# Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics

**M. Chein**
LIRMM (CNRS and Univ. Montpellier II)
161, rue Ada, F-34392 Montpellier Cedex 5
email: *name*@lirmm.fr

**M.L. Mugnier**
LIRMM

**G. Simonet**
LIRMM

## Abstract

We present a graph-based KR model issued from Sowa's conceptual graphs but studied and developed with a specific approach. Formal objects are kinds of labelled graphs, which may be simple graphs or nested graphs. The fundamental notion for doing reasonings, called projection (or subsumption), is a kind of labelled graph morphism. Thus, we propose a *graphical* KR model, where "graphical " is used in the sense of [Sch91], i.e. a model that "uses graph-theoretic notions in an essential and nontrivial way". Indeed, morphism, which is the fundamental notion for any structure, is at the core of our theory. We define two first order logic semantics, which correspond to different intuitive semantics, and prove in both cases that projection is sound and complete with respect to deduction.

*This paper is almost identical to the paper appeared in the KR'98 proceedings. It provides minor corrections.*

## 1 Introduction

Different kinds of *labelled graphs* have long been used for representing knowledge. In artificial intelligence, they have been investigated under the name of *semantic networks* (see for instance [Leh92] for an overview of recent research in this domain). [Sow76] proposed a model of this family, named *conceptual graphs* (CGs), which was developed in [Sow84]. The formal aspects of this model have two mathematical bases: logic and graph theory. Emphasizing one or the other leads to different research lines.

The former, a continuation of Peirce's work on existential graphs [Rob92], investigates CGs as a *diagrammatic* system of logic (for instance [Sow94] [vdB94]). The later develops CG as a *graphical* knowledge representation model, where "graphical" is used in the sense of [Sch91], i.e. a model that "uses graph-theoretic notions in an essential and nontrivial way" (for instance [LE91] [CM92]). Our work is concerned with this second viewpoint.

The basic notion is that of a *simple conceptual graph* (SG). A SG is a labelled bipartite graph where nodes of one class represent entities, nodes of the other class represent relations between these entities, and the labels (which are partially ordered) represent types and referents. The fundamental operation, called "projection", is a *morphism* between SGs. Projection induces a structural *subsumption* relation. SGs are provided with a semantics in first order logic (FOL), classically called $\Phi$. Projection has been proven to be sound with respect to deduction [Sow84]. We proved that it is also complete, when SGs are in a "normal" form [CM92].

In this paper, we provide another FOL semantics, for which completeness holds without any restriction. We also present *nested conceptual graphs* (NGs), which are generalizations of SGs: in NGs, concept nodes may express complex information represented by NGs. We study NGs from a graphical viewpoint. More specifically:

- a NG is a combinatorial object recursively defined from SGs;
- the main notion, projection, is a morphism between NGs, from which a subsumption relation is defined;
- the NG model is provided with two logical semantics in FOL, which extend those for SGs. Soundness and completeness results are given for both cases.

NGs come from several applications in which the SG model proved to be inconvenient for expressing the knowledge involved. Let us give one example of an application we are currently involved in. In this application, NGs are used to represent and simulate the required behavior of human agents in case of emergency situations, for instance a fire in a chemical plant [BBV97]. This behavior is described by NGs which represent situations, pre-conditions and post-conditions of actions to be performed, each of these elements being themselves described by NGs. Knowledge is structured by hierarchical levels and reasonings must respect these levels.

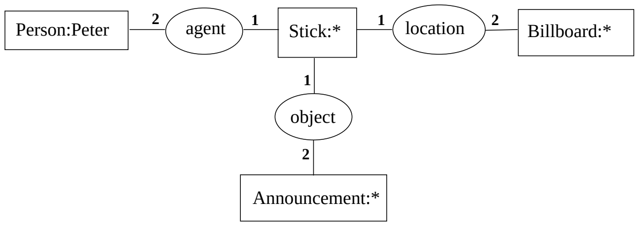
The paper is organized as follows. In section 2, basic definitions and results on SGs are provided. In section 3, co-reference links are added to SGs and we propose a new logical semantics, which is compared to $\Phi$. We prove that projection is sound and complete w.r.t. this semantics. The model is then extended to NGs with co-reference links (section 4). Two logical semantics for NGs are developed, for which projection is proven to be sound and complete (sections 5 and 6). Section 7 is devoted to related works.

## 2 Basic Notions on Conceptual Graphs

In this section, we present main features about SGs. For a more detailed presentation see [Sow84] and [CM92]. Let us first specify some notations. Any partial order or preorder (reflexive and transitive but not necessarily antisymmetric relation) is denoted by $\leq$ or $\geq$. $\models$ is the logical consequence symbol in FOL.

Basic ontological knowledge is encoded in a structure we call a **support**. We consider here a simplified version of a support with 3 components, say $S = (T_C, T_R, I)$. $T_C$ and $T_R$ are type sets, respectively a set of concept types and a set of relation types, partially ordered by an A-Kind-Of relation. Relation types may have any arity greater or equal to 1, and two comparable relation types must have the same arity. $I$ is a set of individual markers. All supports also possess a generic marker denoted by $*$. The following partial order is defined on the set of markers $I \cup \{*\}$: $*$ is the greatest element and elements of $I$ are pairwise noncomparable.

Asserted facts are encoded by SGs. SGs are also used for representing ontological knowledge (such as type definitions) or other kinds of knowledge such as rules or prototypes, but we will not cover these aspects here. A **SG** related to a support $S$, is a labelled bipartite graph $G = (R, C, E, l)$. $R$ and $C$ are the node sets, respectively relation node set and concept node set. $E$ is the set of edges. Edges incident on a relation node are totally ordered, they are numbered from 1 to the degree of the relation node. The *ith* neighbor of a relation $r$ is denoted by $G_i(r)$. Each node has a label given by the mapping $l$. A concept node $c$ is labelled by a couple $(type(c), ref(c))$, where $type(c)$, the type of $c$, belongs to $T_C$, and $ref(c)$, the referent of $c$, either belongs to $I$ — then $c$ is said to be an individual concept node — or is the generic marker $*$ — then $c$ is said to be a generic concept node. A relation node $r$ is labelled by a relation type, $type(r)$, and the number of edges incident on $r$ is equal to the arity of $type(r)$. Figure 1 gives an example of a SG.




This graph may represent the information
"Peter sticks an announcement on a billboard".

Figure 1: A simple graph

SGs are given a semantics in FOL, denoted by $\boldsymbol{\Phi}$ [Sow84]. Given a support $S$, a constant is assigned to each individual marker and an $n$-adic (resp. unary) predicate is assigned to each $n$-adic relation (resp. concept) type. For simplicity, we consider that each constant or predicate has the same name as the associated element of the support. To $S$ is assigned a set of formulas, $\Phi(S)$, which corresponds to the interpretation of the partial orderings of $T_R$ and $T_C$. For all types $t_1$ and $t_2$ such that $t_1 \geq t_2$, one has the formula $\forall x_1 ... x_p (t_2(x_1, \ ..., x_p) \rightarrow t_1(x_1, \ ..., x_p))$, where $p = 1$ for concept types, and $p$ is otherwise the arity of the relation. $\Phi$ maps any graph $G$ on $S$ into a formula $\Phi(G)$ in the following way. First assign to each concept node $c$ a term which is a variable if $c$ is generic, and otherwise the constant corresponding to $ref(c)$. Two distinct generic nodes receive distinct variables. Then assign an atom to each node of $G$: the atom $t_c(e)$ to a concept $c$ where $t_c$ stands for the type of $c$ and $e$ is the term associated with $c$; the atom $t_r(e_1, \ ..., \ e_p)$ to a relation $r$ where $t_r$ stands for the type of $r$ and $e_i$ is the term associated with the *ith* neighbor of $r$. Finally, the conjunction of these atoms is made and the obtained formula is existentially closed. E.g. let us consider the graph $G$ of Figure 1. We call $c_1$, $c_2$, $c_3$, $c_4$ the concept nodes of respective types *Person*, *Stick*, *Announcement*, *Billboard* and assign the variable $y_i$ to the generic concept node $c_i$ for $i = 2, 3, 4$. The formula associated with $G$ is

$\Phi(G) = \exists y_2 y_3 y_4 (Person(Peter) \wedge Stick(y_2) \wedge Announcement(y_3) \wedge Billboard(y_4) \wedge agent(y_2, Peter) \wedge object(y_2, y_3) \wedge location(y_2, y_4))$.

Several graph operations are defined on SGs for doing reasonings. Our work is based on the projection operation [Sow84]. A **projection** from a SG $G = (R_G, C_G, E_G, l_G)$ to a SG $H = (R_H, C_H, E_H, l_H)$ is a mapping $\Pi$ from $R_G$ to $R_H$ and from $C_G$ to $C_H$ which

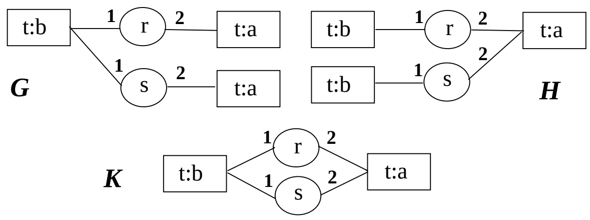
1. preserves adjacency and order on edges: $\forall\ rc \in E_G\ \Pi(r)\Pi(c) \in E_H$ and if $c = G_i(r)$ then $\Pi(c) = H_i(\Pi(r))$
2. may decrease labels: $\forall x \in R_G \cup C_G$ $l_G(x) \geq l_H(\Pi(x))$.

The partial order on relation labels is that of $T_R$. The partial order on concept labels is the product of the partial order on $T_C$ and the partial order on $I \cup \{*\}$. Note that projection is generally not a one-to-one mapping ($\Pi(G)$ is generally not isomorphic to $G$). See for instance Figure 2: $G$ and $H$ project onto $K$.

Projection can also be defined as a graph (homo-) morphism. Let us recall that a graph morphism from a graph $G$ to a graph $H$ is a mapping $f$ from $G$ node set to $H$ node set which preserves adjacency, i.e. if $xy$ is an edge of $G$ then $f(x)f(y)$ is an edge of $H$. When $G$ and $H$ are labelled, morphism may satisfy additional constraints on labels. Projection is a labelled bipartite graph morphism.

Projection defines a **subsumption** relation on SGs: $G \geq H$ ($G$ subsumes $H$) if there is a projection from $G$ to $H$. This relation is a preorder, but not an order. Subsumption checking is an NP-complete problem [CM92]. It becomes polynomial when $G$ is a tree [Mug95] (or more generally when $G$ has a bounded treewidth). Also note that there is a "strong" equivalence between the subsumption checking problem and the constraint satisfaction problem [Dec92], allowing direct transfer of efficient algorithms from one problem to the other [MC96].

Projection is sound and complete with respect to FOL, i.e.: given two SGs $G$ and $H$ defined on $S$, if $G$ subsumes $H$ then $\Phi(S), \Phi(H) \models \Phi(G)$ (soundness [Sow84]); conversely, if $H$ is in normal form and $\Phi(S), \Phi(H) \models \Phi(G)$ then $G$ subsumes $H$ (completeness [CM92]). A SG is said to be in **normal form** if each individual marker appears at most once in concept node labels. A SG $G$ can be put into normal form by merging all concept nodes having the same individual referent, provided that they have the same type.




$\Phi(G) = t(b) \wedge t(a) \wedge t(a) \wedge r(b,\ a) \wedge s(b,\ a)$.
$\Phi(H) = t(b) \wedge t(b) \wedge t(a) \wedge r(b,\ a) \wedge s(b,\ a)$.
$\Phi(G)$ and $\Phi(H)$ are equivalent, while $G$ and $H$ are noncomparable by projection. $K$ is the normal form of $G$ and $H$. $G$ and $H$ project into $K$.

Figure 2: Normal Form

This simple transformation produces a SG – called *the normal form* of $G$ – whose logical semantics is trivially equivalent to that of $G$. See Figure 2.

Let us add that there are strong links between conjunctive query containment in databases and projection. Indeed, it is straightforward to represent any positive and non recursive conjunctive query $q$ by a simple graph $G_q$ in such a way that, given two queries $q$ and $q'$, $q'$ contains $q$ if and only if there is a projection from $G_{q'}$ to $G_q$. In this transformation, the obtained support for SGs is very simple, since all types are noncomparable. More precisely, let us consider a positive and non recursive conjunctive query $q = ans(u) \leftarrow r_1(u_1),\ ...\ r_n(u_n),\ n \geq 1$, where, given $R$ a set of relation names and *dom* a countably infinite set of constants, $r_1\ ...\ r_n$ belong to $R$, $ans$ is not in $R$, $u$ and $u_1\ ...\ u_n$ are tuples of terms (variables or constants of *dom*), and each variable of $u$ occurs at least once in $u_1\ ...\ u_n$. We follow here the notations of [AHV95]. Now, let $S = (T_C,\ T_R,\ I)$ be a support built as follows. $T_C$ is restricted to a single type. $T_R$ is equal to $R \cup Ans$, where $Ans = \{ans_k |\ k \geq 1\}$ and $k$ stands for the arity of $ans$. $I$ is equal to *dom*. Elements of $T_R$ are noncomparable. Then a SG (in normal form) $G_q = (R_q, C_q, E_q, l_q)$ can be assigned to $q$ as follows. $R_q$ is a set of (n+1) nodes labelled by $r_1$, ..., $r_n$, $ans_k$. $C_q$ is in bijection with the set of terms occurring in $u$, $u_1$, ..., $u_n$. The node of $C_q$ assigned to a term $e$ is generic if $e$ is a variable and otherwise individual with the referent $e$. And if $e$ is the $ith$ argument of $r_j$ (or $ans$), then the concept node assigned to $e$ is the $ith$ neighbor of the relation node assigned to $r_j$ (or $ans$). Then, one can immediately check that, given two queries, $q$ and $q'$, there is a homomorphism from $q$ to $q'$ if and only if there is a projection from $G_q$ to $G_{q'}$.

Moreover, it can be proven that the homomorphism theorem ([AHV95], 6.2.3) is equivalent to the soundness and completeness theorem for projection between SGs (in normal form) [CM92], the minimization theorem ([AHV95], 6.2.6) is equivalent to the theorem about irredundant SGs [CM92], and the complexity theorem for query decision problems ([AHV95], 6.2.10) is equivalent to the complexity theorem for SGs decision problems [CM92]. See [SCM98] for more details about these equivalences.

## 3 Co-reference links and another FOL semantics

According to the semantics $\Phi$, merging several individual concept nodes having the same referent (i.e. representing the same entity) does not change the meaning of the graph. However, in some applications, it may be useful to represent the same entity with different concept nodes corresponding to different aspects or viewpoints on this entity. For instance, suppose $G$ represents the organization of a company. $G$ contains three subgraphs, which respectively represent the structures of the staff, the joint production committee and the football team. Some persons appear in more than one of these three structures, but we do not wish to merge the concept nodes representing the same person, which would have the effect of mixing and destroying the three structures. When a specified individual is represented by several nodes, the individual marker suffices for expressing that these nodes represent the same entity. But in the case of an unspecified entity represented by several generic nodes (e.g. a person identified by his/her different functions in the company) an additional structure, called a co-reference link, is needed.

A SG with co-reference links (in short $\mathbf{SG^{ref}}$) is a graph $G = (R, C, E, l, \mathit{co\text{-}ref})$, where **co-ref** is an equivalence relation on the set of $G$ generic concept nodes (the *co-reference* relation).The intuitive semantics of $\mathit{co\text{-}ref}$ is "represents the same entity as". Any basic SG $G = (R, C, E, l)$ may be written in the form $G = (R, C, E, l, \mathit{co\text{-}ref})$ where $\mathit{co\text{-}ref}$ is restricted to the identity relation. The $\mathit{co\text{-}ref}$ relation is naturally extended to an equivalence relation **co-ident** on the set $C$ of all concept nodes of $G$ (the *co-identity* relation). Every equivalence class for the co-identity relation is a set of concept nodes representing the same entity which are either co-referent generic nodes (in that case they are explicitly linked by a co-reference link in a graphical representation of $G$) or individual nodes having the same referent: $\forall c, c' \in C$,
$\mathit{co\text{-}ident}(c, c')$ iff ($\mathit{co\text{-}ref}(c, c')$ or $\mathit{ref}(c) = \mathit{ref}(c') \in I$)

The definitions and results on SGs are modified as follows by the introduction of co-reference links. A **projection** from a SG$^{ref}$ $G = (R_G, C_G, E_G, l_G, \mathit{co\text{-}ref}_G)$ to a SG$^{ref}$ $H = (R_H, C_H, E_H, l_H, \mathit{co\text{-}ref}_H)$ is a projection $\Pi$ from the SG $(R_G, C_G, E_G, l_G)$ to the SG $(R_H, C_H, E_H, l_H)$ that preserves co-identity, i.e. $\forall\ c, c' \in C_G$ if $\mathit{co\text{-}ref}_G(c, c')$ then $\mathit{co\text{-}ident}_H(\Pi(c), \Pi(c'))$. $\Phi$ may be extended to SG$^{ref}$s by associating the same variable with co-referent generic concept nodes. A SG$^{ref}$ $G$ is in normal form iff the co-identity relation on $C_G$ is the identity one (i.e. iff it contains no co-reference links and each individual marker appears at most once in concept node labels). A SG$^{ref}$ is put into normal form by merging its co-identical nodes (provided tha they have the same type). In order to translate the whole information encoded in a graph, we introduce another FOL semantics, called $\mathbf{\Psi}$. In this semantics, two terms are assigned to a concept node $c$. The first term ($e_{\overline{c}}$) represents the co-identity class $\overline{c}$ of $c$ as in $\Phi$ (it is a constant if $c$ is an individual node, otherwise it is the variable associated with $\overline{c}$). The second term ($e_c$) is a variable, which represents the node itself. All these variables are pairwise distinct. $\Psi(S)$ is defined as $\Phi(S)$, except that predicates associated with concept types become binary. $\Psi(G)$ is defined as follows. Assign the atom $t(e_{\overline{c}}, e_c)$ to each concept node $c$ and the atom $t_r(e_{c_1}, \ldots, e_{c_p})$ to each relation node $r$ of $G$, where $c_i$ is the $ith$ neighbor of $r$. $\Psi(G)$ is the existential closure of the conjunction of these atoms. E.g. the formula associated with the graph $G$ of Figure 2 is $\Psi(G) = \exists x_1 x_2 x_3 (t(b, x_1) \wedge t(a, x_2) \wedge t(a, x_3) \wedge r(x_1, x_2) \wedge s(x_1, x_3))$. With the graph $G$ of Figure 1, we assign the variable $y_i$ to the co-reference class restricted to the generic concept node $c_i$ for $i = 2, 3, 4$ (as $G$ does not contain any co-reference link) and the variable $x_i$ to the concept node $c_i$ for $i = 1, 2, 3, 4$. We obtain
$\Psi(G) = \exists y_2 \ \ldots \ y_4 x_1 \ \ldots \ x_4 (\mathit{Person}(\mathit{Peter}, x_1) \wedge \mathit{Stick}(y_2, x_2) \wedge \mathit{Announcement}(y_3, x_3) \wedge \mathit{Billboard}(y_4, x_4) \wedge \mathit{agent}(x_2, x_1) \wedge \mathit{object}(x_2, x_3) \wedge \mathit{location}(x_2, x_4))$.

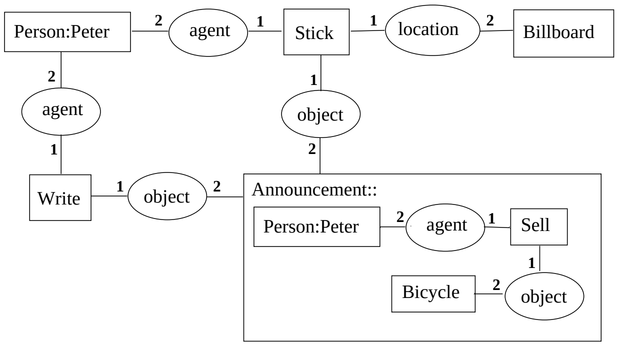
Projection is sound and complete with respect to this semantics without any restriction. E.g. consider the graphs of Figure 2. $\Psi(G)$ and $\Psi(H)$ are noncomparable by deduction and are both deducible from $\Psi(K)$.

**Theorem 1** *Let $G$ and $H$ be two SG$^{ref}$s.*
*$G$ subsumes $H$ iff $\Psi(S), \Psi(H) \models \Psi(G)$.*

**Proof**: the proof is similar to that with the semantics $\Phi$. However we give it here since the complete proof with $\Phi$ is available in French only [MC96]. For any formula $F$, let $\mathcal{C}(F)$ denote the clausal form of $F$. $\mathcal{C}(\Psi(H))$ is the set of atomic clauses obtained from $\Psi(H)$ atoms by substituting Skolem constants to the variables of $\Psi(H)$. Let $\rho$ be this substitution. $\mathcal{C}(\neg\Psi(G))$ contains a unique clause whose literals are the negations of $\Psi(G)$ atoms. The Herbrand Universe $U_H$ of the formula $\Psi(S) \wedge \Psi(H) \wedge \neg\Psi(G)$ is the set of constants appearing in $\mathcal{C}(\Psi(H))$ or in $\Psi(G)$. A *s-substitution* from $G$ to $H$ ($s$ standing for *subsuming*) is a substitution $\sigma$ of the variables of $\Psi(G)$ by $U_H$ constants such that for any atom $t(e_1, ..., e_n)$ of $\Psi(G)$, there is $t' \leq t$ such that $\sigma(t'(e_1, ..., e_n))$ is an atom of $\mathcal{C}(\Psi(H))$. Theorem 1 immediately follows from Lemmas 1 and 2. $\square$

**Lemma 1**: $\Psi(S), \Psi(H) \vDash \Psi(G)$ *iff there is a s-substitution from $G$ to $H$.*
**Proof**: let $\mathcal{C} = \mathcal{C}(\Psi(S) \wedge \Psi(H) \wedge \neg\Psi(G))$. $\Psi(S), \Psi(H) \vDash \Psi(G)$ iff $\mathcal{C}$ is unsatisfiable. If there is a s-substitution from $G$ to $H$ then the empty clause can be obtained from $\mathcal{C}$ by the resolution method, so $\mathcal{C}$ is unsatisfiable. Conversely, let us suppose that $\mathcal{C}$ is unsatisfiable. Let $v$ be the Herbrand interpretation defined by: for any predicate $t$ of arity $n$ and for all constants $a_1$, ..., $a_n$ of $U_H$, $v(t)(a_1, ..., a_n)$ is true iff there is $t' \leq t$ such that $t'(a_1, ..., a_n)$ is an atom of $\mathcal{C}(\Psi(H))$. $v$ is a model of $\mathcal{C}(\Psi(S) \wedge \Psi(H))$. Then $v$ is not a model of $\mathcal{C}(\neg\Psi(G))$, which provides a s-substitution from $G$ to $H$. $\square$

**Lemma 2**: *$G$ subsumes $H$ iff there is a s-substitution from $G$ to $H$.*
**Proof**: let us suppose that $G$ subsumes $H$. Let $\Pi$ be a projection from $G$ to $H$. For any variable $x$ of $\Psi(G)$, let $c$ be the concept node of $G$ such that $x = e_c$ (resp. $e_{\overline{c}}$) and let $\sigma(x)$ be the $U_H$ constant $\rho(e_{\Pi(c)})$ (resp. $\rho(e_{\overline{\Pi(c)}})$). For any atom $t(e_1, ..., e_n)$ of $\Psi(G)$, let $s$ be the node of $G$ associated with this atom. The atom of $\mathcal{C}(\Psi(H))$ associated with $\Pi(s)$ is in the form $\sigma(t'(e_1, ..., e_n))$, with $t' \leq t$. $\sigma$ is a s-substitution from $G$ to $H$.
Conversely, let us suppose that there is a s-substitution $\sigma$ from $G$ to $H$. For any node $s$ of $G$, let $t(e_1, ..., e_n)$ be the atom of $\Psi(G)$ associated with $s$ and let $\Pi(s)$ be a node of $H$ associated with an atom in the form $\sigma(t'(e_1, ..., e_n))$, with $t' \leq t$. $\Pi$ fullfills condition 2 of the projection definition. As for condition 1, if $c = G_i(r)$ then $\sigma(e_c) = \rho(e_{\Pi(c)}) = \rho(e_{H_i(\Pi(r))})$, then $\Pi(c) = H_i(\Pi(r))$. Note that with the semantics $\Phi$, we only have $\sigma(e_{\overline{c}}) = \rho(e_{\overline{\Pi(c)}}) = \rho(e_{\overline{H_i(\Pi(r))}})$, then $\overline{\Pi(c)} = \overline{H_i(\Pi(r))}$ and we need the normality condition to ensure that $\Pi(c) = H_i(\Pi(r))$. $\Pi$ is a projection from $G$ to $H$. $\square$




(* and ** are omitted)

Figure 3: A nested graph

# 4 Nested Graphs

In this section, we define nested graphs (without co-reference links first) and projection between nested graphs.

Let us consider the graph of Figure 1. Suppose one wants to add two pieces of information about the announcement. First, "Peter wrote this announcement", second "this announcement says that Peter is selling a bicycle". Let us assume that these pieces of information are considered to be of different levels, the first one being an *external* information about the announcement, and the second one an *internal* information. If one wants to keep this difference in the representation, then in the former the announcement node is a *black box* which is joined to the Peter node via a node representing a writing, and the latter information can be put inside the announcement node. This last information can be obtained by zooming on the announcement node which in this case becomes a *glass box* (Figure 3). It can also be said that the information nested in a node is pertinent within the context represented by this node. This information is called a *(partial) description* of the node.

Note that a description is not a type definition. First, a description applies to a specific concept node, whereas a graph defining a type applies to all concept nodes of this type. Second, a type definition provides a characterization of a type, i.e. it describes necessary and sufficient conditions for any individual to belong to the type, whereas a description is only a partial information about an individual. Two co-identical nodes may have different descriptions, even in the same context.

**NGs** are defined by structural induction as follows.

1. A *basic NG* is obtained from a SG by adding to the label of each concept node $c$, a third field, denoted

by $Desc(c)$, equal to $**$ ($**$ can be considered as the empty description, and a trivial bijection exists between basic NGs and SGs);

2. Let $G$ be a basic NG, let $c_1,\ c_2,\ ...,\ c_k$ be concept nodes of $G$, and let $G_1,\ G_2,\ ...,\ G_k$ be NGs. The graph obtained by substituting $G_i$ to the description $**$ of $c_i$ for $i = 1,\ ...,\ k$ is a NG.

It is important to note (for following definitions of $A(G)$ and $UC_G$) that if a SG or a NG $H$ is used several times in the construction of a NG $G$, we consider that several copies of $H$ (and not several times the graph $H$ itself) are used in the construction of $G$.

Any NG $G$ has an associated syntactic tree $\mathbf{A(G)}$ whose nodes are SGs appearing in the construction of $G$. More precisely, $A(G)$ is a rooted tree which can be defined by structural induction as follows, with $R(G)$ denoting the root of $A(G)$: if $G$ is the basic NG obtained from the SG $G'$ then $A(G)$ is restricted to the node $G'$; if $G$ is the NG obtained from the basic NG $H$, concept nodes $c_1,\ c_2,\ ...,\ c_k$, and NGs $H_1,\ ...,\ H_k$, then $A(G)$ is built from $A(H_1),\ ...,\ A(H_k)$ by adding $R(H)$ as the root and edges $(R(H), c_i)R(H_i)$ leading from the node $c_i$ of $H$ to the root of $A(H_i)$. Note that for any NG $G$, $(J, c)K$ is an edge of $A(G)$ iff $J$ and $K$ are nodes of $A(G)$, $c$ is a concept node of $J$ and $K$ is the root of the tree associated with the description graph of $c$ (i.e. $K = R(Desc(c))$).

A **NG** can be denoted by $G = (R, C, E, l)$ where $R$, $C$ and $E$ are respectively relation, concept, and edge sets of the SG $R(G)$[1], $l$ is a labelling function of $R$ and $C$ such that $\forall r \in R,\ l(r) \in T_R$ and $\forall c \in C,\ l(c) = (type(c), ref(c), Desc(c))$ with $type(c) \in T_C$, $ref(c) = *$ or $ref(c) \in I$, $Desc(c) = **$ or $Desc(c)$ is a NG. A *complex concept node* is a node with a nonempty (i.e. different from $**$) $Desc$. The set of complex nodes of $G$ is denoted by $D(G)$.

We distinguish the set $C_G$ of concept nodes of a NG $G$ from the set $UC_G$ of concept nodes *appearing in* $G$, i.e. the set of concept nodes of all SGs used in the construction of $G$. The set $UC_G$ is defined by induction on the depth of $G$ (the depth of $G$ is that of the rooted tree $A(G)$, i.e. the maximum length of a path in $A(G)$). $UC_G$ is the union of the set $C_G$ and the sets $UC_{Desc(c)}$ where $c$ is a complex node of $G$ (note that these sets are pairwise disjoint). A projection from a NG $G$ to a NG $H$ is defined by induction on the depth of $G$. A **projection** from a NG $G = (R_G,\ C_G,\ E_G,\ l_G)$ to a NG $H = (R_H,\ C_H,\ E_H,\ l_H)$ is a family of mappings $\Pi = \Pi_0, \Pi_{c_1}\ ...\ \Pi_{c_k}$, with $D(G) = \{c_1\ ...\ c_k\}$, which satisfies:

1. $\Pi_0$ is a projection from the SG $R(G)$ to the SG $R(H)$,

2. $\forall c_i \in D(G),\ \Pi_0(c_i) \in D(H)$ and $\Pi_{c_i}$ is a projection from $Desc(c_i)$ to $Desc(\Pi_0(c_i))$.

A projection $\Pi$ from a NG $G$ to a NG $H$ naturally induces a mapping (also noted $\Pi$ for simplicity) from $UC_G$ to $UC_H$ defined by induction on the depth of $G$. For any $c$ in $UC_G$, if $c$ is in $C_G$ then $\Pi(c)$ is equal to $\Pi_0(c)$ and otherwise $\Pi(c)$ is equal to $\Pi_{c_i}(c)$, where $c_i$ is the node of $D(G)$ such that $c$ is in $UC_{Desc(c_i)}$.

This notion of a projection can be formulated in terms of the rooted trees associated with NGs. Let $A_1 = (X_1, U_1)$ and $A_2 = (X_2, U_2)$ be two rooted trees of SGs with respective roots $R_1$ and $R_2$. An **A-projection** from $A_1$ to $A_2$ is a family of mappings $\varphi = \varphi_0, \varphi_{K_1}\ ...\ \varphi_{K_p}$, with $X_1 = \{K_1\ ...\ K_p\}$, which satisfies:
1. $\varphi_0$ is a mapping from $X_1$ to $X_2$ such that $\varphi_0(R_1) = R_2$,
2. $\forall K \in X_1$, $\varphi_K$ is a projection from $K$ to $\varphi_0(K)$,
3. $\forall (J, c)K \in U_1$, $(\varphi_0(J), \varphi_J(c))\varphi_0(K) \in U_2$.
For all NGs $G$ and $H$, there is a projection from $G$ to $H$ iff there is an A-projection from $A(G)$ to $A(H)$ [Sim96b].

As in SGs, several individual concept nodes with the same marker may appear in a NG, and it may be useful to express that several generic concept nodes represent the same entity. Note that such nodes may be merged if they appear in the same SG of $A(G)$ but not otherwise. Thus, the extension from SGs to NGs makes co-reference links necessary even in applications where the possibility of merging nodes representing the same entity made co-reference links useless in SGs. A NG with co-reference links (in short $\mathbf{NG^{ref}}$) is a graph $G = (R, C, E, l, \textit{co-ref})$, where $\textit{co-ref}$ is an equivalence relation on the set of generic concept nodes appearing in $G$. The relation $\textit{co-ref}$ is extended to an equivalence relation $\textit{co-ident}$ on the set $UC_G$ of concept nodes appearing in $G$ as in the $SG^{ref}$ model. A **projection** $\Pi$ from a $NG^{ref}$ $G$ to a $NG^{ref}$ $H$ must preserve co-identity: $\forall\ c, c' \in UC_G$ if $\textit{co-ref}_G(c, c')$ then $\textit{co-ident}_H(\Pi(c), \Pi(c'))$.

Projection checking between two nested graphs, with or without co-reference links, is an NP-complete problem. For NGs (i.e. without co-reference links) the complexity of this problem is polynomially related to the complexity of projection checking for the class of

[1] More precisely, $R$, $C$ and $E$ may be identified to relation, concept, and edge sets of the SG $R(G)$ respectively, as long as the description field in the concept node labels is not concerned.

simple graphs which make up the NGs. For instance, it is polynomial when these graphs are trees.

## 5 The Semantics $\Phi$

In order to extend $\Phi$ to NG$^{ref}$s, we add one argument to each predicate, called the **context argument**. Thus, a binary predicate is associated with each concept type, and a $(n+1)$-adic predicate is associated with each $n$-adic relation type. For instance, if $x$ (resp. $y$, $z$) is the variable assigned to the generic concept node of type *Announcement* (resp. *Sell*, *Bicycle*), then the interpretation of the atom $obj(x, y, z)$ is: the bicycle $z$ is the object of the sale $y$ *in the context of* the announcement $x$. With $S$ is associated the set $\Phi(S)$ of formulas $\forall z \forall x_1 \ldots x_p(t_2(z, x_1, \ldots, x_p) \rightarrow t_1(z, x_1, \ldots, x_p))$, where $t_1$ and $t_2$ are types such that $t_1 \geq t_2$ and $p$ is the arity of $t_1$ and $t_2$ in the semantics $\Phi$ for SGs. For any NG$^{ref}$ $G$, let $r(G)$ be the number of equivalence classes of the relation $co\text{-}ref_G$. Assign $r(G)$ distinct variables $y_1, \ldots, y_{r(G)}$ to the $co\text{-}ref_G$ classes. For any node $K$ of $A(G)$, if $K$ contains generic nodes then the variables of $\Phi(K)$ are some of the variables $y_1, \ldots, y_{r(G)}$. A formula $\Phi'(t, G')$ is associated with any term $t$ and any subgraph $G'$ of $G$ (i.e. NG$^{ref}$ $G'$ either equal to $G$ or to the description graph of a concept node appearing in $G$). $\Phi'(t, G')$ is the formula associated with $G'$ when $G'$ is in the context represented by the term $t$, or more specifically, when $G'$ is the description graph of a concept node associated with the term $t$. $\Phi'(t, G')$ is defined by induction on the depth of $G'$. For any node $K$ of $A(G)$, $\Phi'(t, K)$ denotes the conjunction of the atoms obtained from those of $\Phi(K)$ by adding the first argument $t$; for any concept node $c$ of $K$, $e_{\overline{c}}$ denotes the term assigned to $c$ in $\Phi(K)$ (which is in fact the term assigned to the co-identity class $\overline{c}$ of $c$).

$$\Phi'(t, G') = \Phi'(t, R(G')) \wedge (\wedge_{c \in D(G')} \Phi'(e_{\overline{c}}, Desc(c)))$$

The formula

$$\mathbf{\Phi(G)} = \exists y_1 \ldots y_{r(G)} \Phi'(a_0, G)$$

is associated with any NG$^{ref}$ $G$ defined on the support $S$, where $a_0$ is a constant representing the general context induced by the support $S$, so that the same constant $a_0$ is used for all NG$^{ref}$s defined on the support $S$.

E.g. the formula associated with the graph $G$ of Figure 3 is

$\Phi(G) = \exists y_2 \ldots y_5 y_7 y_8 (Person(a_0, Peter) \wedge Stick(a_0, y_2) \wedge Announcement(a_0, y_3) \wedge Billboard(a_0, y_4) \wedge Write(a_0, y_5) \wedge agent(a_0, y_2, Peter) \wedge object(a_0, y_2, y_3) \wedge location(a_0, y_2, y_4) \wedge agent(a_0, y_5, Peter) \wedge object(a_0, y_5, y_3) \wedge \Phi'(y_3, Desc(c_3)))$,

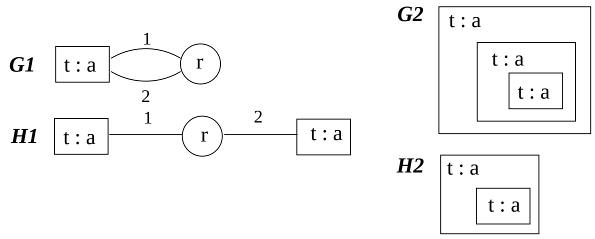



Figure 4: Counterexamples to the completeness result in NG$^{ref}$s with $\Phi$

where $c_3$ is the concept node of type *Announcement* and

$\Phi'(y_3, Desc(c_3)) = Person(y_3, Peter) \wedge Sell(y_3, y_7) \wedge Bicycle(y_3, y_8) \wedge agent(y_3, y_7, Peter) \wedge object(y_3, y_7, y_8)$.

$\Phi(G)$ may also be defined from $A(G)$.

$$\Phi(G) = \exists y_1 \ldots y_{r(G)} (\wedge_{K \text{ node of } A(G)} \Phi'(t_K, K))$$

where $t_K = a_0$ if $K = R(G)$ and otherwise let $(J, c)K$ be the edge of $A(G)$ into $K$, $t_K = e_{\overline{c}}$.

The normality notion and the soundness and completeness result extend to NG$^{ref}$s. Let us first consider two counterexamples to the completeness result (presented in Figure 4). $\Phi(G_1) = t(a_0, a) \wedge r(a_0, a, a)$ and $\Phi(H_1) = t(a_0, a) \wedge t(a_0, a) \wedge r(a_0, a, a)$. $\Phi(G_1)$ and $\Phi(H_1)$ are equivalent formulas, but there does not exist a projection from $G_1$ to $H_1$. The trouble here is that $H_1$ is not a SG$^{ref}$ in normal form. $\Phi(G_2) = t(a_0, a) \wedge t(a, a) \wedge t(a, a)$ and $\Phi(H_2) = t(a_0, a) \wedge t(a, a)$. The trouble here is that there are two co-identical concept nodes appearing in $H_2$ such that one is a complex node and the other is not.

These remarks give some intuition for extending normality to NG$^{ref}$s. Let $G$ be a NG$^{ref}$ and $G'$ and $H'$ be two SG$^{ref}$s nodes of $A(G)$ (resp. two NG$^{ref}$s subgraphs of $G$). An **exact projection** from $G'$ to $H'$ is a projection from $G'$ to $H'$ mapping each generic node $c$ of $C_{G'}$ (resp. $UC_{G'}$) to a generic node of $C_{H'}$ (resp. $UC_{H'}$) co-referent to $c$ in $G$. $G'$ and $H'$ are **exactly equivalent** iff there is an exact projection from $G'$ to $H'$ and from $H'$ to $G'$. A NG$^{ref}$ $G$ is in **normal form** iff

1. every node of $A(G)$ is a SG$^{ref}$ in normal form,

2. for all co-identical concept nodes $c$ and $c'$ appearing in two distinct nodes of $A(G)$, if $c$ is a complex node then $c'$ is a complex node and $R(Desc(c'))$ is exactly equivalent to $R(Desc(c))$.

Note that for all co-identical complex nodes $c$ and $c'$ appearing in a NG$^{ref}$ $G$ in normal form, not only the SG$^{ref}$s $R(Desc(c))$ and $R(Desc(c'))$ are exactly equivalent, but also the whole NG$^{ref}$s $Desc(c)$ and $Desc(c')$. However, putting a NG$^{ref}$ $G$ into normal form (i.e. transforming $G$ into a NG$^{ref}$ $G'$ in normal form such that $\Phi(G) \equiv \Phi(G')$) is not always possible. E.g. consider the graph $H_2$ of Figure 4. If $H'$ were a NG$^{ref}$ in normal form such that $\Phi(H_2) \equiv \Phi(H')$ then $\Phi(H')$ would contain the atom $t(a,a)$, then any concept node of $UC_{H'}$ with referent $a$ would have a concept node with referent $a$ in its description graph, and there would be an infinite chain in $A(H')$, which is impossible. The definition of normality may be weakened into that of $k$-normality in such a way that any NG$^{ref}$ $G$ may be put into $k$-normal form for any $k \geq depth(G)$. The *level* in $G$ of a node $c$ of $UC_G$ is the level in $A(G)$ of the node $K$ of $A(G)$ containing $c$ (i.e. the number of edges of the path in $A(G)$ from $R(G)$ to $K$). A NG$^{ref}$ $G$ is in **$k$-normal form** iff

1. every node of $A(G)$ is a SG$^{ref}$ in normal form,

2. for all co-identical concept nodes $c$ and $c'$ appearing in two distinct nodes of $A(G)$ such that $c$ is a complex node, if the level of $c'$ in $G$ is less than $k$ then $c'$ is a complex node and if $c'$ is a complex node then $R(Desc(c'))$ is exactly equivalent to $R(Desc(c))$.

Note that if $G$ is in normal form then $G$ is in $k$-normal form for any natural integer $k$.

**Theorem 2** *Let $G$ and $H$ be two NG$^{ref}$s and let $k$ be an integer such that $k \geq depth(G)$.*
*If $G$ subsumes $H$ then $\Phi(S), \Phi(H) \models \Phi(G)$. If $H$ is in $k$-normal form and $\Phi(S), \Phi(H) \models \Phi(G)$ then $G$ subsumes $H$.*

**Sketch of proof**: (the detailed proof may be found in [Sim96b]). Lemma 1 is true for NG$^{ref}$s instead of SG$^{ref}$s and the semantics $\Phi$ instead of $\Psi$. And Lemma 2 is true for SG$^{ref}$s and the semantics $\Phi$ instead of $\Psi$ (provided for the implication from right to left that $H$ is a SG$^{ref}$ in normal form). Let us suppose that $G$ subsumes $H$. Let $\varphi = \varphi_0, \varphi_{K_1} \ldots \varphi_{K_p}$ be an A-projection from $G$ to $H$. From Lemma 2, for any node $K$ of $A(G)$, there is a s-substitution $\sigma_K$ from $K$ to $\varphi_0(K)$. Let $\sigma$ be the substitution of the variabes of $\Phi(G)$ obtained from the $\sigma_K$. We prove that $\sigma$ is a s-substitution from $G$ to $H$ and conclude with Lemma 1. Conversely, let us suppose that $H$ is in $k$-normal form with $k \geq depth(G)$ and $\Phi(S), \Phi(H) \models \Phi(G)$. From Lemma 1, there is a s-substitution $\sigma$ from $G$ to $H$. We construct an A-projection $\varphi$ from $A(G)$ to $A(H)$. We define $\varphi_0(K)$ and $\varphi_K$ for any node $K$ of $A(G)$ by induction on the level $l$ of $K$ in $A(G)$. For $l = 0$, $K$ is $R(G)$. The atoms of $\Phi(G)$ (resp $\mathcal{C}(\Phi(H))$) assigned to the nodes of $R(G)$ (resp. $R(H)$) are those with $a_0$ as first argument. Then the restriction of $\sigma$ to the variables of $\Phi(R(G))$ is a s-substitution from $R(G)$ to $R(H)$. From Lemma 2, there is a projection $\Pi_0$ from $R(G)$ to $R(H)$. We define $\varphi_0(R(G)) = R(H)$ and $\varphi_{R(G)} = \Pi_0$. Suppose $\varphi$ is defined up to level $l$. Let $(J,c)K$ be an edge of $A(G)$ with $K$ at level $l+1$. Let $e_1$ be the first argument of the atoms of $\Phi(G)$ assigned to the nodes of $K$. Any atom of $\mathcal{C}(\Phi(H))$ with $\sigma(e_1)$ as first argument is associated with a node of a SG$^{ref}$ $K'$ of $A(H)$, with $K' = R(Desc(c'))$ and $c'$ is co-identical to $\varphi_J(c)$. Then, as $H$ is in $k$-normal form and $level_H(\varphi_J(c)) = level_G(c) < depth(G) \leq k$, $\varphi_J(c)$ is a complex node and $K'$ is exactly equivalent to $R(Desc(\varphi_J(c)))$. From Lemma 3 below, the restriction of $\sigma$ to the variables of $\Phi(K)$ is a s-substitution from $K$ to $R(Desc(\varphi_J(c)))$. From Lemma 2, there is a projection $\Pi_K$ from $K$ to $R(Desc(\varphi_J(c)))$. We define $\varphi_0(K) = R(Desc(\varphi_J(c)))$ and $\varphi_K = \Pi_K$. $\varphi$ is an A-projection from $A(G)$ to $A(H)$, then $G$ subsumes $H$ $\Box$

**Lemma 3**: *For any NG$^{ref}$ $H$, for all exactly equivalent nodes $K$ and $K'$ of $A(H)$ and for any atom $t(a_1, \ldots, a_n)$, there is $t' \leq t$ such that $t'(a_1, \ldots, a_n)$ is an atom of $\mathcal{C}(\Phi(K))$ iff there is $t' \leq t$ such that $t'(a_1, \ldots, a_n)$ is an atom of $\mathcal{C}(\Phi(K'))$.*
**Proof**: the result follows from the fact that if $s$ is a node of $K$ associated with the atom $t'(a_1, \ldots, a_n)$ of $\mathcal{C}(\Phi(K))$ and $\Pi$ is an exact projection from $K$ to $K'$, then the atom of $\mathcal{C}(\Phi(K'))$ associated with the node $\Pi(s)$ of $K'$ is in the form $t''(a_1, \ldots, a_n)$, with $t'' \leq t'$. $\Box$

Any NG$^{ref}$ $G$ may be put into $k$-normal form for any $k \geq depth(G)$ as follows. A rooted tree $A_k$ of SG$^{ref}$s is built level by level from its root to level $k$. Its root is a copy of $R(G)$. If $J$ is a node of $A_k$ at level $l < k$, and if $c$ is a concept node of $J$ ($c$ is a copy of a node $c'$ of $UC_G$) and if there is at least one complex node co-identical to $c'$ in $G$, then add the edge $(J,c)K$ to $A_k$, where $K$ is a copy of the union of the SG$^{ref}$s $R(Desc(c''))$ for every complex node $c''$ co-identical to $c'$ in $G$. Two generic nodes appearing in nodes of $A_k$ are co-referent iff they are copies of co-referent nodes of $UC_G$. Let $G'_k$ be the NG$^{ref}$ such that $A(G'_k)$ is obtained from $A_k$ by putting each node of $A_k$ into normal form (when merging several co-identical concept nodes, only one of the subtrees issued from these nodes is kept, as these subtrees are exact copies of each other in $A_k$). $G'_k$ is in $k$-normal form and $\Phi(G) \equiv \Phi(G'_k)$. $G'_k$ is called *the $k$-normal form* of $G$. E.g. the 2-normal form of the graph $H_2$ of Figure 4 is $H_2$ and its 3-normal form is $G_2$.

The semantics $\Phi$ is unable to express not only that an entity is represented by several concept nodes in a node of $A(G)$, but also that several concept nodes representing the same entity have different descriptions. $\Phi$ may be pertinent in applications in which the meaning of a graph is not changed when merging co-identical concept nodes of a $SG^{ref}$ or replacing the description of co-identical concept nodes by the union of their descriptions (in particular it is the case for applications where graphs are naturally in normal form). But in some applications, each concept node has a specific situation in its $SG^{ref}$ and a specific description; merging these nodes or mixing their descriptions would lead to a loss of information. For instance, let $c$ and $c'$ be two concept nodes appearing in a $NG^{ref}$ $G$ and representing a given lake. $c$ appears in the context of a biological study and its description contains biological information about the lake: animals and plants living in the lake. $c'$ appears in the context of a touristic study and its description contains touristic information about the lake: possibilities of bathing, sailing or walking at the lake. The formulas $\Phi(G)$, $\Phi(H)$, and $\Phi(K)$ are equivalent, where $H$ is obtained from $G$ by exchanging the description graphs of $c$ and $c'$ and $K$ is the $k$-normal form of $G$ and $H$, with $k = max(depth(G), depth(H))$, in which the description of $c$ and $c'$ is the union of the biological and touristic descriptions. Such an equivalence is obviously undesirable. The semantics $\Psi$ is better suited for these applications.

## 6 The Semantics $\Psi$

The semantics $\Psi$ is extended from $SG^{ref}$s to $NG^{ref}$s in the same way as the semantics $\Phi$, except that the context argument is the variable $e_c$ assigned to the concept node $c$ representing the context instead of the term $e_{\overline{c}}$. Thus a description is specific to a concept node rather than to a co-identity class. A context argument is added to each predicate (concept type predicates become 3-adic). For any $NG^{ref}$ $G$ and any node $K$ of $A(G)$, let $nc(K)$ be the number of $K$ concept nodes. Assign $r(G)$ variables $y_1$, ..., $y_{r(G)}$ to the $r(G)$ $co\text{-}ref_G$ classes and, for any node $K$ of $A(G)$, assign $nc(K)$ variables $x_1^K$, ..., $x_{nc(K)}^K$ to $K$ concept nodes. All variables $y_i$ and $x_j^K$ are distinct. The variables of $\Psi(K)$ are $x_1^K$, ..., $x_{nc(K)}^K$ and, if $K$ contains generic nodes, some of the variables $y_1$, ..., $y_{r(G)}$. $\Psi'(t, G')$ is defined by induction on the depth of $G'$. For any node $K$ of $A(G)$, $\Psi'(t, K)$ denotes the conjunction of the atoms obtained from those of $\Psi(K)$ by adding the first argument $t$; for any concept node $c$ of $K$, $e_c$ denotes the variable assigned to $c$ in $\Psi(K)$

$$\Psi'(t, G') = \exists x_1^{R(G')} ... x_{nc(R(G'))}^{R(G')}$$
$$(\Psi'(t, R(G')) \wedge (\wedge_{c \in D(G')} \Psi'(e_c, Desc(c))))$$

The formula

$$\mathbf{\Psi(G)} = \exists y_1 ... y_{r(G)} \Psi'(a_0, G)$$

is associated with any $NG^{ref}$ $G$ defined on the support $S$.

E.g. the formula associated with the graph $G$ of Figure 3 is

$\Psi(G) = \exists y_2 ... y_5 y_7 y_8 (\exists x_1 ... x_5 (Person(a_0, Peter, x_1) \wedge Stick(a_0, y_2, x_2) \wedge Announcement(a_0, y_3, x_3) \wedge Billboard(a_0, y_4, x_4) \wedge Write(a_0, y_5, x_5) \wedge agent(a_0, x_2, x_1) \wedge object(a_0, x_2, x_3) \wedge location(a_0, x_2, x_4) \wedge agent(a_0, x_5, x_1) \wedge object(a_0, x_5, x_3) \wedge \Psi'(x_3, Desc(c_3))))$,

with

$\Psi'(x_3, Desc(c_3)) = \exists x_6 x_7 x_8 (Person(x_3, Peter, x_6) \wedge Sell(x_3, y_7, x_7) \wedge Bicycle(x_3, y_8, x_8) \wedge agent(x_3, x_7, x_6) \wedge object(x_3, x_7, x_8))$.

$\Psi(G)$ may be defined from $A(G)$ as the existential closure of the conjunction of the formulas $\Psi'(t_K, K)$, where $K$ denotes a node of $A(G)$, $t_K = a_0$ if $K = R(G)$ and otherwise $t_K = e_c$, with $(J, c)K$ is the edge of $A(G)$ into $K$.

Projection is sound and complete with respect to $\Psi$.

**Theorem 3** *Let $G$ and $H$ be two $NG^{ref}$s. $G$ subsumes $H$ iff $\Psi(S), \Psi(H) \models \Psi(G)$.*

**Sketch of proof:** A detailed proof is given in [Sim96a], similar to that of Theorem 2 concerning $\Phi$. A more intuitive one is given here, which would not be available for $\Phi$, as the $SG^{ref}$ $Simple(H)$ defined below is not in normal form. Let $S_0$ be the support obtained from $S$ by adding the individual marker $a_0$, a greatest concept type $\top$ (if it does not already exist) and a binary relation type $t_{context}$ (context relation). For any $NG^{ref}$ $G$ on $S$, let $G_0$ be the $NG^{ref}$ on $S_0$ restricted to a concept node labelled $(\top, a_0, G)$ and let $Simple(G)$ be the $SG^{ref}$ on $S_0$ obtained from the union of the nodes of $A(G_0)$ by adding for each edge $(J, c)K$ of $A(G_0)$ $nc(K)$ relation nodes of type $t_{context}$ which relate $c$ to each concept node of $K$. It can be shown that (1) $G$ subsumes $H$ iff $Simple(G)$ subsumes $Simple(H)$ (as both propositions are equivalent to the existence of an A-projection from $A(G)$ to $A(H)$) and (2) $\Psi(S), \Psi(H) \models \Psi(G)$ iff $\Psi(S_0), \Psi(Simple(H)) \models \Psi(Simple(G))$ (as any s-substitution from $G$ to $H$ is a s-substitution from $Simple(G)$ to $Simple(H)$ and conversely). We conclude with the soundness and completeness result on $SG^{ref}$s. □

## 7 Related works

In the conceptual graph domain, several kinds of CGs with complex nodes have been used [CM97]. Most of them follow the formalism proposed by [Sow94]. Briefly, a specific relation type called *description* and a special concept type called *Graph* are introduced.

Concept nodes of type *Graph* may have a graph as referent. Adding an internal information $D$ to a concept node $c$ of a CG $G$ is done by creating another concept node $c'$ of type *Graph* and referent $D$, with $c'$ and $c$ related via a *description* relation. The mapping $\Phi$ for SGs is extended by introducing a special binary predicate *descr* — associated with the *description* relation — whose first argument is a term and second argument is a formula. $\Phi(G)$ contains the atom $descr(e,\ \Phi(D))$, where $e$ is the term associated with $c$. This logical semantics is similar to the logical semantics of context of [McC93], [Guh91], with *descr* playing the same role as the *ist* predicate. [GW96] defines nested conceptual graphs where concept labels have three fields like ours. But these kinds of nested graphs include neither non-trivial graph-theoretic operations (such as projection) for reasoning purposes, nor FOL logical semantics. Another logical semantics for nested graphs, but without co-reference links, is given in [PMC95]. A specific logical language is proposed, where formulas do not belong to classical FOL, and a Gentzen system is defined, which is sound and complete for projection.

At first glance, there is some likeness between NGs and partitioned networks [Hen79]. But these two models actually only share a very general notion of grouping. NGs have been introduced for representing hierarchical knowledge and for doing reasonings respecting the levels. In partitioned networks, "the central idea of partitioning is to allow groups of nodes and arcs to be bundled together into units called spaces, which are fundamental entities in partitioned networks, on the same level as nodes and arcs. [..] Nodes and arcs of different spaces may be linked. [...] Every node and every arc of a network belongs to one or more space" [Hen79]. This quotation points out at least three important differences. First, in NGs, a nested graph is not on the same level as nodes or arcs. Second, nodes of different NGs cannot be individually linked (unless by a co-reference link). Third, nested CGs cannot overlap. Thus, from a descriptive viewpoint, partitioned networks are much more general than NGs. However, as far as we know, partitioned networks are not provided with a morphism notion for doing reasonings.

There are possible links between the NG model and contextual reasoning, although contextual reasoning is not our goal here. Indeed, we developed the NG model for representing knowledge structured by hierarchical levels and doing reasonings which respect these levels. However, NGs may be used for representing a very particular case of contexts (the set of contexts is tree-structured, contextual formulas are restricted to existential positive conjunctive FOL formulas). But none of the logical semantics proposed in this paper seem to be close to one of the logical formalizations of contextual reasoning (e.g. [McC93], [Guh91], [Buv96]). However, more in-depth comparisons will now have to be done.

## 8 Conclusion

In this paper, a graph-based model for hierarchical structured knowledge, "nested graphs", is presented. Projection — a labelled nested graph morphism — is the main reasoning tool. The model is given two FOL semantics, which correspond to different intuitive semantics. Projection proved to be sound and complete with respect to deduction in both cases.

We are developing a workbench, called CoGITo, for building knowledge-based applications, in which every piece of knowledge is described by nested graphs (see [Hae95] [CCC+97] for a previous version based on simple graphs). This workbench forms the skeleton of a modelization language for an industrial application we are currently involved in [BBV97].


### Acknowledgements

We would like to specially thank two referees who drew our attention to the correspondence between the conjunctive query containment in databases and the simple graph projection, as well as M. C. Rousset, who suggested the transformation from NGs to SGs used in the proof of theorem 3.


## References


[AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[BBV97] C. Bos, B. Botella, and P. Vanheeghe. Modeling and simulating human behaviors with conceptual graphs. In *Conceptual Structures: Fulfilling Peirce's Dream, ICCS'97 Proc., LNAI 1257*, pages 275–289. Springer Verlag, 1997.

[Buv96] S. Buvač. Quantificational Logic of Context. In *Proc. AAAI'96*, pages 600—606, 1996.

[CCC+97] B. Carbonneill, M. Chein, O. Cogis, O. Guinaldo, O. Haemmerlé, M.L. Mugnier, and E. Salvat. The COnceptual Graphs at LIRMM Project. In *Proc. of the first CGTOOLS workshop*, pages 5–8, 1997.

[CM92] M. Chein and M.L. Mugnier. Conceptual Graphs: fundamental notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.

[CM97] M. Chein and M.-L. Mugnier. Positive Nested Conceptual Graphs. In *Conceptual Structures: Fulfilling Peirce's Dream, ICCS'97 Proc., LNAI 1257*, pages 95–109. Springer Verlag, 1997.

[Dec92] R. Dechter. Constraint Networks. In *Encyclopedia of Artificial Intelligence*, pages 276—285. Wiley and Sons (second ed.), 1992.

[Guh91] R.V. Guha. Contexts: a formalization and some applications. PhD Thesis, Stanford University , 1991.

[GW96] B.C. Ghosh and V. Wuwongse. Computational Situation Theory in the Conceptual Graph Language. In *Lecture Notes in AI, 1115, Proc. ICCS'96*, pages 188–202, 1996.

[Hae95] O. Haemmerlé. *CoGITo : une plateforme de développement de logiciels sur les graphes conceptuels*. PhD thesis, Université Montpellier II, 1995.

[Hen79] G.C. Hendrix. Encoding Knowledge in Partitioned Networks. In N. Findler, editor, *Associative Networks*, pages 51—92. Academic Press, 1979.

[LE91] R. Levinson and G. Ellis. Multi-Level Hierarchical Retrieval. In *Proc. 6th Annual Workshop on Conceptual Graphs*, pages 67–81, 1991.

[Leh92] F. Lehmann, editor. *Semantics Networks in Artificial Intelligence*. Pergamon Press, 1992.

[MC96] M.L. Mugnier and M. Chein. Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle*, 10(1):7–56, 1996.

[McC93] J. McCarthy. Notes on Formalizing Context. In *Proc. IJCAI'93*, pages 555–560, 1993.

[Mug95] M.L. Mugnier. On generalization/-specialization for conceptual graphs. *J. Expt. Theor. Artif. Intell.*, 7(3):325–344, 1995.

[PMC95] A. Preller, M.-L. Mugnier, and M. Chein. A logic for Nested Graphs. Research Report 95-038, LIRMM, 1995. To appear in Computational Intelligence.

[Rob92] D.D. Roberts. The Existential Graphs. In [Lehmann, 1992], pages 639–663. 1992.

[Sch91] L.K. Schubert. Semantic Networks are in the Eye of the Beholder. In J. F. Sowa, editor, *Principles of Semantic Networks*, pages 95—108. Morgan Kaufmann, 1991.

[SCM98] G. Simonet, M. Chein, and M.-L. Mugnier. Projection in Conceptual Graphs and Query Containment in nr-Datalog. Research Report 98-025, LIRMM, 1998.

[Sim96a] G. Simonet. Une autre sémantique logique pour les graphes emboités. Research Report 96-048, LIRMM, 1996.

[Sim96b] G. Simonet. Une sémantique logique pour les graphes emboités. Research Report 96-047, LIRMM, 1996.

[Sow76] J.F. Sowa. Conceptual graphs for a database interface. *IBM J. Research and development*, 20(4):336–357, 1976.

[Sow84] J.F. Sowa. *Conceptual Structures - Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[Sow94] J.F. Sowa. Logical Foundations for Representing Object-oriented Systems. *J. Expt. Theor. Artif. Intell.*, 5, 1994.

[vdB94] H. van den Berg. Modal Logics for Conceptual Graphs III. In *Proc. ECAI*, pages 356—360, 1994.