The $\mathcal{SG}$ Family: Extensions of Simple Conceptual Graphs

Jean-François Baget
LIRMM
161, rue Ada
34392 Montpellier, Cedex 5
France
`http://www.lirmm.fr/~baget/`

Marie-Laure Mugnier
LIRMM
161, rue Ada
34392 Montpellier, Cedex 5
France
`http://www.lirmm.fr/~mugnier/`

Abstract

We introduce the $\mathcal{SG}$ family of graph-based knowledge representation and reasoning models, basically extensions of the *simple conceptual graphs* model. Objects of these models are *colored simple graphs* and are used to represent *facts*, *rules* and *constraints*. Reasonings are based on graph-theoretic mechanisms, mainly graph homomorphism. Models of this family are defined by the kind of objects composing a knowledge base. In this paper, we focus on the formal definitions of these models, including their operational semantics and relationships with FOL, and we study their decidability properties and computational complexity.

1 Introduction

Conceptual graphs (CGs) have been proposed in [Sowa, 1984] as a knowledge representation and reasoning model, mathematically founded on logics and graph theory. Though mainly studied as a graphical interface for logics or as a diagrammatic system of logics, their graph-theoretic foundations have been less investigated. Most works in this area are limited to *simple conceptual graphs* (*simple graphs* or SGs) [Sowa, 1984; Chein and Mugnier, 1992], corresponding to the positive, conjunctive and existential fragment of FOL. This model has three fundamental characteristics:

1. objects are bipartite *labelled graphs* (nodes represent *entities* and *relations* between these entities);
2. reasonings are based on graph-theoretic operations, mainly a graph homomorphism called *projection*;
3. it is logically founded, reasonings being sound and complete w.r.t. a FOL semantics called Φ.

Main extensions of the SG model, keeping projection-based operations and sound and complete semantics, are *inference rules* [Gosh and Wuwongse, 1995; Salvat, 1998] and *nested graphs* [Chein *et al.*, 1998]; for *general CGs*, [Kerdiles, 1997] introduces an original deduction system, combining analytic tableaux with projection.

We present here a family of extensions of the simple graphs model. The common ground for these extensions is that objects are *colored simple graphs* representing *facts*, *rules* or *constraints*, and operations are based upon projection; the deduction problem asks, given $\mathcal{K}$ a knowledge base and Q a simple graph (which may represent a query, a goal, ...), whether Q can be deduced from $\mathcal{K}$. According to the kinds of objects considered in $\mathcal{K}$, different reasoning models are obtained, composing the $\mathcal{SG}$ family. In this paper, we focus on the formal definitions of these models, including their operational semantics and relationships with FOL, and we study their decidability properties and computational complexity.

In section 2 basic definitions and results about simple graphs are recalled. Section 3 presents an overview of the $\mathcal{SG}$ family. In particular, we explain why we consider SGs graphical features as essential for knowledge modeling and point out that these properties are preserved in the $\mathcal{SG}$ family. In next sections we study the different members of the family.

2 Basic Notions: the $\mathcal{SG}$ Model

Basic ontological knowledge is encoded in a structure called a *support*. Factual knowledge is encoded into *simple graphs* (SGs), which are bipartite labelled multigraphs (there can be several edges between two nodes). Elementary reasonings are computed by a graph homomorphism called *projection*.

Definition 1 (Support) *A* support *is a 4-tuple* $\mathcal{S} = (T_C, T_R, \mathcal{I}, \tau)$. T_C *and* T_R *are two partially ordered finite sets, respectively of* concept types *and* relation types. *Relation types may be of any arity greater or equal to 1.* $\mathcal{I}$ *is the set of* individual markers *and* τ *is a mapping from* $\mathcal{I}$ *to* T_C. *We denote by* $*$ *the* generic marker, *where* $* \notin \mathcal{I}$. *The partial order on* $\mathcal{I} \cup \{*\}$ *considers elements of* $\mathcal{I}$ *as pairwise non comparable, and* $*$ *as its greatest element.*

Definition 2 (Simple Graph) *A* simple graph, *defined on a support* $\mathcal{S}$, *is a bipartite multigraph* $G = (V, E, \lambda)$, *where* $V = (V_C, V_R)$. V_C *and* V_R *are the sets of* concept nodes *and* relation nodes, E *is the set of* edges. *Edges incident on a relation node are numbered from 1 to the degree of the node. We denote by* $G_i(r)$ *the* i^{th} *neighbor of a relation node* r *in* G. *Each node has a label given by the mapping* λ. *A concept node* c *is labelled by a couple* (type(c), marker(c)), *where* type(c) *is an element of* T_C, *called its* type, *and* marker(c) *is an element of* $\mathcal{I} \cup \{*\}$, *called its* marker. *If marker(c) = m is an individual marker, then type(c)=* $\tau(m)$. *A relation node* r *is labelled by* type(r), *an element of* T_R, *called its* type, *and the degree of* r *must be equal to the arity of type(r).*

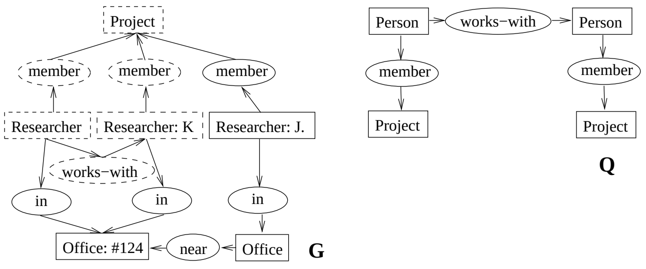

Figure 1: Simple graphs.

In the drawing of a SG, concept nodes are represented by rectangles and relation nodes by ovals. Generic markers are omitted. And since in our examples we use binary relations only, numbers on edges are replaced by directed edges: a relation node is incident to exactly one incoming and one outgoing edge. Fig. 1 shows two (connected) simple graphs assumed to be defined over the same support.

Simple graphs correspond to the existential conjunctive and positive fragment of FOL (by the semantics Φ). Types are mapped to predicates and individual markers to constants. A set of formulas $\Phi(S)$ is assigned to any support $\mathcal{S}$, translating partial orders on types. Given any SG G, a formula $\Phi(G)$ is built as follows. A term is assigned to each concept node: a distinct variable for each generic node, and the constant corresponding to its marker otherwise. An atom $t(c)$ (resp. $t(c_1 \ ... \ c_k)$) is associated to each concept node (resp. relation node r of arity k), where t is the type of the node, and c (resp. c_i) is the term assigned to this node (resp. assigned to $G_i(r)$). Let $\alpha(G)$ be the conjunction of these atoms. $\Phi(G)$ is the existential closure of $\alpha(G)$. E.g. the formula assigned to the dashed subgraph of G in Fig. 1 is $\exists x \exists y (Researcher(x) \wedge Project(y) \wedge Researcher(K) \wedge member(x, y) \wedge member(K, y) \wedge$ *works-with*$(x, K))$.

Definition 3 (Projection) *Let Q and G be two SGs defined on a support $\mathcal{S}$, a* projection *from Q into G is a mapping π from $V_C(Q)$ to $V_C(G)$ and from $V_R(Q)$ to $V_R(G)$ that preserves edges and may decrease node labels:*

1. *$\forall cr \in E(Q), \pi(c)\pi(r) \in E(G)$; and if $c = Q_i(r)$, then $\pi(c) = G_i(\pi(r))$;*
2. *$\forall x \in V(Q)$, $\lambda(\pi(x)) \leq \lambda(x)$ (if x is a concept node, $\leq$ is the product of the orders on T_C and $\mathcal{I} \cup \{*\}$, i.e. $type(\pi(x)) \leq type(x)$ and $marker(\pi(x)) \leq marker(x)$).*

We note $Q \geq G$ (Q subsumes G) if there exists a projection from Q into G. Typically, Q represents a query, G a fact, and projections from Q to G define answers to Q. In Fig. 1, suppose *Researcher* $\leq$ *Person*, then there is one projection from Q into G. The image of Q by this projection is the dashed subgraph of G. Projection is sound and complete w.r.t. the semantics Φ, up to a normality condition for completeness; the *normal form* of a SG G is the SG *nf*(G) obtained by merging concept nodes having the same individual marker. This SG always exists (and is computable in linear time with a naive algorithm). Then: let Q and G be two SGs. $Q \geq nf(G) \Leftrightarrow \Phi(\mathcal{S}), \Phi(G) \models \Phi(Q)$ [Chein and Mugnier, 1992], [Gosh and Wuwongse, 1995].

For the sake of brevity, we consider in what follows that SGs (and more complex constructs built upon SGs) are given in normal form, and put into normal form if needed after a modification. And, since a SG needs not be a connected graph, we confuse a set of SGs with the SG obtained by performing the disjoint union of its elements. In following definition for instance, the SG $\mathcal{G}$ represents a set of SGs.

Definition 4 ($\mathcal{SG}$ Deduction) *Let $\mathcal{G}$ and Q be two SGs. Q can be deduced from $\mathcal{G}$ if $Q \geq \mathcal{G}$.*

The $\mathcal{SG}$ deduction problem is NP-complete [Chein and Mugnier, 1992]. Note that the subsumption relation induced by projection over SGs is a quasi-order. Two graphs are said to be *equivalent* if they project to each other. A SG is said to be *redundant* if it is equivalent to one of its strict subgraphs. Redundancy checking is an NP-complete problem. Each equivalence class admits a unique (up to isomorphism) non redundant graph [Chein and Mugnier, 1992].

3 Overview of the $\mathcal{SG}$ Family

From a modeling viewpoint, the simple graphs model has two essential properties. The objects, simple graphs, are easily understandable by an end-user (a knowledge engineer, or even an expert). And reasonings are easily understandable too, for two reasons: projection is a graph matching operation, thus easily interpretable and visualisable; and the same language is used at interface and operational levels. It follows that reasonings can be explained in a natural manner to the user, step by step, and directly on his own modelization (see for instance the knowledge engineering application described in [Bos *et al.*, 1997] or experiments in document retrieval done by [Genest, 2000], where in both cases the representation language is at the expert level). The $\mathcal{SG}$ family keeps these essential properties.

Let us now informally present the most general model of the family. Throughout this section, we will use examples inspired from a modelization of a knowledge acquisition case study, called Sysiphus-I: it describes a resource allocation problem, where the aim is to assign offices to persons of a research group while fulfilling constraints [Baget *et al.*, 1999].

Simple graphs are the basic constructs, upon which more complex constructs, rules and constraints, are defined, and operations are based on projection. A *rule* expresses a knowledge of form "if A holds then B can be added". It is encoded into a simple graph provided with two colors, highlighting the hypothesis and the conclusion of the rule. In drawings, we represent the hypothesis by white nodes, and the conclusion by gray ones. Rules are used in the following way: if the hypothesis of a rule can be projected into a graph, then the rule is applicable to the graph, and its conclusion can be added to the graph according to the projection. The rule of Fig. 2 can be understood as "for all persons x and y, if x works with y, then y works with x". It can be applied to G of Fig. 1, adding a relation node *(work-with)* with predecessor [Researcher: K] and successor [Researcher].

A *constraint* can be a positive constraint or a negative constraint, expressing a knowledge of form "if A holds, so must B", or "if A holds, B must not". It is also a bicolored simple graph: the first color defines the condition part, and the

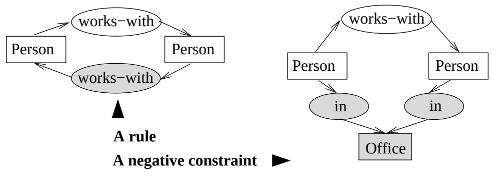

Figure 2: Colored SGs

second color the mandatory (or forbidden) part. A graph G satisfies a positive constraint C if *each* projection from the condition part of C into G can be extended as a projection of the whole C. And G satisfies a negative constraint if *no* projection of C into G can be extended as a projection of the whole C. Fig. 2 represents the negative constraint "two persons working together should not share an office". The graph G of Fig 1 does not satisfy this constraint because "there is a researcher who works with researcher K" (projection of the condition part of C) "and they share office #124" (extension of the projection to a projection of the whole C).

When constraints are involved, we distinguish between two kinds of rules: *inference rules* of form "if A then add B" and *evolution rules* of form "if A then add B, except if it brings inconsistency". Now, a knowledge base contains four sets representing different kinds of knowledge: a set $\mathcal{G}$ of *simple graphs* encoding factual knowledge, a set $\mathcal{R}$ of *inference rules*, a set $\mathcal{E}$ of *evolution rules* and a set $\mathcal{C}$ of *constraints*.

Let us outline the deduction problem: the problem takes in input a knowledge base (KB) and a goal expressed as a SG, and asks whether there is a path of consistent worlds evolving from the initial one to a world satisfying the goal. Factual knowledge describes an initial world; inference rules represent implicit knowledge about worlds; evolution rules represent possible transitions from one world to other worlds; constraints define consistency of worlds; a successor of a consistent world is obtained by an evolution rule application; solving the problem consists in finding a path of consistent worlds evolving from the initial one to a world satisfying the goal.

In the particular case of the Sysiphus-I modelization, $\mathcal{G}$ and $\mathcal{R}$ describe initial information (office locations, persons and group organization), $\mathcal{C}$ represents allocation constraints, $\mathcal{E}$ consists of one evolution rule: "whenever there are a person and an office, try to assign this office to this person". The goal represents a situation where each person of the group has an office. A solution to the problem is a world obtained from the initial one by a sequence of office assignments, where each person has an office, while satisfying allocation constraints.

Let us now specify definitions and notations concerning the $\mathcal{SG}$ family. Rules and constraints are defined as colored SGs.

Definition 5 (colored SGs) *A* colored simple graph *is a pair $K = (G, \rho)$ where G is a SG and ρ is a mapping from $V(G)$ into $\{0, 1\}$. The number associated to a node is called the* color *of the node. We denote by $K_{(i)}$ the subgraph of G induced by i-colored nodes. The subgraph $K_{(0)}$ must form a SG (i.e. the neighbors of a relation node of the hypothesis must also belong to the hypothesis).*

A KB is denoted by $\mathcal{K} = (\mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{C})$. Given a KB $\mathcal{K}$ and a goal Q, the *deduction problem* asks whether Q can be deduced from $\mathcal{K}$ (we note $Q \geq \mathcal{K}$). If we impose some of the sets $\mathcal{R}$, $\mathcal{E}$ or $\mathcal{C}$ to be empty, one obtains specific reasoning models. Note that in the absence of constraints ($\mathcal{C} = \emptyset$), inference and evolution rules have the same behavior, thus $\mathcal{R}$ and $\mathcal{E}$ can be confused. The $\mathcal{SG}$ family is then composed of the six following models.

- the $\mathcal{SG}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\emptyset$, $\emptyset$, $\emptyset$)
- the $\mathcal{SR}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\mathcal{R}$, $\mathcal{E}$, $\emptyset$)
- the $\mathcal{SGC}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\emptyset$, $\emptyset$, $\mathcal{C}$)
- the $\mathcal{SRC}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\mathcal{R}$, $\emptyset$, $\mathcal{C}$)
- the $\mathcal{SEC}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\emptyset$, $\mathcal{E}$, $\mathcal{C}$)
- the $\mathcal{SREC}$ model for $\mathcal{K}$ = ($\mathcal{G}$, $\mathcal{R}$, $\mathcal{E}$, $\mathcal{C}$)

Since a fact has the same semantics as a rule with an empty hypothesis, the set $\mathcal{G}$ is only used in models names when both rule sets $\mathcal{R}$ and $\mathcal{E}$ are empty. The hierarchy of these models is represented in Fig. 3. It highlights the decidability and complexity of the associated deduction problems.

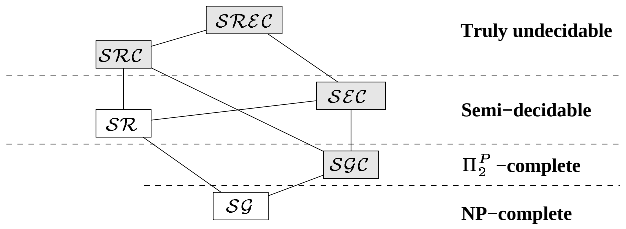

Figure 3: Reasoning models hierarchy for the $\mathcal{SG}$ family

4 SGs and Rules: the $\mathcal{SR}$ Model

A *simple graph rule* (SG rule) embeds knowledge of form "if A then B". The following definition is equivalent to the more traditional one (two SGs related with coreference links) used in [Gosh and Wuwongse, 1995; Salvat, 1998].

Definition 6 (SG Rules) *A* simple graph rule *R is a colored SG. $R_{(0)}$ is called its* hypothesis*, and $R_{(1)}$ its* conclusion.

Deduction depends on the notion of a *rule application*: it is a graph transformation based upon projection.

Definition 7 (Application of a SG Rule) *Let G be a SG, and R be a SG rule. R is* applicable *to G if there exists a projection, say π, from $R_{(0)}$ (the hypothesis of R) into G. In that case, the result of the* application *of R on G according to π is the graph G' obtained by making the disjoint union of G and of a copy of $R_{(1)}$ (the conclusion of R), then, for every edge cr, where $c \in R_{(0)}$ and $r \in R_{(1)}$, adding an edge with the same number between $\pi(c)$ and the copy of r. G' is said to be an* immediate *R*-derivation *from G.*

A derivation is naturally defined as a (possibly empty) sequence of rule applications:

Definition 8 (Derivation) *Let $\mathcal{R}$ be a set of SG rules, and G be a SG. We call $\mathcal{R}$*-derivation *from G to a SG G' a sequence of SGs $G = G_0, \ldots, G_k = G'$ such that, for $1 \leq i \leq k$, G_i is an immediate R-derivation from G_{i-1}, where $R \in \mathcal{R}$.*

To deduce a SG Q, we must be able to derive a SG into which Q can be projected, hence the following definition:

Definition 9 (Deduction in $\mathcal{SR}$) *Let $\mathcal{K} = (\mathcal{G}, \mathcal{R})$ be a KB and let Q be a SG. Q can be deduced from $\mathcal{K}$ (notation $Q \geq (\mathcal{G}, \mathcal{R})$) if there exists an $\mathcal{R}$-derivation from $\mathcal{G}$ to a SG H such that $Q \geq H$.*

The semantics Φ is extended to translate SG rules: let R_0 and R_1 be two SGs, s.t. $R_0 = R_{(0)}$ and R_1 is the SG obtained from $R_{(1)}$ by adding the neighbors of the relation nodes of $R_{(1)}$ which are concept nodes of $R_{(0)}$. Then $\Phi(R) = \forall x_1 \dots x_p \ (\alpha(R_0) \rightarrow \exists y_1 \dots y_q \ \alpha(R_1))$ where x_i are the variables of $\alpha(R_0)$ and y_j are the variables of $\alpha(R_1)$ that do not appear in $\alpha(R_0)$. The following soundness and completeness result is obtained: $Q \geq (\mathcal{G}, \mathcal{R}) \Leftrightarrow \Phi(\mathcal{S}), \Phi(\mathcal{G}), \Phi(\mathcal{R}) \models \Phi(Q)$ [Salvat, 1998]. The $\mathcal{SR}$ deduction problem is semi-decidable [Coulondre and Salvat, 1998].

5 SGs and Constraints: the $\mathcal{SGC}$ Model

Let us now introduce *constraints*, which are used to validate knowledge. In presence of constraints, deduction is defined only on a *consistent* knowledge base.

Definition 10 (Constraints) *A* positive *(resp.* negative*) constraint C is a colored SG. $C_{(0)}$ is called the* trigger *of the constraint, $C_{(1)}$ is called its* obligation *(resp.* interdiction*). A SG G π-*violates *a positive (resp. negative) constraint C if π is a projection of the trigger of C into the non redundant form of G (resp. into G) that cannot be extended (resp. that can be extended) to a projection of C as a whole. G* violates *C if it π-violates C for some projection π. Otherwise, G* satisfies *C.*

Notice there may be two equivalent SGs, such that one satisfies a positive constraint and the other does not. E.g. given a constraint C, take G satisfying C, and the equivalent (redundant) graph H obtained by making the disjoint union of G and the trigger of C. H violates C. We have thus chosen to define constraint satisfaction w.r.t. the non redundant form of a SG. This problem does not occur with negative constraints.

Two constraints C_1 and C_2 are said *equivalent* if, for every graph G, G violates C_1 iff G violates C_2. Any negative constraint C is equivalent to the negative constraint obtained from C by coloring all its nodes by 1. Furthermore, negative constraints are indeed a particular case of positive ones: consider the positive constraint C' obtained from a negative one C by coloring all nodes of C by 0, then adding a concept node typed `NotThere`, colored by 1, where `NotThere` is incomparable with all other types and does not appear anywhere excepted in $\mathcal{C}$. Then a graph G violates C if and only if it violates C'. Positive constraints strictly include negative constraints, in the sense that the associated consistency problems are not in the same complexity class (see theorem 3).

Let us relate our definitions to other definitions of constraints found in the CG literature. The constraints of [Mineau and Missaoui, 1997] correspond to a particular case of our constraints where the trigger and the obligation are not connected. In turn, our constraints are a particular case of the minimal descriptive constraints of [Dibie *et al.*, 1998], where the disjunctive part is restricted to one graph.

Definition 11 (Consistency/Deduction in $\mathcal{SGC}$) *A KB $\mathcal{K} = (\mathcal{G}, \mathcal{C})$ is* consistent *if $\mathcal{G}$ satisfies all constraints of $\mathcal{C}$. A SG Q can be* deduced *from $\mathcal{K}$ if $\mathcal{K}$ is consistent and Q can be deduced from $\mathcal{G}$.*

Note that a Q that violates a constraint of $\mathcal{K}$ may still be deduced from $\mathcal{K}$. It does not matter since Q is a *partial* representation of knowledge deducible from $\mathcal{K}$. How can we translate the notion of consistency into FOL? For negative constraints, the correspondence is immediate, and relies on projection soundness and completeness.

Theorem 1 *A SG G violates a negative constraint $C = (C', \rho)$ iff $\Phi(\mathcal{S}), \Phi(G) \models \Phi(C')$, where C' is the SG underlying C.*

Consistency relative to positive constraints can be explained with FOL, translating "projection" into a notion of logical "substitution" between the formulas associated to graphs (see for instance the S-substitution of [Chein and Mugnier, 1992]). Another bridge can be built using rules. Indeed, a graph G satisfies a positive constraint C if and only if, considering C as a rule, all applications of C on G produce a graph equivalent to G. Or, more specifically:

Property 1 *A SG G π-violates a positive constraint C iff, considering C as a rule, the application of C on G according to π produces a graph not equivalent to G.*

Using soundness and completeness of the $\mathcal{SR}$ deduction, and property 1, one obtains the following relation with FOL:

Theorem 2 *A SG G violates a positive constraint C iff there is a SG G' such that $\Phi(\mathcal{S}), \Phi(G), \Phi(C) \models \Phi(G')$ and not $\Phi(\mathcal{S}), \Phi(G) \models \Phi(G')$, where $\Phi(C)$ is the translation of C considered as a rule.*

The problem "does a given graph satisfy a given constraint?" is co-NP-complete if this constraint is a negative constraint (we must check the absence of projection), but becomes Π_2^P-complete for a positive one (Π_2^P is co-NP$^{\text{NP}}$).

Theorem 3 (Complexity in $\mathcal{SGC}$) *Consistency in $\mathcal{SGC}$ is Π_2^P-complete (but is co-NP-complete if all constraints are negative).*

Sketch of proof: $\mathcal{SGC}$-Consistency belongs to Π_2^P since it corresponds to the language $L = \{x | \forall y_1 \exists y_2 R(x, \ y_1, \ y_2)\}$, where x encodes an instance $G; C$ and $(x, \ y_1, \ y_2) \in R$ iff y_1 is a projection Π_0 from $C_{(0)}$ into G, y_2 is a projection Π from C into G s.t. $\Pi[C_{(0)}] = \Pi_0$. Now, let us consider the Π_2^P-complete problem B_2^c: given a boolean formula E, and a partition $\{X_1, \ X_2\}$ of its variables, is it true that for any truth assignment for the variables in X_1 there exists a truth assignment for the variables in X_2 s.t. E is true? We first show that the special case where E is an instance of 3-SAT remains Π_2^P-complete (let us call this problem 3-SAT$_2^c$). For that we use the polynomial transformation from any boolean formula to a set of clauses described in ([Papadimitriou, 1994], example 8.3), followed by a polynomial transformation from a clause to clauses with 3 literals. The "new" variables are put in the set X_2. We then reduce 3-SAT$_2^c$ to $\mathcal{SGC}$-Consistency. E is mapped to a constraint C: briefly said, there is one concept node [t:*] for each variable and one ternary relation node with type r_i for each clause c_i. $C_{(0)}$ is composed of all concept nodes coming from variables in X_1. G is composed of two individual concept nodes [t:0] and [t:1] corresponding to the truth values and, for each clause c_i there is one relation node of type r_i for each triple of truth values giving the value true to c_i. Note that, since $C_{(0)}$ does not contain any relation node, any truth assignment for the variables of X_1 is a projection from $C_{(0)}$ to G, and reciprocally. □

Corollary 1 *Deduction in $\mathcal{SGC}$ is Π_2^P-complete.*

Note that the theorem 3 can be used to show that consistency of minimal descriptive constraints in [Dibie *et al.*, 1998] is also Π_2^P-complete.

6 Rules and Constraints: $\mathcal{SEC}/\mathcal{SRC}$

The two kinds of rules, inference rules $\mathcal{R}$ and evolution rules $\mathcal{E}$, define two alternative models. In $\mathcal{SEC}$, $\mathcal{G}$ is seen as the initial world, root of a potentially infinite tree of possible worlds, and $\mathcal{E}$ describes the possible evolutions from one world to others. The deduction problem asks whether there is *a path of consistent worlds from $\mathcal{G}$ to a world satisfying Q*.

Definition 12 (Deduction in $\mathcal{SEC}$) *Let $\mathcal{K} = (\mathcal{G}, \mathcal{E}, \mathcal{C})$ be a KB, and let Q be a SG. Q can be deduced from $\mathcal{K}$ if there is an $\mathcal{E}$-derivation $\mathcal{G} = G_0, \ldots, G_k$ such that, for $0 \leq i \leq k$, $(G_i, \mathcal{C})$ is consistent and Q can be deduced from G_k.*

In $\mathcal{SRC}$, $\mathcal{G}$ provided with $\mathcal{R}$ is a finite description of a potentially infinite world, that has to be consistent. Applying a rule to $\mathcal{G}$ can create inconsistency, but a further application of a rule may restore consistency. Let us formalize this notion of *consistency restoration*. Suppose there is a π-violation of a positive constraint C in $\mathcal{G}$; this violation (C, π) is said to be *$\mathcal{R}$-restorable* if there exists an $\mathcal{R}$-derivation from $\mathcal{G}$ into G' such that the projection π of the trigger of C into G' can be extended to a projection of C as a whole. The violation of a negative constraint can never be restored. Note that the $\mathcal{R}$-restoration can create new violations, that must themselves be proven $\mathcal{R}$-restorable.

Definition 13 (Consistency/Deduction in $\mathcal{SRC}$) *A KB $\mathcal{K} = (\mathcal{G}, \mathcal{R}, \mathcal{C})$ is* consistent *if, for any SG G' that can be $\mathcal{R}$-derived from $\mathcal{G}$, for every constraint $C \in \mathcal{C}$, for every π-violation of C in G', (C, π) is $\mathcal{R}$-restorable. A SG Q can be deduced from $\mathcal{K}$ if $\mathcal{K}$ is consistent and Q can be deduced from $(\mathcal{G}, \mathcal{R})$.*

Consider for instance a KB containing the SG G in Fig. 4, expressing the existence of the number 0, a constraint and a rule, both represented by the colored SG K. The constraint asserts that *for every integer n, there must be an integer n', successor of n.* If the rule is an evolution rule, G is seen as an inconsistent initial world (there is no successor of 0 in G) and nothing will be deduced from this KB. If the rule is an inference rule, its application immediately repairs the constraint violation, while creating a new integer, that has no successor, thus a new violation. Finally, every constraint violation could eventually be repaired by a rule application, and the KB should be proven consistent.

Let us point out that the $\mathcal{SR}$ model is obtained from $\mathcal{SRC}$ or $\mathcal{SEC}$ when $\mathcal{C}$ is empty, and $\mathcal{SGC}$ is obtained from $\mathcal{SRC}$ (resp. $\mathcal{SEC}$) when $\mathcal{R}$ (resp. $\mathcal{E}$) is empty.

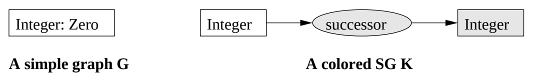

Figure 4: Consistency in $\mathcal{SEC}/\mathcal{SRC}$

Theorem 4 (Complexity in $\mathcal{SEC}/\mathcal{SRC}$) *Deduction in $\mathcal{SEC}$ is semi-decidable. Consistency and deduction in $\mathcal{SRC}$ are truly undecidable.*

Proof: $\mathcal{SEC}$ includes $\mathcal{SR}$ thus $\mathcal{SEC}$-deduction is not decidable. When Q is deducible from $\mathcal{K}$, a breadth-first search of the tree of all derivations from $\mathcal{K}$, each graph being checked for consistency, ensures that G_k is found in finite time. For $\mathcal{SRC}$, we show that checking consistency is truly undecidable. Let $\mathcal{K}$ be a KB where $\mathcal{C}$ contains a positive constraint C^+ and a negative constraint C^-, both with an empty trigger. For proving consistency, one has to prove that $C^- \not\geq (\mathcal{G}, \mathcal{R})$, and the algorithm does not stop in this case (from semi-decidability of deduction in $\mathcal{SR}$). The same holds for the complementary problem (proving inconsistency) taking C^+ instead of C^-, hence the undecidability. □

7 Combining Inference and Evolution

The $\mathcal{SREC}$ model combines both derivation schemes of the $\mathcal{SRC}$ and $\mathcal{SEC}$ models. Now, $\mathcal{G}$ describes an initial world, inference rules of $\mathcal{R}$ complete the description of any world, constraints of $\mathcal{C}$ evaluate the consistency of a world, evolution rules of $\mathcal{E}$ try to make evolve a consistent world into a new, consistent one. The deduction problem is: can $\mathcal{G}$ evolve into a consistent world satisfying the goal?

Definition 14 (Deduction in $\mathcal{SREC}$) *A SG G' is an* immediate $\mathcal{RE}$-evolution *from a SG G if there exists an $\mathcal{R}$-derivation from G into G'' and an* immediate *$\mathcal{E}$-derivation from G'' into G'. An* $\mathcal{RE}$-evolution *from a SG G to a SG G' is a sequence of SGs $G = G_0, \ldots, G_k = G'$ such that, for $1 \leq i \leq k$, G_i is an immediate $\mathcal{RE}$-evolution from G_{i-1}. Given a KB $\mathcal{K} = (\mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{C})$, a SG Q can be* deduced *from $\mathcal{K}$ if there is an $\mathcal{RE}$-evolution $\mathcal{G} = G_0, \ldots, G_k$ where, for $0 \leq i \leq k$, $(G_i, \mathcal{R}, \mathcal{C})$ is consistent, and Q can be deduced from $(G_k, \mathcal{R})$.*

When $\mathcal{E} = \emptyset$, one obtains the $\mathcal{SRC}$ model (there is only one world). When $\mathcal{R} = \emptyset$, one obtains $\mathcal{SEC}$ (information contained in a world is complete). As a generalization of $\mathcal{SRC}$, deduction in $\mathcal{SREC}$ is truly undecidable. Let us now consider a decidable fragment of $\mathcal{SREC}$ (which in particular was sufficient for the Sysiphus-I modelization). First note a rule has only to be applied once according to a given projection: further applications with this projection obviously produce redundant information. Such applications are said to be *useless*. A SG G is said to be *closed* w.r.t. a rule R if all applications of R on G are useless. Given a set of rules $\mathcal{R}$, we note, when it exists, $G^*_{\mathcal{R}}$ (the closure of G w.r.t. $\mathcal{R}$) the smallest graph derived from G that is closed w.r.t. every rule in $\mathcal{R}$. When it exists $G^*_{\mathcal{R}}$ is unique. $\mathcal{R}$ is called a *finite expansion set* (f.e.s.) if, for every SG G, its closure $G^*_{\mathcal{R}}$ exists (and thus can be computed in finite time). If $\mathcal{R}$ is a finite expansion set, the deduction problem in the $\mathcal{SR}$ model becomes decidable (but it is not a necessary condition for decidability).

Property 2 (Finite expansion sets) *Let $\mathcal{K} = (\mathcal{G}, \mathcal{R}, \mathcal{C})$ be a KB where $\mathcal{R}$ is a finite expansion set. Then $\mathcal{K}$ is consistent iff $(\{\mathcal{G}^*_{\mathcal{R}}\}, \mathcal{C})$ is consistent, and a SG Q can be deduced from $(\mathcal{G}, \mathcal{R})$ iff Q can be deduced from $(\{\mathcal{G}^*_{\mathcal{R}}\})$.*

The following decidable case is based on property 2.

Property 3 (Decidable case) *Deduction in $\mathcal{SREC}$ is semi-decidable if $\mathcal{R}$ is a f.e.s. and is decidable if $\mathcal{R} \cup \mathcal{E}$ is a f.e.s.*

Proof: Suppose $\mathcal{R}$ is a f.e.s. When Q can be deduced, one obtains an answer in finite time; we proceed as for $\mathcal{SEC}$ (see proof of theorem 4) but consistency checks are done on the graph closure instead of the graph itself. Now, if $\mathcal{R} \cup \mathcal{E}$ is a f.e.s., $\mathcal{G}^*_{\mathcal{R}\cup\mathcal{E}}$ exists, thus the derivation tree is finite, and consistency checks may only cut some parts of this tree. $\square$

A rule is said to be *range restricted* (by analogy with the so-called rules in Datalog, where all variables of the head must appear in the conclusion) if no generic concept node belongs to its conclusion. Then:

Property 4 *A set of range restricted rules is a f.e.s.*

Proof: Since all graphs are put into normal form, an individual marker appears at most once in a graph. Then the closure of a SG G can be obtained with a derivation of length $\mathcal{O}(n^k)$, where n is the size of $(G, \mathcal{R})$ and k is the greatest arity of a relation type appearing in a rule conclusion. $\square$

8 Conclusion

We propose a family of models that can be seen as the basis of a generic modeling framework. Main features of this framework are the following: a clear distinction between different kinds of knowledge, that fit well with intuitive categories, a uniform graph-based language that keeps essential properties of the SG model, namely readability of objects as well as reasonings. We guess this later point is particularly important for the usability of any knowledge based system. In our framework, all kinds of knowledge are graphs easily interpreted, and reasonings can be graphically represented in a natural manner using the graphs themselves, thus explained to the user on its own modelization.

Technical contributions, w.r.t. previous works on conceptual graphs, can be summarized as follows:

- the representation of different kinds of knowledge as colored SGs: facts, inference rules, evolution rules and constraints.
- the integration of constraints into a reasoning model; more or less similar notions of a constraint had already been introduced in [Dibie *et al.*, 1998; Mineau and Missaoui, 1997] but were only used to check consistency of a simple graph (as in the $\mathcal{SGC}$ model). The complexity of consistency checking was not known.
- a systematic study of the obtained family of models with a complexity classification of associated consistency/deduction problems.

We established links between consistency checking and FOL deduction. The operational semantics of models including constraints, namely $\mathcal{SREC}$, $\mathcal{SRC}$ and $\mathcal{SEC}$, is easy to understand but there is an underlying non monotonic mechanism whose logical interpretation should require non standard logics. The definition of a logical semantics for these models is an open problem.

Acknowledgments

We are indebted to Geneviève Simonet for her careful reading and some error corrections on an earlier version of this paper, and to Michel Chein, as well as the anonymous reviewers, for their helpful comments.

References

[Baget *et al.*, 1999] J.-F. Baget, D. Genest, and M.-L. Mugnier. Knowledge Acquisition with a Pure Graph-Based Knowledge Representation Model – Application to the Sisyphus-I Case Study. In *Proc. of KAW'99*, 1999.

[Bos *et al.*, 1997] C. Bos, B. Botella, and P. Vanheeghe. Modeling and Simulating Human Behaviors with Conceptual Graphs. In *Proc. of ICCS'97, LNAI 1257*, pages 275–289. Springer, 1997.

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

[Chein *et al.*, 1998] M. Chein, M.-L. Mugnier, and G. Simonet. Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics. In *Proc. of KR'98*, pages 524–534. Morgan Kaufmann, 1998.

[Coulondre and Salvat, 1998] S. Coulondre and E. Salvat. Piece Resolution: Towards Larger Perspectives. In *Proc. of ICCS'98, LNAI 1453*, pages 179–193. Springer, 1998.

[Dibie *et al.*, 1998] J. Dibie, O. Haemmerlé, and S. Loiseau. A Semantic Validation of Conceptual Graphs. In *Proc. of ICCS'98, LNAI 1453*, pages 80–93. Springer, 1998.

[Genest, 2000] D. Genest. *Extension du modèle des graphes conceptuels pour la recherche d'informations*. PhD thesis, Université Montpellier II, Dec. 2000.

[Gosh and Wuwongse, 1995] B. C. Gosh and V. Wuwongse. A Direct Proof Procedure for Definite Conceptual Graphs Programs. In *Proc. of ICCS'95, LNAI 954*, pages 158–172. Springer, 1995.

[Kerdiles, 1997] G. Kerdiles. Projection: A Unification Procedure for Tableaux in Conceptual Graphs. In *Proc. of TABLEAUX'97, LNAI 1227*, pages 216–230, 1997.

[Mineau and Missaoui, 1997] G. W. Mineau and R. Missaoui. The Representation of Semantic Constraints in Conceptual Graphs Systems. In *Proc. of ICCS'97, LNAI 1257*, pages 138–152. Springer, 1997.

[Papadimitriou, 1994] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Salvat, 1998] E. Salvat. Theorem proving using graph operations in the conceptual graphs formalism. In *Proc. of ECAI'98*, 1998.

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