Simple Concept Graphs: A Logic Approach

Susanne Prediger

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

Published in: M.-L. Mugnier / M. Chein (eds.): Conceptual Structures: Theory, Tools and Application, Lectures Notes in Artificial Intelligence 1453, Springer Verlag, Berlin–Heidelberg 1998, 225-239

Abstract. Conceptual Graphs and Formal Concept Analysis are combined by developing a logical theory for concept graphs of relational contexts. Therefore, concept graphs are introduced as syntactical constructs, and their semantics is defined based on relational contexts. For this contextual logic, a sound and complete system of inference rules is presented and a standard graph is introduced that entails all concept graphs being valid in a given relational context. A possible use for conceptual knowledge representation and processing is suggested.

Contents

1. Introduction
2. Syntax for the Language of Concept Graphs
3. Semantic for Concept Graphs
4. Reasoning with Concept Graphs
5. The Standard Graph of a Relational Context
6. Contextual Logic for Knowledge Representation
7. Appendix: Formal Proofs

1 Introduction

The first approach combining the theory of Conceptual Graphs and Formal Concept Analysis was described by R.Wille in [Wi97]. For connecting the conceptual structures in both theories, the concept types appearing in conceptual graphs were considered to be formal concepts of formal contexts, the constituents in Formal Concept Analysis. To facilitate this connection, concept graphs, appropriate mathematizations of conceptual graphs, were introduced. The theory of concept graphs of formal contexts was developed as a mathematical structure theory where concept graphs of formal contexts are realizations of abstract concept graphs.

In this paper, a logic approach is presented by developing a logical theory for concept graphs. Therefore, concept graphs are defined as syntactical constructs over an alphabet of objects names, concept names and relation names (Section 2). Then, a contextual semantics is specified. We interpret the syntactical names by objects, concepts and relations of a relational context (Section 3). In this way, we can profit from all notions for concepts that have been developed in Formal Concept Analysis.

The introduced contextual logic is carried on by the study of inferences (Section 4). Based on a model-theoretic notion for the entailment of concept graphs, a sound and complete set of inference rules is established and compared to the notion of projections between concept graphs. With the standard model, we can present another interesting tool for reasoning with concept graphs. In Section 5, we introduce a standard graph for a given relational context. It gives a basis of all concept graphs being valid in the relational context. In the last section, we suggest how this approach can be used for knowledge representation and processing.

2 Syntax for the Language of Concept Graphs

We want to introduce concept graphs as syntactical constructs. Therefore, we need an alphabet of concept names, relation names and objects names. As the theory of conceptual graphs provides an order-sorted logic, we start with ordered sets of names that are not necessarily lattices. These orders are determined by the taxonomies of the domains in view, they formalize ontological background knowledge.

Definition 1. *An* alphabet of concept graphs *is a triple* $(\mathcal{C}, \mathcal{G}, \mathcal{R})$*, where* $(\mathcal{C}, \leq_{\mathcal{C}})$ *is a finite ordered set whose elements are called* concept names*,* $\mathcal{G}$ *is a finite set whose elements are called* object names*, and* $(\mathcal{R}, \leq_{\mathcal{R}})$ *is a set, partitioned into finite ordered sets* $(\mathcal{R}_k, \leq_{\mathcal{R}_k})_{k=1,\ldots,n}$ *whose elements are called* relation names.

Now, we can introduce concept graphs as statements formulated with these syntactical names. That means, we consider the concept graphs to be the well-formed formulas of our formal language. In accordance with the first mathematization of conceptual graphs in [Wi97], the structure of (simple) concept graphs is described by means of directed multi-hypergraphs and labeling functions.

Definition 2. *A* (simple) concept graph *over the alphabet* $(\mathcal{C}, \mathcal{G}, \mathcal{R})$ *is a structure* $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$*, where*

- (V, E, ν) *is a (not necessarily connected) finite directed multi-hypergraph, i. e. a structure where* V *and* E *are finite sets whose elements are called* vertices *and* edges *respectively, and* $\nu\colon E \to \bigcup_{k=1}^{n} V^k$ *is a mapping* ($n \geq 2$)*,*
- $\kappa\colon V \cup E \to \mathcal{C} \cup \mathcal{R}$ *is a mapping such that* $\kappa(V) \subseteq \mathcal{C}$ *and* $\kappa(E) \subseteq \mathcal{R}$*, and all* $e \in E$ *with* $\nu(e) = (v_1, \ldots, v_k)$ *satisfy* $\kappa(e) \in \mathcal{R}_k$*,*
- $\rho\colon V \to \mathfrak{P}(\mathcal{G})\backslash\{\emptyset\}$ *is a mapping.*

For an edge $e \in E$ *with* $\nu(e) = (v_1, \ldots, v_k)$*, we define* $|e| := k$*, and we write* $\nu(e)\big|_i := v_i$ *and* $\rho(e) := \rho(v_1) \times \ldots \times \rho(v_k)$*.*

Apart from some little differences, the concept graphs correspond to the simple conceptual graphs defined in [So84] or [CM92]. We only use multi-hypergraphs instead of bipartite graphs in the mathematization. The application ν assigns to every edge the tuple of all its incident vertices. The function κ labels the vertices and edges by concept and relation names, respectively, and the mapping ρ describes the references of every vertex. In contrast to Sowa, we allow references with more than one object name (i. e. individual marker) but no generic markers, i. e. existential quantifiers, yet. They can be introduced into the syntax easily (cf. [Pre98b]), but in this paper we want to put emphasis on the elementary language. That is why we can omit coreference links here which are only relevant in connection with generic markers.

3 Semantics for Concept Graphs

We agree with J.F.Sowa, when he writes about the importance of a semantics: "To make meaningful statements, the logic must have a theory of reference that determines how the constants and variables are associated with things in the universe of discourse." [So98, p. 27]

Usually, the semantics for conceptual graphs is given by the translation of conceptual graphs into first-order logic (cf. [So84] or [CM92]). For some notions and proofs, a set-theoretic, extensional semantics was developed (cf. [MC96]), but it is rarely used.

We define a semantics based on relational contexts. That means, we interpret the syntactical elements (concept, object and relation names) by concepts, objects and relations of a relational context. We prefer this *contextual semantics* for several reasons. As the basic elements of concept graphs are concepts, we want a semantics in which concepts are considered in a formal, but manifold way. Therefore, it is convenient to use Formal Concept Analysis, which is a mathematization of the philosophical understanding of concepts as units of thought constituted by their extension and intension (cf. [Wi96]). Furthermore, it is essential for Formal Concept Analysis that these two components of a concept are unified on the basis of a specified context. This contextual view is supported by Peirce's pragmatism which claims that we can only analyze and argue within restricted contexts where we always rely on preknowledge and common sense (cf. [Wi97]). Experience has shown that formal contexts are a useful basis for knowledge representation and communication because, on the one hand, they are close enough to reality and, on the other hand, their formalizations allow an efficient formal treatment.

As formal contexts do not formalize relations on the objects, the contexts must be enriched with relational structures. Therefore, R.Wille invented power context families in [Wi97] where relations are described as concepts with extensions consisting of tuples of objects. Using relational contexts in this paper, we have chosen a slightly simpler formalism. Nevertheless, this formalism can be transformed to power context families and vice versa. This is explained in detail in [Pre98a] and will not be discussed in this paper. Let us start with the formal definition of a relational context (originally introduced in [Pri96]).

Definition 3. *A* formal context (G, M, I) *is a triple where* G *and* M *are finite sets whose elements are called* objects *and* attributes*, respectively, and* I *is a binary relation between* G *and* M *which is called an* incidence relation*. A formal context, together with a set* $\mathfrak{R} := \bigcup_{k=1}^{n} \mathfrak{R}_k$ *of sets of* k*-ary relations on* G *is called* relational context *and denoted by* $\mathbb{K} := ((G, \mathfrak{R}), M, I)$*. The concept lattice* $\underline{\mathfrak{B}}(G, M, I) := (\mathfrak{B}(G, M, I), \leq)$ *is also called the* concept lattice of $\mathbb{K}$ *and denoted by* $\underline{\mathfrak{B}}(\mathbb{K})$*.*

For the basic notions in Formal Concept Analysis like the definition of the concept lattice, please refer to [GW98]. We just mention the denotation $g^I := \{m \in M / (g, m) \in I\}$ for $g \in G$ (and dually for $m \in M$) which will be used in the following paragraphs.

Now, we can specify how the syntactical elements of $(\mathcal{C}, \mathcal{G}, \mathcal{R})$ are interpreted in relational contexts by context-interpretations. The object names are interpreted by objects of the context, the concept names by its concepts and the relation names by its relations. In this way, we can embed the order given on $\mathcal{C}$ into the richer structure of the concept lattice. Order-preserving mappings are required because the interpretation shall respect the subsumptions given by the orders on $\mathcal{C}$ and $\mathcal{R}$.

Definition 4. *For an alphabet* $\mathcal{A} := (\mathcal{C}, \mathcal{G}, \mathcal{R})$ *and a relational context* $\mathbb{K} := ((G, \mathfrak{R}), M, I)$ *we call the union* $\iota := \iota_{\mathcal{C}} \dot{\cup} \iota_{\mathcal{G}} \dot{\cup} \iota_{\mathcal{R}}$ *of the mappings* $\iota_{\mathcal{C}}: (\mathcal{C}, \leq_{\mathcal{C}}) \to \underline{\mathfrak{B}}(\mathbb{K})$*,* $\iota_{\mathcal{G}}: \mathcal{G} \to G$ *and* $\iota_{\mathcal{R}}: (\mathcal{R}, \leq_{\mathcal{R}}) \to (\mathfrak{R}, \subseteq)$ *a* $\mathbb{K}$-interpetation *of* $\mathcal{A}$ *if* $\iota_{\mathcal{C}}$ *and* $\iota_{\mathcal{R}}$ *are order-preserving and we have* $\iota_{\mathcal{R}}(\mathcal{R}_k) \subseteq \mathfrak{R}_k$ *for all* $k = 1, \ldots, n$*. The tuple* $(\mathbb{K}, \iota)$ *is called a* context-interpretation *of* $\mathcal{A}$*.*

Having defined how the syntactical elements are related to elements of a relational context, we can explain formally how to distinguish valid statements from invalid statements. Due to our contextual view, the notion of validity also depends on the specified relational context. That means, a concept graph is called valid in a context-interpretation if the assigned objects belong to the extension of the assigned concepts, and if the assigned relations conform with the labels of the edges. Let us make these conditions precise in a formal definition.

Definition 5. *Let* $(\mathbb{K}, \iota)$ *be a context-interpretation of* $\mathcal{A}$*. The concept graph* $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$ *over* $\mathcal{A}$ *is called* valid *in* $(\mathbb{K}, \iota)$ *if*

- $\iota_{\mathcal{G}}\rho(v) \subseteq Ext(\iota_{\mathcal{C}}\kappa(v))$ *for all* $v \in V$ (vertex condition)
- $\iota_{\mathcal{G}}\rho(e) \subseteq \iota_{\mathcal{R}}\kappa(e)$ *for all* $e \in E$ (edge condition).

If $\mathfrak{G}$ *is valid in* $(\mathbb{K}, \iota)$*, then* $(\mathbb{K}, \iota)$ *is called a* model for $\mathfrak{G}$ *and* $\mathfrak{G}$ *is called a* concept graph of $(\mathbb{K}, \iota)$*.*

Note that, theoretically, any formal context could be completed to a model if the relations and the interpretation were chosen in the right way. For a given concept graph $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$, a formal context (G, M, I) and a given mapping $\iota_{\mathcal{G}}: \mathcal{G} \to G$, we can always define an order-preserving mapping $\iota_{\mathcal{C}}: \mathcal{C} \to \underline{\mathfrak{B}}(G, M, I)$ satisfying the vertex condition, for example the mapping with $\iota_{\mathcal{C}}(c) :=$

$\bigvee\{(\iota_{\mathcal{G}}\rho(v)^{II}, \iota_{\mathcal{G}}\rho(v)^{I}) \mid v \in V,\ \kappa(v) \leq_{\mathcal{C}} c\}$. Thus, we can obtain a model by choosing appropriate relations and a mapping $\iota_{\mathcal{R}}$. This shows that looking for an adequate model is not only a matter of formalism. It always depends on the specific purpose.

There is one interesting model for every concept graph, namely its standard model. It codes exactly the information given by the concept graph.

Definition 6. *Let $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$ be a concept graph over the alphabet $(\mathcal{C}, \mathcal{G}, \mathcal{R})$. We define the* standard model *of $\mathfrak{G}$ to be the relational context $\mathbb{K}^{\mathfrak{G}} := ((\mathcal{G}, \mathfrak{R}^{\mathfrak{G}}), \mathcal{C}, I^{\mathfrak{G}})$ together with the $\mathbb{K}^{\mathfrak{G}}$-interpretation $\iota^{\mathfrak{G}} := \iota_{\mathcal{C}} \dot\cup \iota_{\mathcal{G}} \dot\cup \iota_{\mathcal{R}}$ where $\iota_{\mathcal{C}}: \mathcal{C} \to \underline{\mathfrak{B}}(\mathbb{K}^{\mathfrak{G}})$ with $\iota_{\mathcal{C}}(c) := (c^{I^{\mathfrak{G}}}, c^{I^{\mathfrak{G}}I^{\mathfrak{G}}})$, $\iota_{\mathcal{G}} := id_{\mathcal{G}}$, $\mathfrak{R}^{\mathfrak{G}} := \iota_{\mathcal{R}}(\mathcal{R})$. The incidence relation $I^{\mathfrak{G}} \subseteq \mathcal{G} \times \mathcal{C}$ and the mapping $\iota_{\mathcal{R}}$ are defined in such a way that for all $g \in \mathcal{G}$, $c \in \mathcal{C}$, $(g_1, \ldots, g_k) \in \mathcal{G}^k$ and $R \in \mathcal{R}$, we have the following conditions:*

$$\begin{aligned}(g, c) \in I^{\mathfrak{G}} &\iff \exists\, v \in V: \quad \kappa(v) \leq_{\mathcal{C}} c \ \text{ and } \ g \in \rho(v)\\ (g_1, \ldots, g_k) \in \iota_{\mathcal{R}}(R) &\iff \exists\, e \in E: \quad \kappa(e) \leq_{\mathcal{R}} R \text{ and } (g_1, \ldots, g_k) \in \rho(e).\end{aligned}$$

We can read this definition as an instruction of how to construct the standard model. As objects of the context, we take all object names in $\mathcal{G}$, as attributes of the context, we take all concept names in $\mathcal{C}$ and we relate the object name g and the concept name $\kappa(v)$ (i. e., set $(g, \kappa(v)) \in I^{\mathfrak{G}}$) if the object name g belongs to the reference $\rho(v)$ of the vertex v. For preserving the order, we additionally relate g to every concept name c satisfying $\kappa(v) \leq_{\mathcal{C}} c$. Similarly for the relations. It is proved in [Pre98a] that this standard model is really a model for $\mathfrak{G}$.

Constructing a standard model for a given concept graph is the easiest way to find a relational context that codes exactly the information formalized in the concept graph. Thus, it allows us to translate knowledge expressed on the graphical level into knowledge on the contextual level. In the following section, we will see how the standard model helps to characterize inferences of concept graphs on the contextual level.

4 Reasoning with Concept Graphs

4.1 Entailment and Validity in the Standard Model

Having specified a formal semantics, we can easily describe inferences on the semantical level by entailments. For this, we only consider concept graphs over the same alphabet in the whole section. We recall the usual definition.

Definition 7. *Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two concept graphs over the same alphabet. We say that $\mathfrak{G}_1$* entails *$\mathfrak{G}_2$ if $\mathfrak{G}_2$ is valid in every model for $\mathfrak{G}_1$. We denote this by $\mathfrak{G}_1 \models \mathfrak{G}_2$.*

The following proposition explains how the entailment can be characterized by standard models (for the proof see appendix).

Proposition 1. *The concept graph $\mathfrak{G}_1$ entails the concept graph $\mathfrak{G}_2$ if and only if $\mathfrak{G}_2$ is valid in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$ of $\mathfrak{G}_1$.*

That means, using the contextual language, we obtain an effective method for deciding whether a concept graph entails another one or not. Beyond this, we could theoretically concentrate completely on the context level and describe the relation $\models$ by means of inclusion in the standard models as the following lemma shows.

Lemma 1. *Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two concept graphs over the same alphabet with standard models $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$ and $(\mathbb{K}^{\mathfrak{G}_2}, \iota^{\mathfrak{G}_2})$, respectively. They satisfy*

$$\mathfrak{G}_1 \models \mathfrak{G}_2 \quad \iff \quad I^{\mathfrak{G}_1} \supseteq I^{\mathfrak{G}_2} \text{ and } \iota_{\mathcal{R}}^{\mathfrak{G}_1}(R) \supseteq \iota_{\mathcal{R}}^{\mathfrak{G}_2}(R) \text{ for all } R \in \mathcal{R}.$$

Although the lemma is not very practical for reasoning in general, it has important consequences. Firstly, we can see easily that the relation $\models$ is reflexive and transitive, i. e., it is a preorder. Secondly, it implies that equivalent concept graphs (i. e. concept graphs with $\mathfrak{G}_1 \models \mathfrak{G}_2$ and $\mathfrak{G}_2 \models \mathfrak{G}_1$) have identical standard models.

Finally, we can characterize the order that is induced by the preorder $\models$ on the equivalence classes of concept graphs: the lemma shows how it can be characterized by the inclusions in the corresponding standard models. In particular, this allows us to describe the infimum and supremum of concept graphs by join and intersection in the standard model. The infimum of the two equivalence classes of the concept graphs $\mathfrak{G}_1$ and $\mathfrak{G}_2$ is the equivalence class of the juxtaposition $\mathfrak{G}_1 \oplus \mathfrak{G}_2$ (cf. [CM95]). It is not difficult to see that the standard model of this juxtaposition is exactly the standard model one obtains by "joining the standard models": for $(\mathbb{K}^{\mathfrak{G}_1 \oplus \mathfrak{G}_2}, \iota^{\mathfrak{G}_1 \oplus \mathfrak{G}_2})$, we have $I^{\mathfrak{G}_1 \oplus \mathfrak{G}_2} = I^{\mathfrak{G}_1} \cup I^{\mathfrak{G}_2}$ and $\iota_{\mathcal{R}}^{\mathfrak{G}_1 \oplus \mathfrak{G}_2}(R) = \iota_{\mathcal{R}}^{\mathfrak{G}_1}(R) \cup \iota_{\mathcal{R}}^{\mathfrak{G}_2}(R)$ for all $R \in \mathcal{R}$.

Whereas it is a difficult task to construct the supremum of two equivalence classes (if it exists at all), we can deduce immediately from Lemma 1 that its standard model $(\mathbb{K}, \iota)$ is the intersection of the standard models. That means, we have $I = I^{\mathfrak{G}_1} \cap I^{\mathfrak{G}_2}$ and $\iota_{\mathcal{R}}(R) = \iota_{\mathcal{R}}^{\mathfrak{G}_1}(R) \cap \iota_{\mathcal{R}}^{\mathfrak{G}_2}(R)$ for all $R \in \mathcal{R}$.

We conclude that describing the order induced by $\models$ is much easier on the context level than it is on the graph level. Especially suprema and infima can be characterized easily. This shows that for some purposes, it is convenient to translate the information given in concept graphs to the context level. For other purposes, it is interesting to do reasoning only on the syntactical level. But how can we characterize inferences on the syntactical level? It is usually done in two ways: by inference rules that were inspired by Peirce's inference rules for existential graphs and by projections, i. e. graph morphisms that can be supported by graph-theoretical methods and algorithms (cf. [So84], [MC96]).

4.2 Projections

Projections describe inferences from the perspective of graph morphisms. We recall the definition of projections as graph morphisms respecting the labeling functions (cf. [CM92]). It is slightly modified for concept graphs.

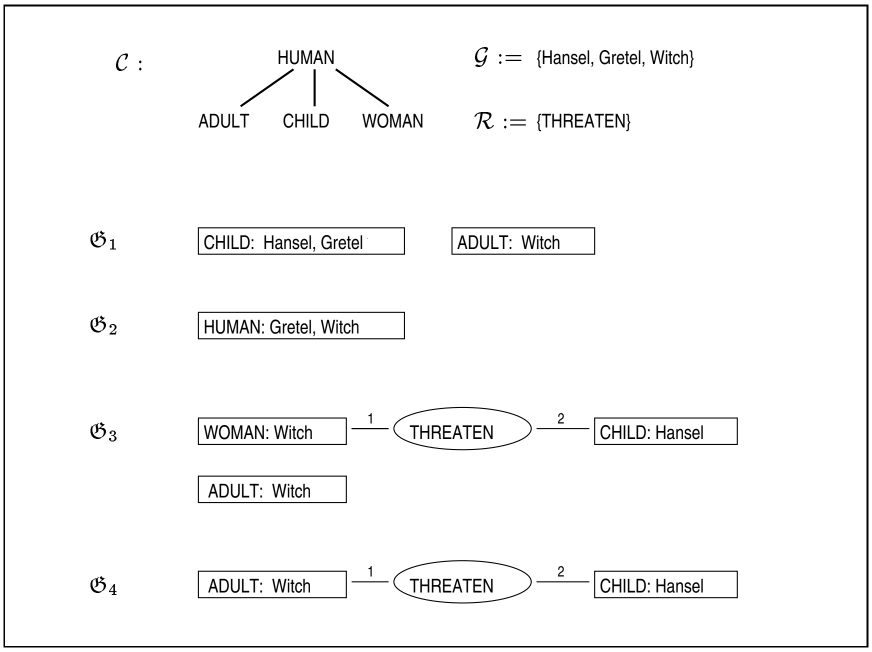

Fig. 1. Counter-examples to the Completeness of the Projection

Definition 8. *For the two concept graphs* $\mathfrak{G}_1 := (V_1, E_1, \nu_1, \kappa_1, \rho_1)$ *and* $\mathfrak{G}_2 := (V_2, E_2, \nu_2, \kappa_2, \rho_2)$ *over the same alphabet, a* projection *from* $\mathfrak{G}_2$ *to* $\mathfrak{G}_1$ *is defined as the union* $\pi_V \dot\cup \pi_E$ *of mappings* $\pi_V : V_2 \to V_1$ *and* $\pi_E : E_2 \to E_1$ *such that* $|e| = |\pi_E(e)|$, $\pi_V(\nu_2(e)|_i) = \nu_1(\pi_E(e))|_i$ *and* $\kappa_1(\pi_E(e)) \leq_{\mathcal{R}} \kappa_2(e)$ *for all edges* $e \in E_2$ *and for all* $i = 1, \ldots, |e|$*; and* $\kappa_1(\pi_V(v)) \leq_{\mathcal{C}} \kappa_2(v)$ *and* $\rho_1(\pi_V(v)) \supseteq \rho_2(v)$ *for all vertices* $v \in V_2$*. We write* $\mathfrak{G}_1 \lesssim \mathfrak{G}_2$ *if there exists a projection from* $\mathfrak{G}_2$ *to* $\mathfrak{G}_1$*.*

The relation $\lesssim$ defines a preorder on the class of all concept graphs, i. e., it is reflexive and transitive but not necessarily antisymmetric. It can be proved that this relation is characterized by the following inference rules (cf. [CM92], for concept graphs see [Pre98a]):

1. Double a vertex.
2. Delete an isolated vertex.
3. Double an edge.
4. Delete an edge.
5. Generalize a concept name.
6. Generalize a relation name.
7. Restrict a reference.
8. Copy the concept graph.

That means, two concept graphs $\mathfrak{G}_1$ and $\mathfrak{G}_2$ satisfy $\mathfrak{G}_1 \lesssim \mathfrak{G}_2$ if and only if $\mathfrak{G}_2$ can be derived from $\mathfrak{G}_1$ by applying these rules (which are elaborated more precisely in the appendix). Note that the Rule 7 (*restrict a reference*) is different to the restriction rule defined in [CM92]: Here, we cannot replace an individual maker by a general marker but delete an individual marker from the set of objects being the reference.

It can be proved that these rules are sound (cf. [Pre98a]). But as the examples in Figure 1 show, they are not complete. It is easy to see that $\mathfrak{G}_2$ must be valid in every model for $\mathfrak{G}_1$. On the other hand, it cannot be derived from $\mathfrak{G}_1$ because by these rules, references can only be restricted, not extended or joined. Even for concept graphs with references of only one element, the rules are not complete, when we consider redundant graphs. This is shown by the concept graphs $\mathfrak{G}_3$ and $\mathfrak{G}_4$. We have $\mathfrak{G}_3 \models \mathfrak{G}_4$, but the concept name WOMAN cannot be replaced by ADULT with the given rules.

4.3 A Sound and Complete Calculus for all Concept Graphs

There are two ways to treat the incompleteness: restricting the class of considered concept graphs (e. g. to concept graphs of normal form like in [MC96]) or extending the rules. As it is convenient for conceptual knowledge processing to allow all concept graphs instead of restricting them to normal forms, we decided to modify and extend the rules. (Note that the introduced rules are usually needed to transform a concept graph into normal form.)

Definition 9. *Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two concept graphs over the same alphabet. We call $\mathfrak{G}_2$* derivable *from $\mathfrak{G}_1$ and denote $\mathfrak{G}_1 \vdash \mathfrak{G}_2$ if it can be derived by the following inference rules (which are elaborated in the appendix):*

1. Double a vertex.
Double a vertex v and its incident edges (several times if v occurs more than once in $\nu(e)$). Extend the mappings κ and ρ to the doubles.

2. Delete an isolated vertex.
Delete a vertex v and restrict κ and ρ accordingly.

3. Double an edge.
Double an edge e and extend the mappings κ and ρ to the double.

4. Delete an edge.
Delete an edge e and restrict the mappings κ and ρ accordingly.

5.* Exchange a concept name.
Substitute the assignment $v \mapsto \kappa(v)$ for $v \mapsto c$ for such a concept name $c \in \mathcal{C}$ for which there is a vertex $w \in V$ with $\kappa(w) \leq_{\mathcal{C}} c$ and $\rho(v) \subseteq \rho(w)$.

6.* Exchange a relation name.
Substitute the assignment $e \mapsto \kappa(e)$ for $e \mapsto R$ for such a relation name $R \in \mathcal{R}$ for which there is an edge $f \in E$ with $\kappa(f) \leq_{\mathcal{C}} R$ and $\rho(e) \subseteq \rho(f)$.

7. Restrict a reference.
Replace the assignment $v \mapsto \rho(v)$ by $v \mapsto A$ with the subset $\emptyset \neq A \subseteq \rho(v)$.

8. Copy the concept graph.
Construct a concept graph that is identical to the first concept graph up to the names of vertices and edges.

9.* Join vertices with equal references.
Join two vertices $v, w \in V$ satisfying $\rho(v) = \rho(w)$ into a vertex $v \vee w$ with the same incident edges and references, and set $\kappa(v \vee w) = c$ for a $c \in \mathcal{C}$ with $\kappa(v) \leq_{\mathcal{C}} c$ and $\kappa(w) \leq_{\mathcal{C}} c$.

10.* Join vertices with corresponding edges.
Join two vertices $v, w \in V$ which have corresponding, but uncommon edges (i. e. for every edge $e \in E$ incident with v there exists an edge e' incident with w, and vice versa, with equal label and equal references, and there incident vertices only differ in v and w once) into a vertex $v \vee w$ with the same incident edges, $\kappa(v \vee w) = c$ for a $c \in \mathcal{C}$ with $\kappa(v) \leq_{\mathcal{C}} c$ and $\kappa(w) \leq_{\mathcal{C}} c$, and $\rho(v \vee w) = \rho(v) \cup \rho(w)$.

We will state these inference rules more precisely in the appendix and prove formally that they are sound and complete. Note that Rule 8 is redundant because it can be substituted by applying Rule 1 and 4.

Proposition 2 (Soundness and Completeness).
Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two concept graphs over the same alphabet. Then, we have

$$\mathfrak{G}_1 \models \mathfrak{G}_2 \quad \Longleftrightarrow \quad \mathfrak{G}_1 \vdash \mathfrak{G}_2.$$

Let us sum up what has been achieved. There are three possibilities for characterizing inferences on concept graphs. The usual model-theoretic way is the entailment (cf. Def. 7). On the syntactical level, we have a sound and complete set of inference rules (cf. Def. 9) whereas the projections cannot be used in the general case due to incompleteness. With their graphical character, the inference rules can visualize inferences and can be intuitively used to derive relatively similar concept graphs by hand. That is why they can support communication about reasoning to a certain degree. For implementation and general questions of decidability, it seems to be more convenient to use the third notion to characterize inferences, namely validity in the standard model (cf. Prop. 1).

5 The Standard Graph of a Relational Context

The construction of a standard model for a given concept graph gives not only an efficient mathematical method for reasoning, but also a mechanism to translate the knowledge given in concept graphs to knowledge formalized in relational contexts. This possibility to translate from the graphical level to the contextual level is important for the development of conceptual knowledge systems.

For such a conceptual knowledge system, the opposite direction is equally important. How can we translate knowledge given in relational contexts into the language of concept graphs? Obviously, we can state many different valid concept graphs for a given relational context. If we look for a so-called standard graph that codes the same information as the relational context, we have to look for a concept graph that entails all other valid concept graphs. For a similar purpose, R.Wille proposed a procedure to construct a *canonical concept graph* in [Wi97]. We will modify this procedure for our purpose here.

We start with a relational context, say $\mathbb{K} := ((G, \mathfrak{R}), M, I)$. For constructing a concept graph, we need an alphabet $(\mathcal{C}, \mathcal{G}, \mathcal{R})$. We define $\mathcal{C} := \underline{\mathfrak{B}}(\mathbb{K})$, $\mathcal{G} := G$ and $\mathcal{R} := \mathfrak{R}$. For every index $k = 1, \ldots, n$, we determine for every relation $R \in \mathfrak{R}_k$ all maximal k-tuples $(A_1, \ldots, A_k)$ of non-empty subsets of G being included in R. All those $(k+1)$-tuples $(R, A_1, \ldots, A_k)$ are collected in the set $E_{\mathbb{K}}$. That means, we define for $R \in \mathfrak{R}_k$ the set

$$\begin{aligned}\mathrm{Ref}^{\max}(R) := \{&A_1 \times \ldots \times A_k \subseteq R \mid \\ &B_1 \times \ldots \times B_k \subseteq R \text{ implies } B_1 \times \ldots \times B_k \not\supset A_1 \times \ldots A_k\}\end{aligned}$$

and obtain the set of edges

$$E_{\mathbb{K}} := \bigcup_{k=1,\ldots,n} \{(R, A_1, \ldots, A_k) \mid R \in \mathfrak{R}_k,\ A_1 \times \ldots \times A_k \in \mathrm{Ref}^{\max}(R)\}.$$

Now, we define

$$\begin{aligned}V_{\mathbb{K}} := \{&A \subseteq G \mid \text{there exists a } (R, A_1, \ldots, A_k) \in E_{\mathbb{K}} \text{ with } A = A_i \text{ for an } i \leq k\} \cup \\ &\{g^{II} \subseteq G \mid g \in G\},\end{aligned}$$

and set $\nu_{\mathbb{K}} : E_{\mathbb{K}} \to \bigcup_{k=1}^{n} V_{\mathbb{K}}^{k}$ with $\nu_{\mathbb{K}}(R, A_1, \ldots, A_k) := (A_1, \ldots, A_k)$ and $\kappa_{\mathbb{K}} : V_{\mathbb{K}} \cup E_{\mathbb{K}} \to \underline{\mathfrak{B}}(\mathbb{K}) \cup \mathfrak{R}$ where $\kappa_{\mathbb{K}}(R, A_1, \ldots, A_k) := R$ and $\kappa_{\mathbb{K}}(A) := (A^{II}, A^{I})$. Finally, we can choose $\rho_{\mathbb{K}}(A) := A$ for all $A \in V_{\mathbb{K}}$.

In this way, we obtain a concept graph $\mathfrak{G}(\mathbb{K}) := (V_{\mathbb{K}}, E_{\mathbb{K}}, \nu_{\mathbb{K}}, \kappa_{\mathbb{K}}, \rho_{\mathbb{K}})$ that is valid in $(\mathbb{K}, id)$ and is called the *standard graph* of $\mathbb{K}$.

Proposition 3. *The standard graph* $\mathfrak{G}(\mathbb{K})$ *of a relational context* $\mathbb{K}$ *entails every concept graph* $\mathfrak{G}'$ *that is valid in* $(\mathbb{K}, id)$.

This proposition (which is proved in the appendix) guarantees the demanded property of the standard graph. It is an irredundant graph that entails all concept graphs which are valid in its context. Thus, the standard graph is the counterpart to the standard model. With the standard model, we gather all the information given in the concept graph and have a tool to translate it from the graph level into the context level. Vice versa, we can translate information from the context level to the graph level by constructing the standard graph.

The relationship between a context $\mathbb{K}$ and the context $\mathbb{K}^{\mathfrak{G}(\mathbb{K})}$, belonging to the standard model $(\mathbb{K}^{\mathfrak{G}(\mathbb{K})}, \iota^{\mathfrak{G}(\mathbb{K})})$ of $\mathfrak{G}(\mathbb{K})$ can also be described: The proof of Prop. 3 shows that the context $\mathbb{K}^{\mathfrak{G}(\mathbb{K})}$ only differs from $\mathbb{K}$ because its set of attributes is not reduced and the attributes have different names. Their concept lattices are isomorphic.

Vice versa, starting with a concept graph $\mathfrak{G}$ and constructing the standard model $(\mathbb{K}^{\mathfrak{G}}, \iota^{\mathfrak{G}})$, we cannot say that the standard graph of $\mathbb{K}^{\mathfrak{G}}$ is isomorphic to $\mathfrak{G}$ in the formal sense because it is not a concept graph over the same alphabet. Nevertheless, it encodes the same information in an irredundant form.

6 Contextual Logic for Knowledge Representation

With the approach to contextual logic presented in this paper, we have proposed a logic for concept graphs that is equipped with a model-theoretic foundation and in which inferences can be characterized in multiple ways. From a computational point of view, an efficient method has been presented to do reasoning by checking validity in the corresponding standard models.

The major domain of application we have in mind for this logic, is conceptual knowledge representation and processing. In particular, the contextual semantics allows an integration of concept graphs into conceptual knowledge systems like TOSCANA that are based on Formal Concept Analysis. Vice versa, an integration of concept lattices and various methods of Formal Concept Analysis into tools for conceptual graphs is possible. For this purpose, the separation of syntax and semantics is less important than the possibility of expressing knowledge on two different levels, the graph level and the context level. With the standard model and the standard graph, we have developed two notions that help to translate knowledge from one level to the other. With it, the foundation is laid for conceptual knowledge systems which combine the advantages of both languages.

For example, we can imagine a system that codes knowledge in relational contexts and provides, with the concept graphs, a graphical language as interface and representation tool for knowledge. In such a system, the knowledge engineer could extend a given knowledge base by constructing new concept graphs over the existing alphabet. Then, implemented algorithms on the graph level or on the context level (whatever is more convenient for the special situation) could check whether the new concept graph is already valid in the context (i. e., the information is redundant) or whether it represents additional information. Concept lattices could be used to find the conceptual hierarchy on the concepts and to determine the conceptual patterns and dependencies of concepts and objects. Obviously, we could profit from all the methods and algorithms already existent for conceptual graphs.

The architecture of conceptual knowledge systems including relational contexts and concept graphs should be discussed, and the role of the different languages should be further explored. As the expressivity of the developed language is still quite limited, the extensions by quantifiers and nested concept graphs are considered in current research.

7 Appendix: Formal Proofs

Proof of Proposition 1. We only have to prove that $\mathfrak{G}_2$ is valid in an arbitrary model $(\mathbb{K}, \lambda)$ for $\mathfrak{G}_1$ with $\mathbb{K} := ((G, \mathfrak{R}), M, J)$ and $\lambda := \lambda_{\mathcal{G}} \dot{\cup} \lambda_{\mathcal{C}} \dot{\cup} \lambda_{\mathcal{R}}$ if $\mathfrak{G}_2$ is valid in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$ of $\mathfrak{G}_1$ with $\mathbb{K}^{\mathfrak{G}_1} = ((\mathcal{G}, \mathfrak{R}^{\mathfrak{G}_1}), \mathcal{C}, I^{\mathfrak{G}_1})$.

As a result of the vertex condition for $\mathfrak{G}_1$ in the model $(\mathbb{K}, \lambda)$, we have $\lambda_{\mathcal{G}}\, \rho_1(v) \subseteq Ext\, \lambda_{\mathcal{C}} \kappa_1(v) \subseteq Ext\, \lambda_{\mathcal{C}}(c)$ for all concept names $c \in \mathcal{C}$ and for all vertices $v \in V_1$ with $\kappa_1(v) \leq_{\mathcal{C}} c$ (because $\lambda_{\mathcal{C}}$ is order-preserving). It follows $\lambda_{\mathcal{G}}(\bigcup\{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} c\}) \subseteq Ext\, \lambda_{\mathcal{C}}(c)$ for all $c \in \mathcal{C}$. As a result of the vertex condition for $\mathfrak{G}_2$ in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$, we have $\rho_2(w) \subseteq$

$Ext\,\iota_{\mathcal{C}}^{\mathfrak{G}_1}(\kappa_2(w)) := \bigcup\{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} \kappa_2(w)\}$ for all vertices $w \in V_2$. This implies for all $w \in V_2$ the vertex condition $\lambda_{\mathcal{G}}(\rho_2(w)) \subseteq \lambda_{\mathcal{G}}(\bigcup\{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} \kappa_2(w)\}) \subseteq Ext\,\lambda_{\mathcal{C}}(\kappa_2(w))$. For the edge condition, one can proceed similarly. □

Proof of Soundness: $\mathfrak{G}_1 \vdash \mathfrak{G}_2 \Rightarrow \mathfrak{G}_1 \models \mathfrak{G}_2$.
Due to the transitivity of $\models$, it suffices to show soundness for each single inference rule. Therefore, we will give the exact definition of every inference rule by describing the derived concept graph $\mathfrak{G}_2$. Then, we can prove the entailment by using Prop. 1 and checking that $\mathfrak{G}_2$ is valid in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$ of $\mathfrak{G}_1 := (V_1, E_1, \nu_1, \kappa_1, \rho_1)$. Because of $\iota_{\mathcal{G}}^{\mathfrak{G}_1} := id_{\mathcal{G}}$, $\mathrm{Ext}(\iota_{\mathcal{C}}^{\mathfrak{G}_1} c) = \bigcup\{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} c\}$ for all $c \in \mathcal{C}$ and $\iota_{\mathcal{R}}^{\mathfrak{G}_1} R = \bigcup\{\rho_1(e) \mid e \in E_1, \kappa_1(e) \leq_{\mathcal{R}} R\}$ for all $R \in \mathcal{R}$ (cf. Def. 6), we only have to convince ourselves that $\mathfrak{G}_2$ satisfies the following vertex and edge conditions:

$$\forall w \in V_2 : \ \rho_2(w) \subseteq \bigcup\{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} \kappa_2(w)\} \qquad \text{(vertex condition)}$$
$$\forall f \in E_2 : \ \rho_2(f) \subseteq \bigcup\{\rho_1(e) \mid e \in E_1, \kappa_1(e) \leq_{\mathcal{R}} \kappa_2(f)\} \qquad \text{(edge condition).}$$

1. *Double a vertex.* The concept graph derived by doubling the vertex $v \in V_1$ is $\mathfrak{G}_2 := (V_2, E_2, \nu_2, \kappa_2, \rho_2)$ which is defined by
   - $V_2 := V_1 \backslash \{v\} \,\dot\cup\, \{(v,1),(v,2)\}$,
   - $E_2 := E_1 \backslash E_v \,\dot\cup\, E^v$ with
     $E_v := \{e \in E_1 \mid \nu_1(e)|_i = v$ for some $i = 1, \ldots, |e|\}$ and
     $E^v := \{(e,\delta) \mid e \in E_v,\ \delta \in \{1,2\}^{[e,v]}\}$ where $[e,v] := \{i \mid \nu_1(e)|_i = v\}$,
   - $\nu_2|_{E_1 \backslash E_v} := \nu_1|_{E_1 \backslash E_v}$ and
     $$\nu_2(e,\delta)|_i := \begin{cases} \nu_1(e)|_i & \text{if } i \notin [e,v] \\ (v,\delta(i)) & \text{if } i \in [e,v] \end{cases} \quad \text{for all } (e,\delta) \in E^v,$$
   - $$\begin{aligned} \kappa_2\colon V_2 \cup E_2 &\to \mathcal{C} \cup \mathcal{R} \\ x &\mapsto \kappa_1(x) \text{ for all } x \in V_1\backslash\{v\} \cup E_1 \backslash E_v \\ (v,j) &\mapsto \kappa_1(v) \text{ for } j = 1,2 \\ (e,\delta) &\mapsto \kappa_1(e) \text{ for all } (e,\delta) \in E^v, \end{aligned}$$
   - $\rho_2|_{V_1\backslash\{v\}} := \rho_1|_{V_1\backslash\{v\}}$ and $\rho_2(v,j) := \rho_1(v)$ for $j = 1,2$.

   For this derived concept graph $\mathfrak{G}_2$, the vertex and edge condition can be checked easily. It is left to the reader.
2. *Delete an isolated vertex.* If $v \in V_1$ is an isolated vertex of $\mathfrak{G}_1$ (i.e., there is no edge $e \in E_1$ and no $i = 1, \ldots, |e|$ with $\nu_1(e)|_i = v$), the components of the concept graph $\mathfrak{G}_2$ derived by deleting the isolated vertex v are defined as follows: $V_2 := V_1 \backslash \{v\}$, $E_2 := E_1$, $\nu_2 := \nu_1$, $\kappa_2 := \kappa_1|_{V_2 \cup E_1}$ and $\rho_2 := \rho_1|_{V_2}$. These components obviously satisfy the vertex and edge conditions.
3. *Double an edge.* The concept graph $\mathfrak{G}_2$ derived by doubling the edge $e \in E_1$ is defined by $V_2 := V_1$, $E_2 := E_1 \backslash \{e\} \cup \{(e,1),(e,2)\}$ where $(e,1),(e,2) \notin E_1$, $\nu_2|_{E_1\backslash\{e\}} := \nu_1|_{E_1\backslash\{e\}}$ and $\nu_2(e,j) := \nu_1(e)$ for $j = 1,2$, $\kappa_2|_{V_1 \cup (E_1\backslash\{e\})} := \kappa_1|_{V_1 \cup (E_1\backslash\{e\})}$ and $\kappa_2(e,j) := \kappa_1(e)$ for $j = 1,2$ and $\rho_2 := \rho_1$. It satisfies the vertex and edge conditions.

4. *Delete an edge.* Deleting the edge $e \in E_1$, one obtains the concept graph $\mathfrak{G}_2 := (V_1, E_1\backslash\{e\}, \nu_1|_{E_1\backslash\{e\}}, \kappa_1|_{V_1\cup(E_1\backslash\{e\})}, \rho_1)$ which satisfies the vertex and edge conditions.
5.* *Exchange a concept name.* The concept graph derived by substituting the concept name $\kappa_1(v)$ for a $c \in \mathcal{C}$ for which there is a vertex $w \in V_1$ with $\kappa_1(w) \leq_{\mathcal{C}} c$ and $\rho_1(v) \subseteq \rho_1(w)$, is defined by $\mathfrak{G}_2 := (V_1, E_1, \nu_1, \kappa_2, \rho_1)$ with $\kappa_2|_{(V_1\backslash\{v\})\cup E_1} := \kappa_1|_{(V_1\backslash\{v\})\cup E_1}$ and $\kappa_2(v) := c$. The edge condition is obviously satisfied, and the vertex condition is satisfied because $\kappa_1(w) \leq_{\mathcal{C}} c$ implies $\rho_1(v) \subseteq \rho_1(w) \subseteq \mathrm{Ext}\ \iota_{\mathcal{C}}^{\mathfrak{G}_1} c$.
6.* *Exchange a relation name.* The concept graph derived by substituting the relation name $\kappa_1(e)$ for such an $R \in \mathcal{R}$ for which there is an edge $f \in E_1$ with $\kappa_1(f) \leq_{\mathcal{C}} R$ and $\rho_1(e) \subseteq \rho_1(f)$, is defined by $\mathfrak{G}_2 := (V_1, E_1, \nu_1, \kappa_2, \rho_1)$ with $\kappa_2|_{V_1\cup(E_1\backslash\{e\})} := \kappa_1|_{V_1\cup(E_1\backslash\{e\})}$ and $\kappa_2(e) := R$. It satisfies the edge condition because $\kappa_1(f) \leq_{\mathcal{R}} R$ implies $\rho_1(e) \subseteq \rho_1(f) \subseteq \iota_{\mathcal{R}}^{\mathfrak{G}_1} R$.
7. *Restrict references.* The concept graph derived by restricting the reference $\rho_1(v)$ of the vertex $v \in V_1$ to the reference A with $\emptyset \neq A \subseteq \rho_1(v)$, is defined by $\mathfrak{G}_2 := (V_1, E_1, \nu_1, \kappa_1, \rho_2)$ with $\rho_2|_{V_1\backslash\{v\}} := \rho_1|_{V_1\backslash\{v\}}$ and $\rho_2(v) := A$. From $A \subseteq \rho_1(v)$ we deduce the vertex condition.
8. *Copy the concept graph.* For a copied concept graph $\mathfrak{G}_2$, there exist two bijections $\varphi_V: V_1 \to V_2$ and $\varphi_E : E_1 \to E_2$ such that $\kappa_1(v) = \kappa_2(\varphi_V(v))$ and $\rho_1(v) = \rho_2(\varphi_V(v))$ for all $v \in V_1$, and $\varphi_V(\nu_1(e)) = \nu_2(\varphi_E(e))$ and $\kappa_1(e) = \kappa_2(\varphi_E(e))$ for all $e \in E_1$. It trivially satisfies the vertex and edge conditions.
9*. *Join vertices with equal references.* The concept graph derived from $\mathfrak{G}_1$ by joining the two vertices v and w with equal references (i. e. with $\rho_1(v) = \rho_1(w)$) is $\mathfrak{G}_2 := (V_2, E_1, \nu_2, \kappa_2, \rho_2)$ with
   - $V_2 := V_1\backslash\{v, w\} \dot\cup \{v \vee w\}$,
   - $\nu_2|_i(e) := \begin{cases} v \vee w & \text{if } \nu_1(e)|_i = v \text{ or } \nu_1(e)|_i = w \\ \nu_1(e)|_i & \text{otherwise} \end{cases}$

     for all $e \in E_1$, $i = 1, \ldots, |e|$,
   - $\kappa_2|_{(V_1\backslash\{v,w\})\cup E_1} := \kappa_1|_{(V_1\backslash\{v,w\})\cup E_1}$ and $\kappa_2(v \vee w) := c$

     for a $c \in \mathcal{C}$ with $\kappa_1(v) \leq_{\mathcal{C}} c$ and $\kappa_1(w) \leq_{\mathcal{C}} c$,
   - $\rho_2|_{V_1\backslash\{v,w\}} := \rho_1|_{V_1\backslash\{v,w\}}$ and $\rho_2(v \vee w) := \rho_1(v)$.

   The vertex and edge conditions are satisfied from $\rho_1(v) = \rho_2(v\vee w)$, $\kappa_1(v) \leq_{\mathcal{C}} c$ and $\kappa_1(w) \leq_{\mathcal{C}} c$, we deduce $\mathrm{Ext}(\kappa_1(v)) \cup \mathrm{Ext}(\kappa_1(w)) \subseteq \mathrm{Ext}(\kappa_2(v \vee w))$.
10*. *Join vertices with corresponding edges.*
   Let us assume that the vertices $v, w \in V_1$ have corresponding, but uncommon edges, that means for every edge $e \in E_v$ (i. e., that is incident with v) there exists an edge $e' \in E_w$ and vice versa with $\kappa_1(e) = \kappa_1(e')$, $\nu_1(e)|_i = v$ for exactly one $i \in \{1, \ldots, |e|\}$ and $\nu_1(e')|_i = w$, $\nu_1(e)|_j \neq w$ and $\nu_1(e')|_j \neq w$ for all $j = 1, \ldots, |e|$, and $\rho_1(\nu_1(e)|_j) = \rho_1(\nu_1(e')|_j)$ if $\nu_1(e)|_j \neq v$.
   Then the concept graph derived from $\mathfrak{G}_1$ by joining the two vertices v and w is $\mathfrak{G}_2 := (V_2, E_1, \nu_2, \kappa_2, \rho_2)$ where V_2 and κ_2 are defined as in Rule 9*, and ρ_2 is defined by $\rho_2|_{V_1\backslash\{v,w\}} := \rho_1|_{V_1\backslash\{v,w\}}$ and $\rho_2(v \vee w) := \rho_1(v) \cup \rho_1(w)$. The

vertex and edge conditions are satisfied because $\kappa_1(v) \leq_{\mathcal{C}} c$ and $\kappa_1(w) \leq_{\mathcal{C}} c$ imply Ext $\kappa_1(v) \cup$ Ext $\kappa_1(w) \subseteq$ Ext $\kappa_2(v \vee w)$. □

Proof of Completeness: $\mathfrak{G}_1 \models \mathfrak{G}_2 \Rightarrow \mathfrak{G}_1 \vdash \mathfrak{G}_2$.
We will prove completeness by using so-called *stars*, which are concept graphs with only one edge and its incident vertices. For a given concept graph $\mathfrak{G} := (V, E, \nu, \kappa, \rho)$, the *stars of* $\mathfrak{G}$ are all those stars which are subgraphs of $\mathfrak{G}$, i. e. all concept graphs $\mathfrak{G}' := (V', E', \nu|_{V' \cup E'}, \kappa|_{V' \cup E'}, \rho|_{V' \cup E'})$ where $E' := \{e\}$ for an edge $e \in E$ and $V' := \{\nu(e)|_i \mid i = 1, \ldots, |e|\}$. The stars are interesting because we can derive a concept graph from the set of all its stars and its isolated vertices using Rule 9* (*join vertices with equal references*). Consequently, it suffices to prove that every star $\mathfrak{A}$ of $\mathfrak{G}_2$ can be derived from $\mathfrak{G}_1$ if $\mathfrak{G}_1$ entails $\mathfrak{G}_2$. Using Rule 8 (*copy concept graph*), we obtain enough copies to derive all stars of $\mathfrak{G}_2$ from which we can derive $\mathfrak{G}_2$.

Let $\mathfrak{G}_1$ and $\mathfrak{G}_2$ be two concept graphs with $\mathfrak{G}_1 \models \mathfrak{G}_2$ and let $\mathfrak{A}$ be a star of $\mathfrak{G}_2$ with edge f and vertices $w_1, w_2, \ldots, w_k$. For deriving $\mathfrak{A}$ from $\mathfrak{G}_1$, we proceed in three steps.

(i.) First, we derive stars from $\mathfrak{G}_1$ such that, for every tuple $(g_1, \ldots, g_k)$ of objects in $\rho(f)$, there is a star $\mathfrak{A}^{g_1,\ldots,g_k}$ with edge $e_{g_1,\ldots,g_k}$ and $\rho(e_{g_1,\ldots,g_k}) = (g_1, \ldots, g_k)$.

(ii.) Then, we join these stars in several steps by joining the corresponding vertices. We obtain a star $\mathfrak{B}$ with an edge f' that has the same references as the star $\mathfrak{A}$ of $\mathfrak{G}_2$. But it does not necessarily have the same concept and relation names.

(iii.) In order to adapt the concept and relation names by Rules 5* and 6*, we first have to derive isolated vertices v_i for every vertex w_i of $\mathfrak{A}$ with $\kappa_1(v_i) = \kappa_2(w_i)$ and $\rho_1(v_i) \supseteq \rho_2(w_i)$. Then, we can finally deduce a copy of $\mathfrak{A}$ from $\mathfrak{B}$.

i) As $\mathfrak{A}$ is valid in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$ and $\kappa_2(f) \in \mathcal{R}$, there exists a set $T \subseteq \kappa_1(E_1)$ of relations such that $\iota_{\mathcal{R}}^{\mathfrak{G}_1}\kappa_2(f) = \bigcup \{\iota_{\mathcal{R}}^{\mathfrak{G}_1} R \mid R \in T\}$. Consequently, for all $(g_1, \ldots, g_k) \in \rho_2(f)$, there exists an $R \in T$ such that $\iota_{\mathcal{G}}^{\mathfrak{G}_1}(g_1, \ldots, g_k) = (g_1, \ldots, g_k) \in \iota_{\mathcal{R}}^{\mathfrak{G}_1}(R)$. Because of $R \in \kappa_1(E_1)$, we can find an edge $e_{g_1,\ldots,g_k} \in E_1$ with $(g_1, \ldots, g_k) \in \rho_1(e_{g_1,\ldots,g_k})$.

By means of Rule 2 and 4 (*delete vertices and edges*), we can derive, for all tuples $(g_1, \ldots, g_k) \in \rho_2(f)$, the corresponding star of $\mathfrak{G}_1$ with the edge $e_{g_1,\ldots,g_k}$. Using Rule 7 (*restrict references*), we restrict the references to $g_1, \ldots, g_k$. In this way, we derive stars denoted by $\mathfrak{A}^{g_1,\ldots,g_k}$ with vertices denoted by $v_{g_1}, \ldots, v_{g_k}$.

ii) In the first substep, we join the k^{th} vertices of all stars $\mathfrak{A}^{g_1,\ldots,g_k}$ where the first $k-1$ references are identical. For every tuple $(\overline{g}_1, \ldots, \overline{g}_{k-1}) \in \rho_2(w_1) \times \ldots \times \rho_2(w_{k-1})$, we consider all stars $\mathfrak{A}^{\overline{g}_1,\ldots,\overline{g}_{k-1},g_k}$ with $g_k \in \rho_2(w_k)$ and unify the relation names $\kappa(e_{\overline{g}_1,\ldots,\overline{g}_{k-1},g_k})$ by Rule 6* (*exchange relation names*) into a common relation name $R_{e_{\overline{g}_1,\ldots,\overline{g}_{k-1}}}$. As all g_k belong to $\rho_2(w_k)$, they satisfy $\kappa(e_{\overline{g}_1,\ldots,\overline{g}_{k-1},g_k}) \leq_{\mathcal{R}} \kappa(f)$. Thus, we find a common relation name $R_{e_{\overline{g}_1,\ldots,\overline{g}_{k-1}}} \leq_{\mathcal{R}} \kappa(f)$.

Thereafter, we join the k^{th} vertices of all changed concept graphs $\mathfrak{A}^{\overline{g}_1,\ldots,\overline{g}_{k-1},g_k}$ by Rule 10* (*join vertices with corresponding edges*). Then, we join their first, then second, and finally $(k-1)^{\text{th}}$ vertices. After deleting the double edges (Rule 4), we obtain a star with k vertices that we denote by $\mathfrak{A}^{\overline{g}_1,\ldots,\overline{g}_{k-1}}$. It has an edge $e_{\overline{g}_1,\ldots,\overline{g}_{k-1}}$, and we have $\rho(e_{\overline{g}_1,\ldots,\overline{g}_{k-1}}) = \{\overline{g}_1\} \times \ldots \times \{\overline{g}_{k-1}\} \times \rho_2(w_k)$ and $\kappa(e_{\overline{g}_1,\ldots,\overline{g}_{k-1}}) \leq_{\mathcal{R}} \kappa(f)$.

In the second substep, we join the vertices of all those stars $\mathfrak{A}^{g_1,\ldots,g_{k-1}}$ (which all have the same k^{th} reference) that correspond in the $(k-1)^{\text{th}}$ reference. Applying Rule $6^*, 10^*$ and 4, we obtain concept graphs $\mathfrak{A}^{g_1,\ldots,g_{k-2}}$ with the edge $e_{g_1,\ldots,g_{k-2}}$ satisfying $\rho(e_{g_1,\ldots,g_{k-2}}) = \{g_1\} \times \ldots \times \{g_{k-2}\} \times \rho_2(w_{k-1}) \times \rho_2(w_k)$. After k steps of joining, we obtain a star $\mathfrak{B}$ with edge f' that has the same references as the edge f of $\mathfrak{A}$.

iii) As $\mathfrak{A}$ is valid in the standard model $(\mathbb{K}^{\mathfrak{G}_1}, \iota^{\mathfrak{G}_1})$, every vertex w_i of $\mathfrak{A}$ satisfies $\rho_2(w_i) \subseteq Ext(\iota_{\mathcal{G}}^{\mathfrak{G}_1} \kappa_2(w_i)) = \bigcup \{\rho_1(v) \mid v \in V_1, \kappa_1(v) \leq_{\mathcal{C}} \kappa_2(w_i)\}$. Thus, for every vertex w_i of $\mathfrak{A}$, we can use Rule 4 (*delete edges*) and derive all isolated vertices $v \in V_1$ with $\kappa_1(v) \leq_{\mathcal{C}} \kappa_2(w_i)$. By means of Rule 10* (*join vertices with corresponding edges*), they can be joined into an isolated vertex v_i with $\kappa_1(v_i) = \kappa_2(w_i)$ and $\rho_1(v_i) \supseteq \rho_2(w_i)$.

Finally, we can exchange the concept and relation names (Rules 5* and 6*) and, by means of Rule 2 (*delete all isolated vertices*), we obtain a concept graph that is isomorphic to $\mathfrak{A}$. Taken as a whole, this proves $\mathfrak{G}_1 \vdash \mathfrak{A}$. □

Proof of Proposition 3. Let $\mathfrak{G}'$ be valid in $(\mathbb{K}, id)$. We prove the assertion by showing that $\mathfrak{G}'$ is valid in the standard model of the standard graph $\mathfrak{G}(\mathbb{K}) := \mathfrak{G}$ and by using Prop. 1. The standard model $(\mathbb{K}^{\mathfrak{G}}, \iota^{\mathfrak{G}})$ of the concept graph $\mathfrak{G}$ with $\mathbb{K}^{\mathfrak{G}} = ((G, \mathfrak{R}), \underline{\mathfrak{B}}(\mathbb{K}), I^{\mathfrak{G}})$ satisfies $\iota_{\mathcal{G}}^{\mathfrak{G}} = id_G$, and $\iota_{\mathfrak{R}}^{\mathfrak{G}}(R) = \bigcup \{\rho_{\mathbb{K}}(e) \mid e \in E, \kappa_{\mathbb{K}}(e) \leq R\} = R$ for all $R \in \mathfrak{R}$. This implies $\iota_{\mathfrak{R}}^{\mathfrak{G}} = id_{\mathfrak{R}}$. Consequently, the edge condition for $\mathfrak{G}'$ in $(\mathbb{K}^{\mathfrak{G}}, \iota^{\mathfrak{G}})$ is satisfied. Furthermore, we have $\iota_{\mathcal{C}}^{\mathfrak{G}}(c) := (c^{I^{\mathfrak{G}}}, c^{I^{\mathfrak{G}} I^{\mathfrak{G}}})$ and, according to the definition of the incidence relation in the standard model, we have for every concept name $c \in \mathcal{C}$ the equations $c^{I^{\mathfrak{G}}} := \bigcup \{\rho_{\mathbb{K}}(A) \mid \kappa_{\mathbb{K}}(A) \leq c\} = \bigcup \{g^{II} \mid (g^{II}, g^I) \leq c\} = \bigcup \{g^{II} \mid g \in c^I\} = c^I$. Thus, the vertex condition is also satisfied. □

References

[CM92] Chein, M., Mugnier, M.-L.: Conceptual Graphs: Fundamental Notions. Revue d'Intelligence Artificielle 6 (1992) 365–406

[CM95] Chein, M., Mugnier, M.-L.: Conceptual Graphs are also Graphs. Rapport de Recherche 95003, LIRMM, Université Montpellier II (1995)

[GW98] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin–Heidelberg, to appear 1998

[MC96] Mugnier, M.-L., Chein, M.: Représenter des Connaissances et Raisonner avec des Graphes. Revue d'Intelligence Artificielle 10 (1996) 7–56

[Pre98a] Prediger, S.: Einfache Begriffsgraphen: Syntax und Semantik. Preprint, FB Mathematik, TU Darmstadt (1998)

[Pre98b] Prediger, S.: Existentielle Begriffsgraphen. Preprint, FB Mathematik, TU Darmstadt (1998)

[Pri96] Priß, U.: The Formalization of WordNet by Methods of Relational Concept Analysis. In: Fellbaum, C. (ed.): WordNet - An Electronic Lexical Database and some of its Applications. MIT-Press (1996)

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

[So98] Sowa, J. F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. PWS Publishing Co., Boston, to appear 1998

[We95] Wermelinger, M.: Conceptual Graphs and First-Order Logic. In: Ellis, G. et al. (eds.): Conceptual Structures: Applications, Implementations and Theory, Proceedings of the ICCS '95. Springer, Berlin–New York (1995) 323–337

[Wi96] Wille, R.: Restructuring Mathematical Logic: an Approach based on Peirce's Pragmatism. In: Ursini, A., Agliano, P. (eds.): Logic and Algebra. Marcel Dekker, New York (1996) 267–281

[Wi97] Wille. R.: Conceptual Graphs and Formal Concept Analysis. In: Lukose, D. et. al. (eds.): Conceptual Structures: Fulfilling Peirce's Dream, Proceedings of the ICCS'97. Springer, Berlin–New York (1997) 290–303