Tractable and Decidable Fragments of Conceptual Graphs*

F. Baader, R. Molitor, and S. Tobies

LuFg Theoretical Computer Science,
RWTH Aachen, Germany.
E-mail: {baader,molitor,tobies}@informatik.rwth-aachen.de

Abstract. It is well-known that problems like validity and subsumption of general CGs are undecidable, whereas subsumption is NP-complete for simple conceptual graphs (SGs) and tractable for SGs that are trees. We will employ results on decidable fragments of first-order logic to identify a natural and expressive fragment of CGs for which validity and subsumption is decidable in ExpTime. In addition, we will extend existing work on the connection between SGs and description logics (DLs) by identifying a DL that corresponds to the class of SGs that are trees. This yields a tractability result previously unknown in the DL community.

1 Introduction

Conceptual graphs (CGs) are an expressive formalism for representing knowledge about an application domain in a graphical way. Since CGs can express all of first-order predicate logic (FO), they can also be seen as a graphical notation for FO formulae.

In knowledge representation, one is usually not only interested in *representing* knowledge, one also wants to *reason* about the represented knowledge. For CGs, one is, for example, interested in validity of a given graph, and in the question whether one graph subsumes another one. Because of the expressiveness of the CG formalism, these reasoning problems are undecidable for general CGs. In the literature [14, 16, 12] one can find complete calculi for validity of CGs, but implementations of these calculi have the same problems as theorem provers for FO: they may not terminate for formulae that are not valid, and they are very inefficient. To overcome this problem, one can either employ incomplete reasoners, or try to find decidable (or even tractable) fragments of the formalism. This paper investigates the second alternative.

The most prominent decidable fragment of CGs is the class of simple conceptual graphs (SGs), which corresponds to the conjunctive, positive, and existential fragment of FO (i.e., existentially quantified conjunctions of atoms). Even for this simple fragment, however, subsumption is still an NP-complete problem [5]. SGs that are trees provide for a tractable fragment of SGs, i.e., a class of simple

* This work was partially supported by the *Deutsche Forschungsgemeinschaft* Grant No. GRK 185/3-98 and Project No. GR 1324/3-1

conceptual graphs for which subsumption can be decided in polynomial time [13]. In this paper, we will, on the one hand, describe a decidable fragment of CGs that is considerably more expressive than SGs. On the other hand, we will identify a tractable fragment of SGs that is larger than the class of trees.

Instead of trying to prove new decidability or tractability results for CGs from scratch, our idea was to transfer decidability results from first-order logics [4] and from description logics [9, 10] to CGs. The goal was to obtain "natural" sub-classes of the class of all CGs in the sense that these sub-classes are defined directly by syntactic restrictions on the graphs, and not by conditions on the first-order formulae obtained by translating CGs into FO.

Although description logics (DLs) and CGs are employed in very similar applications (e.g., for representing the semantics of natural language sentences), it turned out that these two formalisms are quite different for several reasons: (1) conceptual graphs[1] are interpreted as closed FO formulae, whereas DL concept descriptions are interpreted by formulae with one free variable; (2) most DLs do not allow for relations of arity > 2; (3) SGs are interpreted by existential sentences, whereas almost all DLs considered in the literature allow for universal quantification; (4) because DLs use a variable-free syntax, certain identifications of variables expressed by cycles in SGs and by co-reference links in CGs cannot be expressed in DLs. As a consequence of these differences, we could not identify a natural fragment of CGs corresponding to an expressive DL whose decidability was already shown in the literature. We could, however, obtain a new tractability result for a DL corresponding to SGs that are rooted, arc- and node labeled trees. This correspondence result strictly extends the one in [7]. In addition, we have extended the tractability result from SGs that are trees to SGs that can be transformed into trees using a certain "cycle-cutting" operation.

An interesting decidable fragment of FO, which has recently been introduced by van Benthem [15], is the so-called *loosely guarded fragment* of FO. It contains (the first-order translations of) many modal logics and description logics, but is not restricted to unary and binary relations. We could identify a fragment of CGs corresponding to the loosely guarded fragment of FO in the sense that the first-order translation of CGs belonging to this fragment are equivalent (though not necessarily identical) to a loosely guarded formula, and every loosely guarded formula can be obtained in this manner. The characterization of this fragment is given by syntactic restrictions on the graphs, and for a given graph it is easily decidable whether it belongs to this fragment.

2 Preliminaries

To fix our notation, we recall basic definitions and results on conceptual graphs. Basic ontological knowledge from the application domain is coded in the **support**, which is a structure of the form $\mathcal{S} = \langle N_C, N_R, N_I \rangle$. Here N_C is the set of *concept types*, which is ordered by a partial order $\leq_C$ expressing the *is-a-kind-of*

[1] Here, we restrict our attention to the first order fragment of CGs.

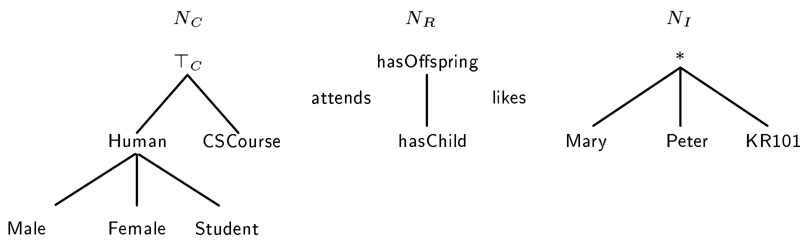

Fig. 1. An example of a support.

relation between concept types. We require N_C to contain a distinguished element $\top_C$ representing the entire domain, i.e., $\top_C$ is the greatest element w.r.t. $\leq_C$. Similarly, the set of *relation types*, N_R, is ordered by a partial order $\leq_R$. Each element of N_R has a fixed arity, and relation types with different arity are incomparable by $\leq_R$. The set of *individual markers* is denoted by N_I; an additional *generic marker* is denoted by $*$. We define a partial order $\leq_I$ on $N_I \cup \{*\}$ such that $*$ is the greatest element, and all other elements are pairwise incomparable.

Throughout the paper, we consider examples over the support $\langle N_C, N_R, N_I \rangle$ shown in Fig. 1, where all relation types are assumed to have arity 2.

A **simple graph** (SG) over the support $\mathcal{S}$ is a labeled bipartite graph of the form $g = \langle C, R, E, \ell \rangle$, where C and R are the node sets, called *concept nodes* and *relation nodes*, respectively, and $E \subseteq C \times R$ is the edge relation. The labeling ℓ labels g in the following way: Each concept node $c \in C$ is labeled by a pair $\ell(c) = (\mathit{type}(c), \mathit{ref}(c)) \in N_C \times (N_I \cup \{*\})$, called the *type* and the *referent* of c. If $\mathit{ref}(c) = *$, then c is called a *generic concept node*, otherwise c is called an *individual concept node*. Each relation node $r \in R$ is labeled with a relation type $\ell(r) \in N_R$. All edges that are incident to the same relation node are labeled by ℓ with sets of natural numbers in such a way that, if $\ell(r)$ has arity n, then all numbers from $\{1, \ldots, n\}$ appear exactly once in these labels. The concept node c linked to r by an edge with $j \in \ell(c, r)$ is called the jth neighbor of r and is denoted by $r(j)$. The set of all simple graphs over $\mathcal{S}$ is denoted by $SG(\mathcal{S})$.

SGs can be combined into more complex structures called **graph propositions**. A graph proposition p is a negated graph proposition, or a box that contains a SG (which may be empty) and finitely many graph propositions (see Fig. 2). These boxes are also called the *contexts* of the proposition. We require that all simple graphs appearing in a graph proposition have disjoint node sets. In a linear notation one can represent graph propositions as expressions generated by the EBNF grammar

$$p ::= \left[g \; p^* \right] \mid \neg p,$$

where g stands for a SG. For each SG g in p there is exactly one context p' which contains g at top level. This context is called *the context of* g, and we say that p' *contains* all nodes of g. We say that a context p *(strictly) dominates* a context q

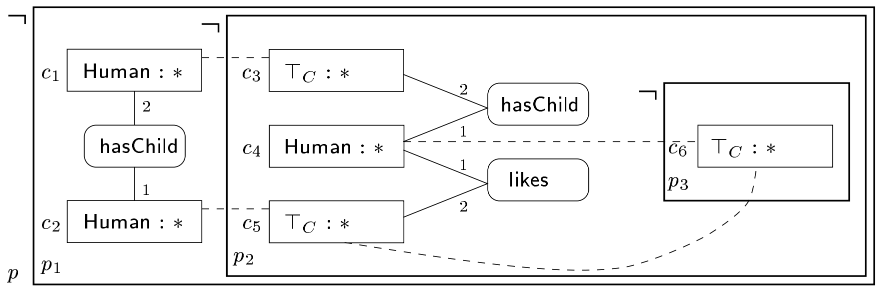

Fig. 2. An example of a conceptual graph.

iff q is (strictly) contained in p. The set of all concept nodes of all simple graphs occurring in p is denoted by $C(p)$.

A **conceptual graph** (CG) over the support $\mathcal{S}$ is a pair $G = \langle p, \mathit{coref}\rangle$, where p is a graph proposition over $\mathcal{S}$ and *coref*, the set of *coreference links*, is a symmetric binary relation over $C(p)$ satisfying the following property: for each pair of concept nodes $(c_1, c_2) \in \mathit{coref}$ contained in the contexts p_1 and p_2, respectively, either p_1 dominates p_2 or p_2 dominates p_1. We denote the set of all CGs over $\mathcal{S}$ by $CG(\mathcal{S})$.

An example of a CG built over the support from Fig. 1 is shown in Fig. 2 using the usual graphical notation. It asserts that for each parent and child there is another parent of that child who likes the first parent (see the translation of this graph into FO below). The extra markers p_i and c_i will be used in a later section of this paper to refer to the different parts of the CG.

Both SGs and CGs are given a semantics in FO by the **operator** Φ. Let $G = \langle p, \mathit{coref}\rangle \in CG(\mathcal{S})$ be the CG to be translated, and let $\mathcal{V}$ be a countably infinite set of variables. Firstly, we fix two mappings *id* and *links* as follows: *id* assigns a unique variable to each generic concept node $c \in C(p)$, and its individual marker to each individual concept node; we define $\mathit{links}(c)$ to consist of $\mathit{id}(c')$ of all concept nodes c' that are linked to c by a coreference link and are contained in a context dominating the context of c. The triple $\widehat{G} = \langle p, \mathit{id}, \mathit{links}\rangle$ obtained this way is translated by Φ into FO as follows:

– For a SG $g = \langle C, R, E, \ell\rangle$ we define $\phi(g) = \bigwedge_{c\in C} \phi(c) \wedge \bigwedge_{r\in R} \phi(r)$, where

$$\phi(c) := \bigwedge_{s\in \mathit{links}(c)} \mathit{id}(c) \doteq s \wedge \begin{cases} \mathit{id}(c) \doteq \mathit{id}(c) & \text{if } \mathit{type}(c) = \top_C \\ P(\mathit{id}(c)) & \text{if } \mathit{type}(c) = P \end{cases},$$

and, for a relation node r with $\ell(r) = S$ of arity n,

$$\phi(r) := S(\mathit{id}(r(1)), \ldots, \mathit{id}(r(n))).$$

The *quantifier prefix* of g is $\phi_p(g) := \exists x_1 \ldots \exists x_k$, where $\{x_1, \ldots, x_k\} := \{\mathit{id}(c) \mid c \in C\} \cap \mathcal{V}$.

– The *operator* Φ is defined by induction on the structure of graph propositions:
  - $\Phi[g\ p_1 \dots p_m] := \phi_p(g).\Big(\phi(g) \wedge \bigwedge_{j=1,\dots,m} \Phi(p_j)\Big)$,
  - $\Phi[\neg p] := \neg\Phi[p]$.

For a SG g we define its FO semantics $\Phi(g)$ by $\phi_p(g).\phi(g)$.

To translate the graph G from Fig. 2, we set $id(c_i) = x_i$ $(i = 1, \dots, 6)$. For *links*, this yields $links(c_3) = \{x_1\}$, $links(c_5) = \{x_2\}$, $links(c_6) = \{x_4, x_5\}$, and $links(c_i) = \emptyset$ for $i = 1, 2, 4$. After eliminating equalities of the form $x_i \doteq x_i$ we obtain the following FO formula:

$$\begin{aligned}\Phi(G) = \neg(\exists x_1 x_2.(&\mathsf{Human}(x_1) \wedge \mathsf{Human}(x_2) \wedge \mathsf{hasChild}(x_2, x_1) \wedge \\ &\neg(\exists x_3 x_4 x_5.(x_3 \doteq x_1 \wedge x_5 \doteq x_2 \wedge \mathsf{Human}(x_4) \wedge \\ &\quad \mathsf{hasChild}(x_4, x_3) \wedge \mathsf{likes}(x_4, x_5) \wedge \\ &\quad \neg(\exists x_6.(x_6 \doteq x_4 \wedge x_6 \doteq x_5))))))\end{aligned}$$

Note that the sub-formula $\neg(\exists x_6.(x_6 \doteq x_4 \wedge x_6 \doteq x_5)$ only expresses $x_4 \not\doteq x_5$.

We can also define the semantics of the order relations in the support by a FO formula. For a given support $\mathcal{S} = \langle N_C, N_R, N_I \rangle$, the partial orders $\leq_C$ and $\leq_R$ are interpreted as follows: $P_1 \leq_C P_2$ corresponds to the formula $\forall x.P_1(x) \to P_2(x)$, and for two relation types of arity n, $S_1 \leq_R S_2$ yields the formula $\forall x_1 \dots x_n.S_1(x_1, \dots, x_n) \to S_2(x_1, \dots, x_n)$. We define $\Phi(\mathcal{S})$ to be the conjunction of all these formulae.

Validity with respect to a support $\mathcal{S}$ for a CG G can be defined with the help of the operator Φ: G is *valid* iff $\Phi(\mathcal{S}) \to \Phi(G)$ is a valid FO formula.

Subsumption with respect to a support $\mathcal{S}$ for two SGs or CGs G, H is defined as follows: G is *subsumed* by H $(G \sqsubseteq H)$ iff $\Phi(\mathcal{S}) \wedge \Phi(G) \to \Phi(H)$ is a valid FO formula.

A SG g is said to be in **normal form** iff each individual marker $a \in N_I$ appears at most once as a referent of a concept node in g. Subsumption of two simple graphs g, h can be characterized by the existence of certain homomorphisms from h to g. To be more precise, if there exists a homomorphism from h to g, then $g \sqsubseteq h$ [14], and if $g \sqsubseteq h$ then there is such a homomorphism provided that g is in normal form [5].

Subsumption for SGs over a support $\mathcal{S}$ is an NP-complete problem [5]. Like Peirce's existential graphs, CGs are as expressive as FO formulae [14]. Thus, validity and subsumption for CGs are undecidable.

3 A Tractable Fragment of Simple Graphs

In this section, we introduce the description logic $\mathcal{ELIRO}^1$ as well as the class of rooted SGs. We will show that $\mathcal{ELIRO}^1$-concept descriptions can be translated into equivalent rooted SGs that are trees, and thus that subsumption in $\mathcal{ELIRO}^1$ can be decided in polynomial time. In addition, we extend the known tractability result for trees to a larger fragment of SGs.

Table 1. Syntax and semantics of $\mathcal{ELIRO}^1$-concept descriptions.

Construct name	Syntax	Semantics	
top-concept	$\top$	$x = x$	
primitive concept $P \in N_C$	P	$P(x)$	
conjunction	$C \sqcap D$	$\Psi_C(x) \wedge \Psi_D(x)$	$\mathcal{EL}$
existential restriction	$\exists r.C$	$\exists y.\Psi_r(x,y) \wedge \Psi_C(y)$	
constant $a \in N_I$	$\{a\}$	$x = a$	$\mathcal{O}^1$
primitive role $r \in N_R$	r	$r(x,y)$	
inverse role for $r \in N_R$	r^-	$r(y,x)$	$\mathcal{I}$
role conjunction	$r_1 \sqcap r_2$	$\Psi_{r_1}(x,y) \wedge \Psi_{r_2}(x,y)$	$\mathcal{R}$

Description Logics

In DLs, knowledge from an application domain is represented by so-called *concept descriptions*. Concept and role descriptions are inductively defined with the help of a set of *constructors*, starting with a set N_I of *constants*, a set N_C of *primitive concepts*, and a set N_R of *primitive roles*. The constructors determine the expressive power of the DL. In this paper, we consider concept descriptions built from the constructors shown in Table 1. The resulting DL is denoted by $\mathcal{ELIRO}^1$. Due to the fact that referents of individual concept nodes in SGs are single constants $a \in N_I$, we restrict ourselves to $\mathcal{ELIRO}^1$-concept descriptions in which *each conjunction contains at most one constant.*

The semantics of a concept description C (resp. a role description r) is defined by a FO formula $\Psi_C(x)$ with one free variable (resp. $\Psi_r(x,y)$ with two free variables): see Table 1 for the inductive definition of these formulae. Given an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of the signature $\langle N_C, N_R, N_I\rangle$, the concept description C is interpreted as $C^{\mathcal{I}} := \{\delta \in \Delta \mid \mathcal{I} \models \Psi_C(\delta)\}$.

For example, the concept description

$$D = \mathsf{Female} \sqcap \exists \mathsf{likes}.\mathsf{Male} \sqcap \exists \mathsf{has\text{-}child}.(\mathsf{Student} \sqcap \exists \mathsf{attends}.\mathsf{CScourse})$$

describes all women who like a man and have a child that is a student attending a CScourse. The semantics of D is given by the following FO formula:

$$\begin{aligned}\Psi_D(x_0) = {} & \mathsf{Female}(x_0) \wedge \exists x.(\mathsf{likes}(x_0,x) \wedge \mathsf{Male}(x)) \wedge \\ & \exists y.(\mathsf{has\text{-}child}(x_0,y) \wedge \mathsf{Student}(y) \wedge \exists z.(\mathsf{attends}(y,z) \wedge \mathsf{CScourse}(z))).\end{aligned}$$

In order to obtain a structured representation of the knowledge about the application domain one is interested in the subsumption hierarchy formed by the concept descriptions. Using their FO semantics, *subsumption* between concept descriptions is defined as $C \sqsubseteq D$ iff $\forall x_0.\Psi_C(x_0) \to \Psi_D(x_0)$ is valid.

Rooted Simple Graphs

We are interested in a class of SGs corresponding to $\mathcal{ELIRO}^1$-concept descriptions. On the one hand, we must restrict our attention to *connected* SGs over a support $\mathcal{S} = \langle N_C, N_R, N_I \rangle$ containing only *binary* relation types, because $\mathcal{ELIRO}^1$-roles correspond to binary relations and, as we will see, $\mathcal{ELIRO}^1$-concept descriptions always describe connected structures. Because of the restriction to binary relations, we can dispense with explicit relation nodes: instead we consider directed edges between concept nodes labeled by a relation type.

On the other hand, we must (1) deal with the different semantics of SGs and concept descriptions (closed formulae vs. formulae with one free variable), and (2) introduce conjunctions of types in SGs since conjunctions of primitive concepts may occur in $\mathcal{ELIRO}^1$-concept descriptions. In order to handle (2), we allow for concept nodes labeled by a set of concept types $\{P_1, \ldots, P_n\} \subseteq N_C$, where the empty set corresponds to $\top_C$. Due to (1), we extend the notion of SGs by introducing one distinguished concept node called the *root* of the SG.

Formally, we restrict the attention to *unordered* supports $\langle N_C, N_R, N_I \rangle$ where the orders on N_C and N_R are the identity relations.[2] Given such an unordered support $\langle N_C, N_R, N_I \rangle$ we define a *rooted SG* $\mathcal{G} = (V, E, c_0, \ell)$ over this support as a SG where V is a set of concept nodes, $E \subseteq V \times N_R \times V$ is a set of directed edges labeled by relation types from N_R, c_0 is the root of $\mathcal{G}$, and ℓ labels each $c \in V$ by a set of concept types $\{P_1, \ldots, P_n\} \subseteq N_C$ and a referent from $N_I \cup \{*\}$.

Given an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of $\langle N_C, N_R, N_I \rangle$, the semantics of a rooted SG $\mathcal{G}$ is given by $\{\delta \in \Delta \mid \mathcal{I} \models \Phi(\mathcal{G})(\delta)\}$, where Φ is an extension of the Φ operator from SGs to rooted SGs. To be more precise, the FO formula $\Phi(\mathcal{G})(x_0)$ with one free variable x_0 is obtained from $\mathcal{G}$ as follows. Let $id : V \to (\mathcal{V} \setminus \{x_0\}) \cup N_I$ be a mapping as defined in Section 2. Each concept node $c \in V$ with $type(c) = \{P_1, \ldots, P_n\}$ yields a conjunction $P_1(id(c)) \wedge \ldots \wedge P_n(id(c))$, and each edge $c_1 r c_2 \in E$ yields $r(id(c_1), id(c_2))$. Now, $\Phi(\mathcal{G})(x_0)$ is defined as the conjunction of $x_0 \doteq id(c_0)$ and the formulae corresponding to concept nodes and edges, where all variables except x_0 are existentially quantified.

For example, the rooted SG $\mathcal{G}_1$ with root c_0 depicted in Fig. 3 describes all women that are a daughter of Peter, and have a dear son that likes Peter and is a student attending the CScourse number KR101.

Just as for SGs, subsumption between rooted SGs can be characterized by the existence of a homomorphism. Here, the notion of a homomorphism between SGs w.r.t. a support $\mathcal{S}$ [6] must be adapted to rooted SGs. A *homomorphism* from $\mathcal{H} = (V_H, E_H, d_0, \ell_H)$ to $\mathcal{G} = (V_G, E_G, c_0, \ell_G)$ is a mapping $\varphi : V_H \to V_G$ such that (1) $\varphi(d_0) = c_0$, (2) $type_H(d) \subseteq type_G(\varphi(d))$ and $ref_H(d) \geq_I ref_G(\varphi(d))$ for all $d \in V_H$, and (3) $\varphi(d) r \varphi(d') \in E_G$ for all $d r d' \in E_H$.

The proof of the following theorem in [3] is similar to the proof of soundness and completeness of the characterization of subsumption in [6].

Theorem 1. *Let $\mathcal{G}$ be a rooted SG in normal form and $\mathcal{H}$ a rooted SG. Then $\mathcal{G} \sqsubseteq \mathcal{H}$ iff there exists a homomorphism from $\mathcal{H}$ to $\mathcal{G}$.*

[2] It should be noted that the restriction to unordered supports is without loss of generality since the order relation on N_C can be encoded into the type set labels, and the one on N_R into multiple edges between nodes. Vice versa, the introduction of sets of types is not a real extension since their effect can be simulated by an appropriately extended ordered support (see [3] for details).

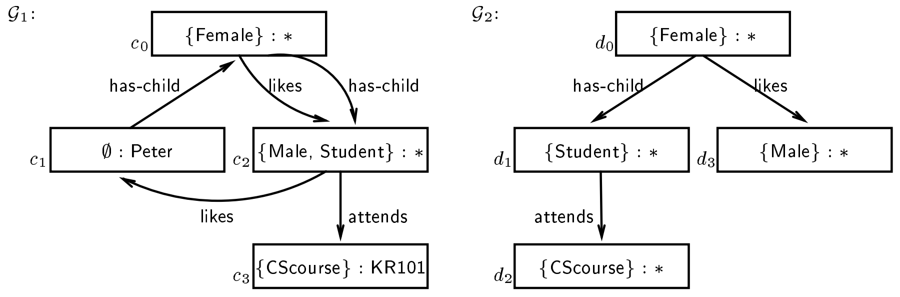

Fig. 3. Two rooted simple graphs.

For example, the rooted SG $\mathcal{G}_2$ with root d_0 depicted in Fig. 3 subsumes $\mathcal{G}_1$ because mapping d_0 onto c_0, d_1 and d_3 onto c_2, and d_2 onto c_3 yields a homomorphism from $\mathcal{G}_2$ to $\mathcal{G}_1$.

Unlike SGs over an arbitrary support, rooted SGs can be transformed (in polynomial time) into equivalent rooted SGs in normal form by identifying all concept nodes $c_1, \ldots, c_n$ having the same referent $a \in N_I$ and defining the type set of the resulting concept node as $\bigcup_{1 \leq i \leq n} \mathit{type}(c_i)$.

For SGs over a support $\mathcal{S}$, subsumption is known to be an NP-complete problem. The known algorithms deciding $g \sqsubseteq h$ w.r.t. $\mathcal{S}$ are based on the characterization of subsumption by homomorphisms, and thus require the subsumee g to be in normal form. In order to obtain a subsumption algorithm for rooted SGs, we must simply adjust the conditions tested for nodes and edges according to the modified conditions on homomorphisms between rooted SGs. Conversely, subsumption of SGs w.r.t. $\mathcal{S}$ can be reduced to subsumption of rooted SGs [3]. This shows that subsumption for rooted SGs is also an NP-complete problem.

In [13], a polynomial-time algorithm is introduced that can decide $g \sqsubseteq \mathbf{t}$ w.r.t. a support $\mathcal{S}$ provided that $\mathbf{t}$ is a tree and g is a SG in normal form. In this context, a SG $\mathbf{t}$ is called a *tree* iff $\mathbf{t}$ contains no cycles of length greater than 2. The notion of a tree can be adapted to rooted SGs $\mathcal{T}$ by viewing $\mathcal{T}$ as an *undirected graph.* A simple modification of the algorithm in [13] yields a polynomial time algorithm deciding $\mathcal{G} \sqsubseteq \mathcal{T}$ for a rooted SG $\mathcal{T}$ that is a tree and a rooted SG $\mathcal{G}$ in normal form [3].

Now, we will show that this algorithm also yields a polynomial-time algorithm for subsumption of $\mathcal{ELIRO}^1$-concept descriptions.

Translating concept descriptions into rooted simple graphs

The main idea underlying the translation is to represent a concept description C as a tree T_C. Intuitively, C is represented by a tree with root c_0 where all atomic concepts and constants occurring in the top-level conjunction of C yield the label of c_0, and each existential restriction $\exists r.C'$ in this conjunction yields an r-successor that is the root of the tree corresponding to C'. For example, the concept description C below yields the tree T_C in Fig. 4:

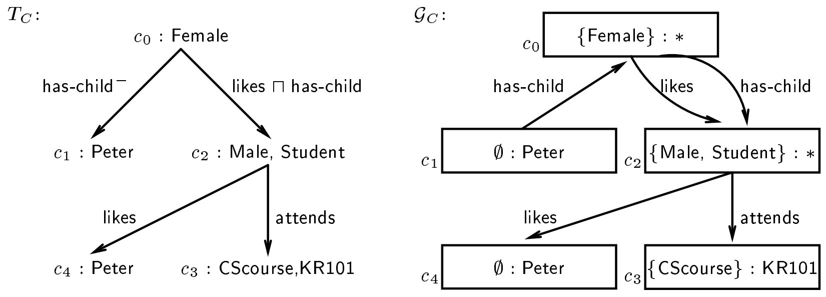

Fig. 4. Translating $\mathcal{ELIRO}^1$-concept descriptions into rooted simple graphs.

$$\begin{aligned} C := {} & \mathsf{Female} \sqcap \exists \mathsf{has\text{-}child}^-.\{\mathsf{Peter}\} \sqcap \exists(\mathsf{likes} \sqcap \mathsf{has\text{-}child}). \\ & \quad (\mathsf{Male} \sqcap \mathsf{Student} \sqcap \exists \mathsf{attends}.(\mathsf{CScourse} \sqcap \{\mathsf{KR101}\}) \sqcap \exists \mathsf{likes}.\{\mathsf{Peter}\}). \end{aligned}$$

Now, we can define the rooted SG $\mathcal{G}_C$ corresponding to C as follows. The nodes in T_C yield the set of concept nodes V of $\mathcal{G}_C$. The label $\ell(c)$ of a concept node $c \in V$ is determined by the label $\ell_T(c)$ of c in T_C, i.e., $type(c)$ is the set of all atomic concepts occurring in $\ell_T(c)$ and, if there is a constant $a \in \ell_T(c)$, then $ref(c) := a$; otherwise $ref(c) := *$. Note that $ref(c)$ is well-defined because we have restricted $\mathcal{ELIRO}^1$-concept descriptions to those containing at most one constant in each conjunction. Finally, the set of edges of $\mathcal{G}_C$ is obtained from the edges in T_C: conjunctions of roles are decomposed ($c(r_1 \sqcap \ldots \sqcap r_n)d$ yields n edges $cr_1d, \ldots, cr_nd$) and inverse roles are redirected (cr^-d yields the edge drc). In our example, we obtain the rooted SG $\mathcal{G}_C$ depicted in Fig. 4, which is a tree.

Using the recursive definition of the tree T_C, it can be shown [3] that C is equivalent to $\mathcal{G}_C$, i.e., $\forall x_0.\Psi_C(x_0) \leftrightarrow \Phi(\mathcal{G}_C)(x_0)$ is a valid FO formula. Conversely, any rooted SG $\mathcal{G}$ that is a tree can be translated into an equivalent concept description $C_\mathcal{G}$ [3]. Thus, there is a 1–1 correspondence between $\mathcal{ELIRO}^1$-concept descriptions and rooted SGs that are trees. Because of this correspondence, we can reduce subsumption in $\mathcal{ELIRO}^1$ to subsumption between rooted SGs, i.e., $C \sqsubseteq D$ iff $\mathcal{G}_C \sqsubseteq \mathcal{G}_D$. Since $\mathcal{G}_C$ is a tree and $\mathcal{G}_D$ can be transformed into normal form (in polynomial time), subsumption for $\mathcal{ELIRO}^1$-concept description is polynomial-time decidable by applying the polynomial-time algorithm mentioned above to the tree $\mathcal{G}_C$ and the normal form of $\mathcal{G}_D$. This yields the following tractability result for $\mathcal{ELIRO}^1$ [3]:

Theorem 2. *Subsumption $C \sqsubseteq D$ of $\mathcal{ELIRO}^1$-concept descriptions can be decided in time polynomial in the size of C and D.*

Strictly speaking, the above argument shows tractability only for concept descriptions where each conjunction contains only one constant. The result can, however, easily be extended to general $\mathcal{ELIRO}^1$-concept descriptions [3].

Extending the tractability result
We will now extend the tractability result from (rooted) SGs that are trees to

(rooted) SGs that can be transformed into trees by "cutting cycles" of length greater than 2. For a given rooted SG $\mathcal{G}$, we can eliminate an (undirected) cycle $c_0, \ldots, c_n$ where $c_0 = c_n$ in $\mathcal{G}$ by applying the *split-operation* on concept nodes as introduced for SGs in [5]. To be more precise, we (1) arbitrarily choose a node $c_i \in \{c_1, \ldots, c_n\}$, (2) introduce a new node c labeled like c_i, and (3) replace all edges between c_{i-1} and c_i by edges between c_{i-1} and c. We then say that the cycle is *cut in* c_i. Obviously, any cyclic SG $\mathcal{G}$ can be transformed into an acyclic SG $\mathcal{G}^*$ by applying this operation a polynomial number of times. In general, however, the resulting SG $\mathcal{G}^*$ need not be equivalent to $\mathcal{G}$.

As an example, consider the rooted SG $\mathcal{G}_1$ in Fig. 3. On the one hand, we can eliminate the cycle c_1, c_0, c_2, c_1 by introducing a new node c_4 with label $(\emptyset, Peter)$ and replacing the edge $c_2\mathsf{likes}c_1$ by $c_2\mathsf{likes}c_4$. The resulting tree coincides with the tree $\mathcal{G}_C$ in Fig. 4, and it is equivalent to $\mathcal{G}_1$ because $\mathcal{G}_1$ is a normal form of $\mathcal{G}_C$. On the other hand, if we introduce a new node c labeled $(\{Female\}, *)$ and replace $c_1\mathsf{has\text{-}child}c_0$ by $c_1\mathsf{has\text{-}child}c$, then the resulting tree is not equivalent to $\mathcal{G}_C$ because the student's mother and Peter's child need no longer to be the same person.

The following proposition introduces a condition on rooted SGs that ensures that rooted SGs satisfying this condition can be transformed into equivalent trees by applying the split operation to individual concept nodes [3].

Proposition 1. *If each cycle of length greater than 2 in the rooted SG $\mathcal{G}$ contains at least one individual node c, then $\mathcal{G}$ can be transformed into an equivalent tree $\mathcal{G}^*$ in time polynomial in the size of $\mathcal{G}$.*

Consequently, $\mathcal{G} \sqsubseteq \mathcal{H}$ can be decided in polynomial time if $\mathcal{G}$ is a rooted SG in normal form and $\mathcal{H}$ satisfies the premise of the proposition. It is easy to see that this tractability result also applies to (non-rooted) SGs over a support.

4 The Loosely Guarded Fragment of Conceptual Graphs

Due to the expressiveness of the CG formalism, all the interesting reasoning problems (such as subsumption and validity) are undecidable for general CGs. We will identify a large class of CGs for which both validity and subsumption are decidable. This fragment, which we will call *loosely guarded fragment of CGs*, will be defined directly by syntactic restrictions on graphs. This allows for an efficient test for guardedness of graphs. The fragment corresponds to the so-called loosely guarded fragment of FO. In [1], the guarded fragment of FO was defined in an attempt to find a generalization of modal logics that still enjoys the nice properties of modal logics (like decidability, finite axiomatizability, etc.). In the same work, decidability of this fragment was shown. In [15], an even larger decidable fragment of FO was introduced, the loosely guarded fragment.

Definition 1. *Let Σ be a set of constant and relation symbols including equality (called the* signature*). The* loosely guarded fragment *$LGF(\Sigma)$ of first-order logic is defined inductively as follows:*

1. *Every atomic formula over Σ belongs to LGF(Σ).*
2. *LGF(Σ) is closed under the Boolean connectives $\neg, \wedge, \vee, \rightarrow$, and $\leftrightarrow$.*
3. *If $\mathbf{x}, \mathbf{y}$ are tuples of variables, if $\beta(\mathbf{x}, \mathbf{y})$ is a formula from LGF(Σ), and if $\alpha_1 \wedge \cdots \wedge \alpha_n$ is a conjunction of atoms, then*

   $$\exists \mathbf{x}.((\alpha_1 \wedge \cdots \wedge \alpha_n) \wedge \beta(\mathbf{x}, \mathbf{y})) \quad \textit{and} \quad \forall \mathbf{x}.((\alpha_1 \wedge \cdots \wedge \alpha_n) \rightarrow \beta(\mathbf{x}, \mathbf{y}))$$

   belong to LGF(Σ), provided that, for every variable x in $\mathbf{x}$ and every variable z in $\mathbf{x}$ or $\mathbf{y}$, there is an atom α_j (the guard*) such that x and z occur in α_j.*

An exact complexity result for the satisfiability problem of the loosely guarded fragment was shown by Grädel [11]. It turned out that the complexity of the satisfiability problem in *LGF*(Σ) depends on the arity of the relation symbols in the signature Σ. In general, the problem is 2-ExpTime-complete. However, if the arity of all relation symbols in Σ is bounded by a constant, then the satisfiability problem for *LGF*(Σ) is "only" ExpTime-complete; in particular, this is the case if Σ is finite.

The definition of the loosely guarded fragment of FO gives rise to the definition of a corresponding fragment of CGs, which we will call the *loosely guarded fragment of CGs.* The restrictions defining this fragment guarantee that all quantifiers in the FO translation of a loosely guarded graph can either be eliminated, or are loosely guarded in the sense of Def. 1. The same must apply to any variable appearing free in a sub-formula of the FO translation of a loosely guarded graph. To state the appropriate restrictions on the CGs, we identify the nodes representing free and bound variables in the contexts of a graph. These will be the *new* and *external* nodes introduced in the following definition.

Definition 2. *Let $G = \langle p, coref \rangle$ be a CG over $\mathcal{S}$. A concept node $c \in C(p)$ contained in a context q of p is called* external *iff it has a coreference link to a strictly dominating concept node. It is called* old *iff it satisfies one of the following conditions:*

– *c is an external or an individual concept node.*
– *c is linked by a coreference link to another old node in the same context q.*

Nodes that are not old are called new.

In the CG of Fig. 2, c_3, c_5, c_6 are external nodes while c_1, c_2, c_4 are new nodes. Note that c_4 is a new node even though it is linked by a chain of coreference links to the old node c_5. This is a desired effect of the definition since coreference links inside one context express equality of concept nodes, while the coreference links from c_4 and c_5 to c_6 are used to express inequality of c_4 and c_5.

Definition 3 (The loosely guarded fragment of conceptual graphs). *A CG $G = \langle p, coref \rangle \in CG(\mathcal{S})$ is called* loosely guarded *iff it satisfies the following:*

1. *If $(c_1, c_2) \in coref$ and the context p_1 of c_1 strictly dominates the context p_2 of c_2, then for each context q such that q lies between p_1 and p_2 (i.e. p_1 strictly dominates q and q strictly dominates p_2) it holds that q is labeled by a simple graph g containing no new nodes.*

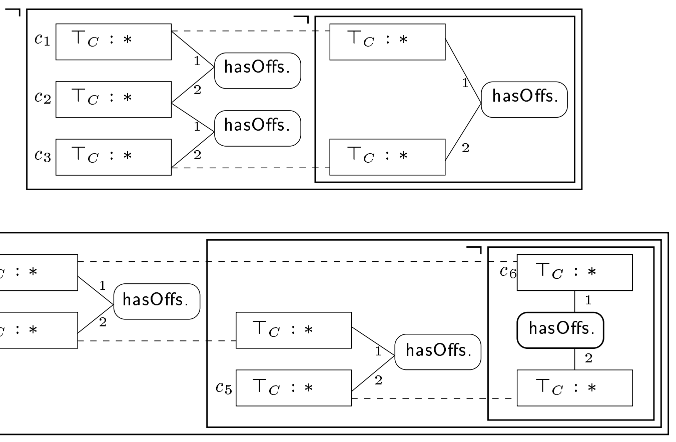

Fig. 5. Two graphs that are not loosely guarded.

2. *For each simple graph $g = \langle C, R, E, \ell \rangle$ labeling a context of G, either g contains no new nodes, or g satisfies the following: if $C = \{c\}$, then type$(c) \neq \top_C$ or there is an $r \in R$ such that $(c, r) \in E$; if $|C| > 1$, then for each pair of distinct nodes $c, d \in C$ such that c is new and d is not an individual concept node, there is an $r \in R$ satisfying $\{(c, r), (d, r)\} \subseteq E$.*

With lgCG$(\mathcal{S})$ we denote the set of all loosely guarded CGs over the support $\mathcal{S}$.

An example of a lgCG is the graph in Fig. 2. In fact, Property 1 is obviously satisfied since no coreference link crosses more than one context. Property 2 is satisfied as well; for example, in the context p_2, the new node c_4 shares a relation node with both c_3 and c_5. The nodes c_2 and c_4 need not be adjacent to the same relation node since both are old nodes.

CGs that violate one of the properties required by Def. 3 need not be equivalent to a loosely guarded FO formula. For example, transitivity of a binary relation symbol is an assertion that cannot be expressed by a loosely guarded formula [11]. Figure 5 shows two CGs that assert transitivity of the binary relation hasOffspring. The upper graph is not loosely guarded because it violates Property 2: c_1 and c_3 are both new nodes, but they are not adjacent to the same relation node. The lower graph is not loosely guarded because it violates Property 1: c_4 and c_6 are linked by a coreference link that spans a context containing the new node c_5.

Note that, even though the definition of lgCGs may look quite complex at first sight, it is a purely syntactic definition using easily testable properties of graphs. Indeed, it is easy to show that membership of a given CG over $\mathcal{S}$ in *lgCG*$(\mathcal{S})$ can be tested in polynomial time [2]. The name "loosely guarded fragment of CGs" is justified by the main theorem of this section:

Theorem 3. *Let $\mathcal{S} = \langle N_C, N_R, N_I \rangle$ be a support and let $\Sigma_{\mathcal{S}}$ be the corresponding FO signature $\Sigma_{\mathcal{S}} = N_C \cup N_R \cup N_I$.*

1. *For each $G \in lgCG(\mathcal{S})$ there exists a formula $\varphi_G \in LGF(\Sigma_{\mathcal{S}})$ such that φ_G is equivalent to $\Phi(G)$. In addition, φ_G is computable from G in polynomial time.*
2. *For each closed formula $\varphi \in LGF(\Sigma_{\mathcal{S}})$ there is a graph $G_\varphi \in lgCG(\mathcal{S})$ such that $\Phi(G_\varphi)$ is equivalent to φ. In addition, G_φ is computable from φ in polynomial time.*

A complete proof of this theorem can be found in [2]. Here, we will illustrate the main idea underlying the proof of the first part, using the example in Fig. 2. The transformation of $\Phi(G)$ into a loosely guarded formula works inductively over the structure of G. Hence, we start with the innermost context p_3 of G. The formula $\varphi_3(x_4, x_5) := \Phi(p_3) = \exists x_6.(x_6 \doteq x_6 \wedge x_6 \doteq x_4 \wedge x_6 \doteq x_5)$ is loosely guarded, and it is equivalent to the simpler loosely guarded formula $\varphi_3' = x_4 \doteq x_5$. The formula for the context p_2,

$$\varphi_2(x_1, x_2) := \Phi(p_2) = \exists x_3 x_4 x_5.(x_3 \doteq x_1 \wedge x_5 \doteq x_2 \wedge \mathsf{Human}(x_4) \wedge \mathsf{hasChild}(x_4, x_3) \wedge \mathsf{likes}(x_4, x_5) \wedge \neg\varphi_3'(x_4, x_5)),$$

is not loosely guarded. In order to obtain a loosely guarded formula, we eliminate the identifiers of the old nodes (in this case x_3, x_5) together with their quantifiers, using the fact that φ_2 contains the conjuncts $x_3 \doteq x_1$ and $x_5 \doteq x_2$. Re-ordering the conjuncts yields the formula

$$\varphi_2'(x_1, x_2) = \exists x_4.(\mathsf{hasChild}(x_4, x_1) \wedge \mathsf{likes}(x_4, x_2) \wedge \mathsf{Human}(x_4) \wedge x_4 \not\doteq x_2),$$

which is loosely guarded and equivalent to φ_2. The necessary guards are given by the first two conjuncts, which correspond to the relation nodes adjacent to c_3, c_4, c_5. The existence of such nodes in a loosely guarded graph is guaranteed by Property 2 of Def. 3.

Since the context p_1 does not contain old nodes, the next steps (in which we also treat the two negation signs) directly yields the loosely guarded equivalent φ_G of $\Phi(G) = \Phi(p)$:

$$\varphi_G := \neg\exists x_1 x_2.(\mathsf{hasChild}(x_2, x_1) \wedge \mathsf{Human}(x_1) \wedge \mathsf{Human}(x_2) \wedge \neg\varphi_2'(x_1, x_2)).$$

Summing up, the techniques used to transform $\Phi(p)$ into its loosely guarded equivalent φ_G are: (1) Elimination of identifiers and the corresponding quantifiers for old nodes; and (2) using Property 2 of Def. 3 to find the appropriate guards for the remaining quantified variables. As has already been pointed out, Property 1 of Def. 3 is necessary to ensure that no free variable of a sub-formula escapes the guards (see Fig. 5).

The following theorem is an immediate consequence of part 1 of Theorem 3 and the known complexity results for the loosely guarded fragment of FO.

Theorem 4. *Let $\mathcal{S}$ be a finite support. Then subsumption and validity of loosely guarded CGs over $\mathcal{S}$ is decidable in deterministic exponential time.*

Because of part 2 of Theorem 3, the ExpTime-hardness result for $LGF(\Sigma_{\mathcal{S}})$ also transfers to $lgCG(\mathcal{S})$.

5 Conclusion

Although the characterization of the loosely guarded fragment of conceptual graphs may appear to be a bit complex, it can easily be checked whether a CG belongs to this fragment.It should also be easy to support the knowledge engineer in designing CGs belonging to this fragment by showing external and new nodes in different colors, and by pointing out new nodes that are not yet guarded. Another interesting point is that there are theorem-provers that are complete for FO, and behave as a decision procedure (i.e., always terminate) for the loosely guarded fragment [8]. If such a prover is used to prove validity of general CGs, then one automatically has a decision procedure if the CGs are loosely guarded.

References

1. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *J. of Philosophical Logic*, 27(3):217–274, 1998.
2. F. Baader, R. Molitor, and S. Tobies. The Guarded Fragment of Conceptual Graphs. LTCS-Report 98-10, available at http://www-lti.informatik.rwth-aachen.de/ Forschung/Papers.html.
3. F. Baader, R. Molitor, and S. Tobies. On the Relationship between Descripion Logics and Conceptual Graphs. LTCS-Report 98-11, available at http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.
4. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.
5. M. Chein and M. L. Mugnier. Conceptual graphs: fundamental notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.
6. M. Chein, M. L. Mugnier, and G. Simonet. Nested graphs: a graph-based knowledge representation model with FOL semantics. In *Proc. KR'98*, 1998.
7. P. Coupey and C. Faron. Towards correspondences between conceptual graphs and description logics. In *Proc. ICCS'98*, LNCS 1453, 1998.
8. H. de Nivelle. A resolution decision procedure for the guarded fragment. In *Proc. CADE-15*, LNCS 1421, 1998.
9. F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134(1):1–58, 1997.
10. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In *Foundation of Knowledge Representation*, CSLI-Publications, 1996.
11. E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 1999. To appear.
12. G. Kerdiles and E. Salvat. A sound and complete CG proof procedure combining projections with analytic tableaux. In *Proc. ICCS'97*, LNCS 1257, 1997.
13. M. L. Mugnier and M. Chein. Polynomial algorithms for projection and matching. In *Proc. 7th Workshop on Conceptual Structures, 1992*, LNCS 754, 1993.
14. John F. Sowa. *Conceptual Structures.* Addison-Wesley, 1984.
15. J. van Benthem. Dynamic Bits and Pieces. Technical Report LP-1997-01, ILLC, University of Amsterdam, The Netherlands, 1997.
16. M. Wermelinger. Conceptual graphs and first-order logic. In *Proc. ICCS'95*, LNCS 954, 1995.