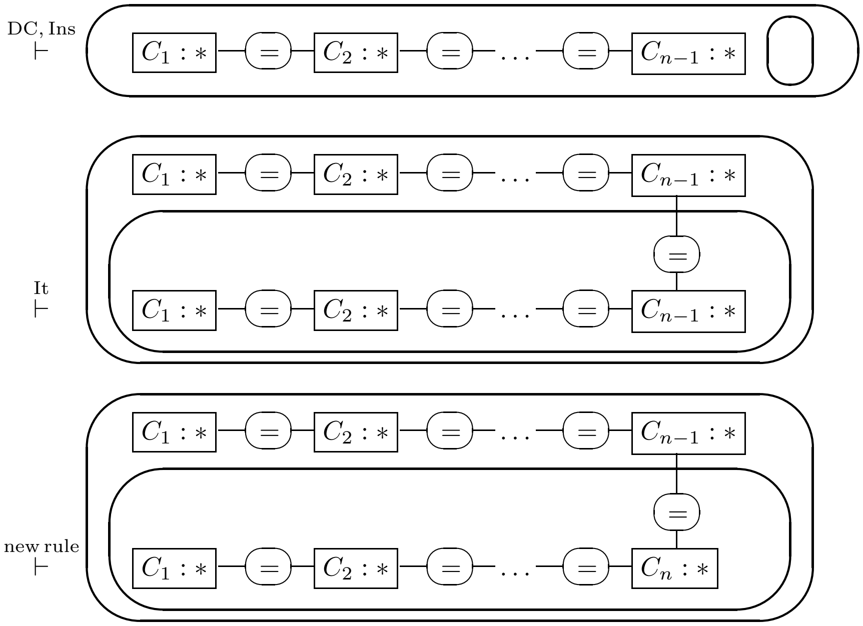
Now the new rules for the calculus are:

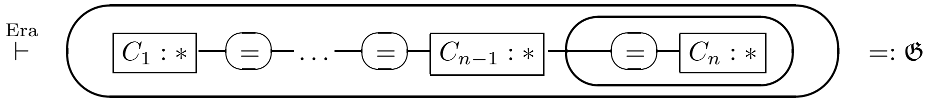
1. **generalization of concepts**
   Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be a CGwCs and let $v_1, \ldots, v_{n-1} \in V$ be vertices with $v_1 \theta_{\mathfrak{G}} v_2 \theta_{\mathfrak{G}} \ldots \theta_{\mathfrak{G}} v_{n-1}$ such that $c := cut(v_1) = \ldots = cut(v_{n-1})$ is an even context, and $\kappa(v_i) = C_i$ for all $1 \leq i \leq n-1$. Then for each v_i, $\kappa(v_i) = C_i$ may be replaced by $\kappa(v_i) = C_n$.
2. **specialization of concepts**
   The preceding rule may be reversed in negative contexts.

We have to show that these rules satisfy Eqn. (7) and Eqn. (8).

The soundness of the rules can be proven analogously to the soundness of the rules 'generalization' and 'specialization' in $\vdash_1$ (see [2]), hence Eqn. (7) holds.

Now we derive with the new rule a graph $\mathfrak{G}$ as follows:





Let $\mathcal{M} \in \mathfrak{M}_1 \backslash \mathfrak{M}_2$ be an arbitrary contextual model. Eqn. (10) immediately yields that $\mathcal{M} \not\models \mathfrak{G}$. We conclude that Eqn. (8) is satisfied, thus Thm. 1 yields that $\vdash_2$ is complete.

4.2 Background Knowledge for Objects

Incorporating concept implications into $\vdash_2$ was straight forward. First of all, concept implications can be considered to be a generalization of the idea of a type hierarchy, i.e., in our alphabet, we have an order on the concept names. But we do not have a corresponding order on the object names in a type hierarchy, which could analogously be generalized to object implications. Moreover, the proof for the completeness in the last subsection heavily relies on the fact that we can quantify over objects (to be more precisely: In the system of CGwCs, we can express universal quantification and material implications). Thus, the structure of the proof in the last subsection cannot be adopted for object implications, as we have no possibility to quantify over concepts.

The main idea now is as follows: If we had 'enough' concept names to describe *all* concepts of a contextual structure, we do not need an universal quantification. Then we could hopefully adopt the idea of the last section.

First of all, we have to clarify the meaning of 'enough'. Usually, the concept names do not encompass all concepts in a contextual structure. But it is not necessary that each concept (A, B) of a given contextual model is generated by a concept name (i.e., there exists a concept name C with $\lambda_{\mathcal{C}}(C) = (A, B)$). The following further restriction for the models in $\mathfrak{M}_2$ will turn out to be sufficient:

> **General Restriction for Models $\boldsymbol{\mathcal{M} \in \mathfrak{M}_2}$**: The concepts of $\mathcal{M}$ which are generated by concept names are $\bigwedge$-dense in the set of all concepts. $(*)$

Now let an object implication $g_1 \wedge \ldots \wedge g_{n-1} \to g_n$ for object names $g_1, \ldots, g_n \in \mathcal{G}$ be given. This implication yields the following restriction for models:

$$\mathcal{M} \in \mathfrak{M}_2 \;:\Longleftrightarrow\; \mathcal{M} \in \mathfrak{M}_1 \text{ , } \mathcal{M} \text{ satisfies } (*) \text{ and } \lambda_{\mathcal{G}}(g_1) \vee \ldots \vee \lambda_{\mathcal{G}}(g_{n-1}) \geq \lambda_{\mathcal{G}}(g_n)$$

The rules capturing this restriction are defined analogously to the rules for concept implications. They are as follows:

1. **generalization of objects**
   Let $\mathfrak{G} := (V, E, \nu, \top, Cut, area, \kappa, \rho)$ be a CGwC and let $v_1, \ldots, v_n \in V$ be vertices such that $c := cut(v_1) = \ldots cut(v_{n-1}) = cut(v_n)$ is an even context, we have $\rho(v_i) = g_i$ for all $1 \leq i \leq n-1$, and there is a (fixed) $C \in \mathcal{C}$ with $\kappa(v_i) = C$ for all $1 \leq i \leq n-1$. Then, for each v_i, $\rho(v_i) = g_i$ may be replaced by $\rho(v_i) = g_n$.
2. **specialization of objects**
   The preceding rule may be reversed in negative contexts.

Again, the soundness of these rules can be proven analogously to the soundness of the rules 'generalization' and 'specialization' in $\vdash_1$, so Eqn. (7) holds.

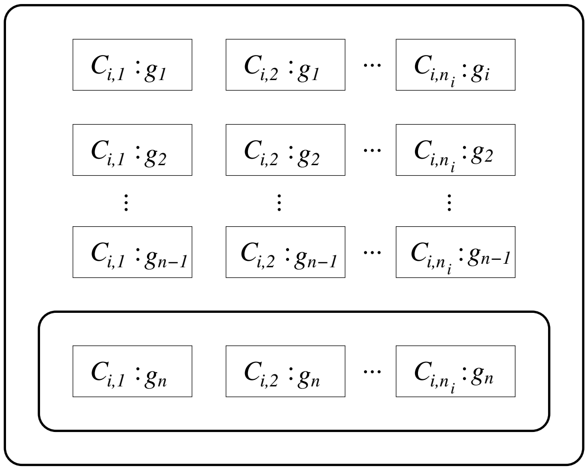
The proof of Eqn. (8) for these rules is more complex than for object implications. In contrast to object implications, we will construct for each model $\mathcal{M} \in \mathfrak{M}_2$ a graph $\mathfrak{G}_{\mathcal{M}}$ which satisfies Eqn. (8). So let $\mathcal{M} \in \mathfrak{M}_2$ be a fixed model.

Let $M := \{m_1, \ldots, m_k\}$ be an enumeration of the attributes of $\mathcal{M}$. Due to $(*)$, each attribute concept $\mu(m_i)$, $1 \leq i \leq k$, of $\mathcal{M}$ is the meet of some concepts generated by concept names. We fix these concept names as follows: For each $1 \leq i \leq k$, let $C_{i,1}, \ldots, C_{i,n_i}$ be n_i concepts names with $\mu(m_i) = \lambda_{\mathcal{C}}(C_{i,1}) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{i,n_i})$. In order to construct a graph $\mathfrak{G}_M$ for Eqn. (8), we transform the model restriction $\gamma(\lambda_{\mathcal{G}}(g_1)) \vee \ldots \vee \gamma(\lambda_{\mathcal{G}}(g_{n-1})) \geq \gamma(\lambda_{\mathcal{G}}(g_n))$ as follows:

$$\begin{aligned}
&\gamma(\lambda_{\mathcal{G}}(g_1)) \vee \ldots \vee \gamma(\lambda_{\mathcal{G}}(g_{n-1})) \geq \gamma(\lambda_{\mathcal{G}}(g_n)) \\
&\iff \lambda_{\mathcal{G}}(g_1)^I \cap \ldots \cap \lambda_{\mathcal{G}}(g_{n-1})^I \subseteq \lambda_{\mathcal{G}}(g_n)^I \\
&\iff \forall m \in \mathcal{M} : \lambda_{\mathcal{G}}(g_1) I m \wedge \ldots \wedge \lambda_{\mathcal{G}}(g_{n-1}) I m \Rightarrow \lambda_{\mathcal{G}}(g_n) I m \\
&\iff \forall m \in \mathcal{M} : \lambda_{\mathcal{G}}(g_1) \in \mathrm{Ext}(\mu(m)) \wedge \ldots \wedge \lambda_{\mathcal{G}}(g_{n-1}) \in \mathrm{Ext}(\mu(m)) \Rightarrow \\
&\qquad\qquad \lambda_{\mathcal{G}}(g_n) \in \mathrm{Ext}(\mu(m)) \\
&\iff \forall 1 \leq i \leq k : \lambda_{\mathcal{G}}(g_1) \in \mathrm{Ext}[\lambda_{\mathcal{C}}(C_{i,1}) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{i,n_i})] \wedge \ldots \wedge \\
&\qquad\qquad \lambda_{\mathcal{G}}(g_{n-1}) \in \mathrm{Ext}[\lambda_{\mathcal{C}}(C_{i,1}) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{i,n_i})] \Rightarrow \\
&\qquad\qquad \lambda_{\mathcal{G}}(g_n) \in \mathrm{Ext}[\lambda_{\mathcal{C}}(C_{i,1}) \wedge \ldots \wedge \lambda_{\mathcal{C}}(C_{i,n_i})] \\
&\iff \forall 1 \leq i \leq k : \lambda_{\mathcal{G}}(g_1) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,1})) \wedge \ldots \wedge \lambda_{\mathcal{G}}(g_1) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,n_i})) \wedge \ldots \wedge \\
&\qquad\qquad \lambda_{\mathcal{G}}(g_{n-1}) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,1})) \wedge \ldots \wedge \lambda_{\mathcal{G}}(g_{n-1}) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,n_i})) \Rightarrow \\
&\qquad\qquad \lambda_{\mathcal{G}}(g_n) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,1})) \wedge \ldots \wedge \lambda_{\mathcal{G}}(g_n) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C_{i,n_i})) \qquad (11)
\end{aligned}$$

Recall that $\lambda_{\mathcal{G}}(g) \in \mathrm{Ext}(\lambda_{\mathcal{C}}(C))$ is, as a concept graph, expressed by the concept box $\boxed{C : g}$. So we see that the last equation (which depends on i) can be represented by a concept graph as follows:

$\mathfrak{G}_{\mathcal{M},i} :=$



It is easy to see that each graph $\mathfrak{G}_{\mathcal{M},i}$ can be derived from $\vdash_2$. Let $\mathfrak{G}_{\mathcal{M}}$ be the juxtaposition of all $\mathfrak{G}_i$, $1 \leq i \leq k$, (which can be derived as well).
Similar to concept implications, Eqn. (11) yields that $\mathcal{M} \not\models \mathfrak{G}_{\mathcal{M}}$, so Eqn. (8) is satisfied. Again Thm. 1 yields that $\vdash_2$ is complete.

5 Discussion of the Results and Further Research

It has been the goal of the paper to show that object- and concept implications, which Wille introduced to semantical concept graphs, can adequately be handled in syntactical concept graphs as well. At a first glance, due to the proven results, the goal of the paper is fulfilled. A closer observation shows valuable insights into the different structure of semantical and syntactical concept graphs.

We have started with CGwCs, without any background knowledge. Then we argued that the incorporation of background knowledge corresponds to a restrictions of our models. The basic idea of this paper is that these restrictions can already be "described" with our starting CGwCs, which has been captured by Eqn. (8). Satisfying Eqn. (8) was possible because the expressive power of CGwCs is strong enough. In contrast to the syntactical approach, for the semantical approach to concept graphs, background knowledge was introduced to concept graphs in which we have no possibility to express negation. Due to the weak expressive power of those graphs, the restrictions of our models cannot be described with syntactical concept graphs *without* cuts. Thus, the idea of this paper cannot be adopted to incorporate background knowledge to syntactical concept graphs without cuts. So the question arises how background knowledge can be handled in these graphs.

In [2, 3], it is shown that the calculus for concept graph with cuts is still complete if it is restricted to the class of concept graphs *without* cuts. The proof relies on completely different methods than the proofs in this paper: It is based on so-called standard-models and standard-graphs. It is likely that the methods of [2, 3] can be adopted to incorporate background knowledge to syntactical concept graphs without cuts, but of course, this has to be worked out in a precise manner.

The next remarkable fact is that the handling of object implications had been significant more complex than the handling of concept implications: We needed the further restriction $(*)$ on page 10. For concept implications, we did not need a similar restriction. The main reason for this is the fact that -similar to first order logic- we can express with CGwCs an universal quantification for objects (in other words: we have statements which range over the set of objects). But we cannot express an universal quantification over concepts or relations. It should be stressed that *with* an universal quantification for concepts, we would have the full expressive power of second order logic. But then, due to the theorems of Lindström (see [10, 5]), it is impossible to obtain a sound and complete calculus at all! Thus, an extension in the language which allows us to express a quantification over concepts or relations is is even not desirable.

In Formal Concept Analysis, thus in the semantical approach for concept graphs, we have no restriction which corresponds to $(*)$. We have already argued that concept implications can be reduced to attribute implications. Moreover, we have the well-known duality for objects and attributes. Thus, due to this duality principle, it is obvious that -for the semantical approach- concept implications and attribute implications can be treated similar.

The different treatment of object- and concept implication in the syntactical approach would not arise if we used use the "classical" relational models of first order logic (instead of the contextual models) as semantics for concept graphs. Remember that a relational model is a pair (U, I), consisting of an universe of discourse U, and an interpretation function I which maps object names to objects and relation names to (extensional) relations. Particularly, we do not have more relations in this model than relation names in the underlying alphabet. For contextual models, the situation is different: We may have many more concepts or relations than we can describe with our language of syntactical concept graphs over a fixed alphabet. For this reason we had to introduce restriction $(*)$. If $(*)$ is satisfied, then, roughly speaking, each concept or relation of a contextual model can be described within our language of CGwCs, even if the concept or relation has no name.

In practice, the restriction $(*)$ should have no considerable effects. This can easily be seen if we examine how Toscana-Systems are set up. This process usually starts with fixing a set of objects and multi-valued attributes, from which –usually with the process of scaling– a formal context and its concepts is derived. The crucial point for our consideration is that the knowledge experts determine the objects, attributes and the incidence relations of the context, then, afterwards, the concept lattice is derived and examined. Particularly, a knowledge experts explicates only attributes and attribute concepts, but no further concepts. Of course, these explicated attribute concepts are $\bigwedge$-dense in the set of all concepts of the formal context, i.e., ($*$) will obviously be fulfilled. In this sense, for incorporating the results of this paper in the development of Toscana-systems, we probably do not have to take care of ($*$). Nonetheless, the impact of ($*$) to models has to be further investigated.

References

[1] R. B. Brandom: *Making it explicit. Reasoning, representing, and discursive commitment.* Harvard University Press, Cambridge 1994. Suhrkamp, 2001.

[2] F. Dau: *The Logic System of Concept Graphs with Negation (And Its Relationship to Predicate Logic).* Lecture Notes in Artificial Intelligence, Vol. 2892, ISBN 3-540-20607-8. Springer, Berlin–Heidelberg–New York, 2003.

[3] F. Dau: *Concept Graphs without Negations: Standardmodels and Standardgraphs.* In: Moor, A., Lex, W., Ganter, B. (Eds.): Conceptual Structures for Knowledge Creation and Communication. LNAI 2746, ISBN 3-540-40576-3, Springer-Verlag, Heidelberg-Berlin, 2003, 243 pp. (This paper is a part of [2] as well.)

[4] F. Dau, J. Klinger: *From Formal Concept Analysis to Contextual Logic.* To appear in the proceedings of the First International Conference on Formal Concept Analysis, 2003.

[5] H.-D. Ebbinghaus, J. Flum, W. Thomas: *Einführung in die mathematische Logik.* Wissenschaftliche Buchgesellschaft Darmstadt, 1978.

[6] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin–Heidelberg–New York 1999.

[7] J. Klinger: *Semiconcept Graphs: Syntax and Semantics.* Diploma thesis, TU Darmstadt, 2001.

[8] J. Klinger: *Simple Semiconcept Graphs: A Boolean Logic Approach.* In: H.S. Delugach, G. Stumme, (Eds.): Conceptual Structures: Broadening the Base. LNAI 2120, Springer-Verlag, Berlin–Heidelberg, 2001.

[9] J. Klinger: *Semiconcept Graphs with Variables.* In G. Angelova, D. Corbett, U: Priss (Eds.): Conceptual Structures: Integration and Interfaces. LNAI 2393, Springer Verlag, Berlin–Heidelberg, 2002.

[10] P. Lindström: *On extensions of elementary logic.* Theoria 35, 1969.

[11] M. Nicholson: *The pocket Aussie fact book.* Penguin Books Australia, Ringwood 1999.

[12] S. Pollandt: *Relational Constructions on Semiconcept Graphs.* In: B. Ganter, G. Mineau (Eds.): Conceptual Structures: Extracting and Representing Semantics. Contributions to ICCS 2001, Stanford.

[13] S. Pollandt: *Relation Graphs: A Structure for Representing Relations in Contextual Logic of Relations.* In G. Angelova, D. Corbett, U: Priss (Eds.): Conceptual Structures: Integration and Interfaces. LNAI 2393, Springer Verlag, Berlin–Heidelberg 2002.

[14] S. Prediger: *Kontextuelle Urteilslogik mit Begriffsgraphen. Ein Beitrag zur Restrukturierung der mathematischen Logik.* Shaker Verlag 1998.

[15] S. Prediger: *Simple Concept Graphs: A Logic Approach.* In: M. -L. Mugnier, M. Chein (Eds.): Conceptual Structures: Theory, Tools and Applications. LNAI 1453, Springer Verlag, Berlin–New York 1998, 225–239.

[16] L. Schoolmann, R. Wille: *Concept Graphs with Subdivision: A Semantic Approach.* In: A. d. Moor, W. Lex, B. Ganter (Eds.): Conceptual Structures for Knowledge Creation and Communication, LNAI, Springer Verlag, Berlin–Heidelberg 2003.

[17] J. F. Sowa: *Conceptual Structures: Information Processing in Mind and Machine.* Addison Wesley Publishing Company Reading, 1984.

[18] J. F. Sowa: *Conceptual Graphs Summary.* In: T. E. Nagle, J. A. Nagle, L. L. Gerholz, P. W. Eklund (Eds.): Conceptual Structures: current research and practice, Ellis Horwood, 1992, 3–51.

[19] J. F. Sowa: *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[20] R. Wille: *Plädoyer für eine Philosophische Grundlegung der Begrifflichen Wissensverarbeitung.* In: R. Wille, M. Zickwolff (Eds.): Begriffliche Wissensverarbeitung: Grundfragen und Aufgaben. B.I.–Wissenschaftsverlag, Mannheim, 1994, 11–25.

[21] R. Wille: *Restructuring Mathematical Logic: An Approach Based on Peirce's Pragmatism.* In: A. Ursini, P. Agliano (Eds.): Logic and Algebra. Marcel Dekker, New York 1996, 267–281.

[22] R. Wille: *Conceptual Graphs and Formal Concept Analysis.* In: D. Lukose et al. (Eds.): Conceptual Structures: Fulfilling Peirce's Dream. LNAI 1257, Springer Verlag, Berlin–New York 1997, 290–303. y

[23] R. Wille: *Contextual Logic Summary.* In: G. Stumme (Ed.): Working with Conceptual Structures. Contributions to ICCS 2000. Shaker, Aachen 2000, 265–276.

[24] R. Wille: *Lecture Notes on Contextual Logic of Relations.* FB4-Preprint, TU-Darmstadt, 2000.

[25] R. Wille: *Existential Concept Graphs of Power Context Families.* In G. Angelova, D. Corbett, U: Priss (Eds.): Conceptual Structures: Integration and Interfaces. LNAI 2393, Springer Verlag, Berlin–Heidelberg 2002.

[26] R. Wille: *Conceptual Contents as Information – Basics for Contextual Judgement Logic.* In: A. d. Moor, W. Lex, B. Ganter (Eds.): Conceptual Structures for Knowledge Creation and Communication, LNAI, Springer Verlag, Berlin–Heidelberg 2003.