Some Structural Properties of the Logic of Rules

Jean-François Baget, Marie-Laure Mugnier, Michel Leclère, Eric Salvat

Some Structural Properties of the Logic of Rules

Jean-François Baget
INRIA / LIRMM
baget@lirmm.fr

Michel Leclère
LIRMM
leclere@lirmm.fr

Marie-Laure Mugnier
LIRMM
mugnier@lirmm.fr

Éric Salvat
IMERIR
salvat@imerir.com

Abstract

We study the properties of semantic consequence in a particular subset of first-order logic with equality and no function symbols, under the unique name assumption. Formulas in this subset of logic are rules of form $\forall\vec{x}(H \rightarrow \exists\vec{y}C)$ where the hypothesis H and the conclusion C are conjunctions of atoms (possibly with equality), $\vec{x}$ contains the variables in H and $\vec{y}$ the variables in C that are not in H. This subset is particularly expressive since it can encode a universal Turing machine, at the cost of decidability of reasoning.

We propose new decidable subclasses of this problem, by combining restrictions on both the structure of the rules themselves and the structure of the interactions between rules, encoded in the graph of rule dependencies. The most general decidable subclass presented here is based on a mixed forward/backward chaining algorithm. Finally, we relate our rules to other notions (tuple-generating dependencies in databases, conceptual graph rules, the TBox in description logics) and explain how the associated deduction problems could benefit from our results.

Introduction

In this paper, we study satisfiability, validity, and semantic consequence of formulas in the rule fragment of first-order logic (FOL) without functions, with equality, and under the unique name assumption, as often the case in KR. Its formulas are rules of form $\forall\vec{x}(H \rightarrow \exists\vec{y}C)$ where the hypothesis H and the conclusion C are conjunctions of atoms. This fragment of FOL is particularly expressive: it is a computation model (it can encode a universal Turing machine), but satisfiability and deduction are undecidable.

Our goal is to present new decidable subsets of this rule logic. To achieve that goal, we study the internal structure of rules as well as the structure of their interactions, and combine them in a mixed forward/backward chaining framework. The foundations of this work have been developed using the *conceptual graphs* (CGs) formalism: a graph-theoretical encoding of these formulas highlighting their *structure*. This work extends our previous results by considering a more expressive language (equality is added), and by strictly generalizing the decidable fragment of rules (the notion of finite unification sets is new).

In the first section, we show that we can restrict our study, without loss of generality, to the satisfiability and validity of *standard knowledge bases*, partitioned into *facts*, *standard rules*, *constraints*, and *equality rules*, and to deducing a fact from such a KB. In the following section, we present a forward chaining algorithm (SM96) that allows us to state the complexity and decidability of our three decision problems in different fragments of our logic of rules. *Finite expansion sets* of (BM02) ensure that forward chaining will generate in finite time a fact that contains all information that can be deduced from rules. Then, we present the backward chaining mechanism of (SM96). Its originality resides in its ability, in the rewriting of queries, to erase more than one atom (since the *unifiers* involved unify whole subsets of atoms at the same time). The *finite unification sets* introduced here ensure that backward chaining will generate a finite number of non redundant rewritings of the query. The next section combines finite expansion and finite unification sets in a mixed forward/backward chaining algorithm, relying on the *graph of rule dependencies* of (Bag04). From this algorithm stems a new decidable fragment for the logic of rules. Finally, we relate our rules to other notions (tuple-generating dependencies in databases, conceptual graph rules, the TBox in description logics) and explain how the associated deduction problems could benefit from our results.

Preliminaries

Definition 1 (Facts, rules) *An* atom *is either a* standard atom $p(t_1, \ldots, t_k)$ *built from a predicate* p *of arity* k *and a* k*-tuple of terms (variables or constants, but* no other function symbols*), the* true atom ■*, the* false atom □ *or an* equality atom $x \doteq y$ *between two terms. A* fact *is a set of atoms. A* rule *is a couple* (H, C) *of facts where* H *is called the* hypothesis *of the rule and* C *its conclusion. We note respectively terms*(X)*, vars*(X) *and consts*(X) *the sets of terms, variables and constants appearing in the atoms of a fact or a rule* X*. If* $R = (H, C)$ *is a rule, we call* frontier *of* R *and note fr*(R) *the set of variables that appear both in* H *and* C*.*

Semantics

The translation Φ assigns a closed formula in first-order logic (FOL) to each fact or rule. If F is a fact, we note $F^\wedge$ the conjunction of all atoms in F. Then $\Phi(F)$ is the existential closure of $F^\wedge$. If $R = (H, C)$ is a

rule, then $\Phi(R) = \forall h_1 \dots \forall h_k (H^\wedge \rightarrow (\exists c_1 \dots \exists c_q C^\wedge))$ where $\{h_1, \dots, h_k\} = \mathit{vars}(H)$ and $\{c_1, \dots, c_q\} = \mathit{vars}(C) \setminus \mathit{vars}(H)$. $\Phi(R)$ can be equivalently written as $\forall f_1 \dots \forall f_p ((\exists h_1 \dots \exists h_l H^\wedge) \rightarrow (\exists c_1 \dots \exists c_m C^\wedge))$ where $\{f_1, \dots, f_p\} = \mathit{fr}(R)$, $\{h_1, \dots, h_l\} = \mathit{vars}(H) \setminus \mathit{fr}(R)$ and $\{c_1, \dots, c_m\} = \mathit{vars}(C) \setminus \mathit{fr}(R)$. Every formula in the FOL fragment naturally associated with the forms above can be translated back into a fact or a rule (this inverse translation removes multiple occurrences of the same atom). We consider the usual semantics of FOL formulas with equality, assuming, as often in KR, the *unique name assumption* (two distinct constants cannot have the same interpretation). A model of a fact or rule X is thus a model of the formula $\Phi(X)$, X is said *valid* (resp. *satisfiable*) when $\Phi(X)$ is valid (resp. satisfiable), and X is a *semantic consequence* of the facts or rules $X_1, \dots, X_p$ (and we note $X_1, \dots, X_p \models X$) when $\Phi(X)$ is a semantic consequence of $\Phi(X_1), \dots, \Phi(X_p)$. Two facts or rules X and X' are said *equivalent* (and we note $X \equiv X'$) iff $X \models X'$ and $X' \models X$.

Since any fact F is equivalent to a rule $(\emptyset, F)$, we can consider that a knowledge base (KB) is composed of rules. The general problems we address here can be stated as follows: given a KB, is its set of rules satisfiable (resp. valid) ? Given a KB and a rule R, is R a semantic consequence of the rules in the KB ? We respectively call these decision problems SATISFIABILITY?, VALIDITY? and DEDUCTION?.

Standard knowledge bases

We will now distinguish between different kinds of rules, and refine the above problems according to the kinds of rules that composes the KB.

Definition 2 (Standard KBs) *A* standard fact *is either a fact that contains only standard atoms or the* absurd fact $\{\Box\}$. *A* standard rule *is a rule (H, C) where both H and C are non empty disjoint sets of standard atoms. A* constraint *is a rule of form $(H, \{\Box\})$ where H is a non empty set of standard atoms. We note this constraint $\neg H$ since $\Phi((H, \{\Box\})) \equiv \neg\Phi(H)$. An* equality rule *is a rule of form $(H, \{t \doteq t'\})$ where H is a non empty set of standard atoms and t, t' are either two distinct variables in $\mathit{vars}(H)$ or a constant and a variable in H. A* standard KB *is a KB that contains one standard fact, a set of standard rules, a set of constraints, and a set of equality rules.*

Property 1 (Standard KBs) *Any rule KB is semantically equivalent to a standard KB.*

Computing an equivalent standard KB takes linear time. From now on, we thus focus on standard KBs.

Depending on the kind of rules a (standard) KB contains, different logical fragments are obtained. Let us note $\mathcal{F}$ a standard fact, $\mathcal{R}$ a set of standard rules, $\mathcal{E}$ a set of equality rules and $\mathcal{C}$ a set of constraints. The fragment $\mathcal{FRCE}$ is obtained with KBs of form $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{E}, \mathcal{C})$, it thus encodes the whole rules fragment (PROP. 1). More specific fragments are obtained by considering only some of the above sets (they are denoted by the names of the sets considered, e.g. the $\mathcal{F}$-fragment or the $\mathcal{FC}$-fragment). A short notation for "the problem P in the fragment f", will be f-P, e.g. $\mathcal{F}$-DEDUCTION?.

From deduction of rules to deduction of facts

The general deduction problem (which asks if a given rule can be deduced) can be reformulated as the deduction problem of a fact instead of a rule, as pointed out below.

Definition 3 (Substitution) *A* substitution *is a mapping from a set V of variables to a set T of terms. Let $\sigma : V \rightarrow T$ be a substitution, and F be a fact. We note $\sigma(F)$ the fact obtained by replacing, for each variable $x \in V \cap \mathit{terms}(F)$, each occurrence of x in F by $\sigma(x)$. If $R = (H, C)$ is a rule, we note $\sigma(R) = (\sigma(H), \sigma(C))$.*

A *safe substitution* is a bijection from V to a set of new variables (*i.e.* that does not appear in the formulas involved in the reasoning). A *freezing substitution* is a bijection from V to a set of new constants.

Property 2 *Let $\mathcal{R}$ be a set of rules, and $R = (H, C)$ be a standard rule. Then $\mathcal{R} \models R$ iff $\sigma(H), \mathcal{R} \models \sigma(C)$ where σ is a freezing substitution of $\mathit{vars}(H)$.*

DEDUCTION? can be recast, without loss of generality, as "is a fact Q a semantic consequence of a standard KB?".

The Case of Constraints

Finally, to further bound the our investigation, and without loss of generality, let us consider the following properties:

Property 3 (Constraints) *Let us consider a standard KB $\mathcal{K}$ containing a set $\mathcal{C}$ of constraints. Since no constraint is valid, $\mathcal{K}$ is valid iff $\mathcal{C} = \emptyset$ and $\mathcal{K} \setminus \mathcal{C}$ is valid. $\mathcal{K}$ is unsatisfiable iff $\mathcal{K} \setminus \mathcal{C}$ is unsatisfiable or there is $\neg H \in \mathcal{C}$ such that $\mathcal{K} \setminus \mathcal{C} \models H$. Let Q be a fact, then $\mathcal{K} \models Q$ iff $\mathcal{K}$ is unsatisfiable or $\mathcal{K} \setminus \mathcal{C} \models Q$.*

Semantic properties of a standard KB with constraints can thus be reformulated in terms of those of a standard KB without constraints. The next section is devoted to solving problems in $\mathcal{FRE}$ with a forward chaining mechanism.

Forward Chaining in $\mathcal{FRE}$

We will first concentrate on the $\mathcal{F}$ fragment, which provides the basic mechanism for reasoning, namely homomorphism.Then we focus on reasoning with standard and equality rules, in a forward chaining framework.

Reasoning in $\mathcal{F}$ and $\mathcal{FC}$

The problems $\mathcal{F}$-SATISFIABILITY? and $\mathcal{F}$-VALIDITY? are trivial: the only valid standard fact is the empty fact $\emptyset$ and the only unsatisfiable standard fact is the absurd fact $\{\Box\}$.

Note that the computation of validity and satisfiability of facts was already included when computing the standard equivalent KB, but this computation took only linear time.

Solving $\mathcal{F}$-DEDUCTION?, is based on a particular kind of substitution called a homomorphism (it is indeed a homomorphism, if we see facts as algebraic structures).

Definition 4 (Homomorphism) *Let F and Q be two facts. A homomorphism from Q to F is a substitution such that $\sigma(Q) \subseteq F$.*

Theorem 1 ($\mathcal{F}$-DEDUCTION?) *Let F and Q be two standard facts. Then $F \models Q$ iff $F = \{\Box\}$ or there is a homomorphism from Q to F.*

Proof: This result is basically from (CM92) (soundness and completeness of the simple CGs homomorphism). □

Deciding whether there is a homomorphism from Q to F is NP-complete. It becomes polynomial when Q has a structure decomposable into trees. More precisely, let FOL($\exists, \wedge$) be the fragment of FOL composed of existentially closed, positive and conjunctive formulas, i.e. formulas corresponding to facts. Let k be a fixed positive integer. If in all problem instances, Q is in the k-variable fragment of FOL($\exists, \wedge$), or equivalently if it has a treewidth less than k when seen as a graph, then homomorphism checking is polynomial (KV00). Similarly, if Q is a k-guarded formula, or equivalently if has a hypertreewidth bounded by k when seen as a hypergraph, then homomorphism is polynomial too (GLS01).

From PROP. 3 and the complexity of the decision problems in $\mathcal{F}$, it follows that $\mathcal{FC}$-VALIDITY? can be decided in linear time, $\mathcal{FC}$-SATISFIABILITY? is co-NP-complete and $\mathcal{FC}$-DEDUCTION? is NP-complete. Note that $\mathcal{FC}$-SATISFIABILITY? and $\mathcal{FC}$-DEDUCTION? become polynomial when we impose the same restrictions on constraints that the above-mentioned restrictions on Q.

Reasoning in $\mathcal{FREC}$

Let us now consider reasoning in general standard KBs (thanks to PROP. 3, we can restrict that study to $\mathcal{FRE}$).

Property 4 (Validity of standard and equality rules) *There is no valid equality rule. A standard rule (H, C) is valid iff $\sigma(H) \models \sigma(C)$ where σ is a freezing substitution of vars(H) (this is indeed a particular case of* PROP. *2).*

As a consequence, $\mathcal{FR}$-VALIDITY? is an NP-complete problem (and so is $\mathcal{FR}$-VALIDITY?). It can be solved in linear time when there is no standard rule in the KB.

Definition 5 (Application of a rule) *Let F be a standard fact and $R = (H, C)$ be a standard rule or an equality rule. Then R is* applicable *to F iff there is a homomorphism σ from H to F. The* application of R on F according to σ *produces a standard fact $\alpha(F, R, \sigma)$ (called an* immediate derivation *of F) built as follows: if R is a standard rule, then $\alpha(F, R, \sigma) = F \cup \sigma(\sigma'(C))$ where σ' is a safe substitution of vars$(C) \setminus fr(R)$; and if R is an equality rule, then either a variable (say t) of the equality $t \doteq t'$ is mapped by σ to a variable, or none is. In the first case, $\alpha(F, R, \sigma) = \sigma'(F)$, where σ' is the substitution from $\{\sigma(t)\}$ that maps $\sigma(t)$ to the constant a if $t' = a$, and to $\sigma(t')$ otherwise. In the second case, if $\sigma(t) = \sigma(t')$ or $\sigma(t) = t'$, then $\alpha(F, R, \sigma) = F$, otherwise $\alpha(F, R, \sigma) = \{\Box\}$.*

Definition 6 (Derivation) *Let F and F' be two standard facts, and $\mathcal{R}$ be a set of standard and equality rules. F' is called an $\mathcal{R}$-*derivation *of F if there is a finite sequence (called the* derivation sequence*) $F_0 = F, F_1, \ldots, F_k = F'$ of standard facts such that $\forall 1 \leq i \leq k$, there is a rule $R = (H, C) \in \mathcal{R}$ and a homomorphism σ from H to F_{i-1} with $F_i = \alpha(F_{i-1}, R, \sigma)$.*

Since standard rules cannot yield unsatisfiability:

Property 5 ($\mathcal{FRE}$-SATISFIABILITY?) *A $\mathcal{FR}$ KB is unsatisfiable if and only if $F = \{\Box\}$. A $\mathcal{FRE}$ KB $(F, \mathcal{R}, \mathcal{E})$ is unsatisfiable iff $\{\Box\}$ is an $(\mathcal{R} \cup \mathcal{E})$-derivation of F.*

Theorem 2 ($\mathcal{FRE}$-DEDUCTION?) *Let $K = (F, \mathcal{R}, \mathcal{E})$ be a $\mathcal{FRE}$ KB, and Q be a standard fact. Then $K \models Q$ iff there is a homomorphism from Q or $\{\Box\}$ to a $(\mathcal{R} \cup \mathcal{E})$-derivation of F.*

If Q is a semantic consequence of the KB $K = (F, \mathcal{R}, \mathcal{E})$, then a breadth-first exploration of all possible rule applications will necessarily generate either the absurd fact (when we have deduced that two constants are equal, which is absurd due to the unique name assumption) or a standard fact containing an answer to Q. No sound and complete algorithm, however, is ensured to stop on all negative instances of the problem: $\mathcal{FRE}$-DEDUCTION? is a semi-decidable problem (see in particular (BM02)).

As a consequence, $\mathcal{FRE}$-UNSATISFIABILITY? is also semi-decidable, and thus $\mathcal{FRE}$-SATISFIABILITY? is undecidable. Note that, since the application of an equality rule either merges two variables or replaces a variable by a constant, an $\mathcal{E}$-derivation has a length less than $|terms(F)|$. Thus both $\mathcal{FE}$-DEDUCTION? and $\mathcal{FE}$-SATISFIABILITY? are NP-complete problems (and so are $\mathcal{FEC}$-DEDUCTION? and $\mathcal{FEC}$-SATISFIABILITY?).

Decidable subclasses of $\mathcal{FRE}$-DEDUCTION?

Since the satisfiability and deduction problems are no longer decidable as soon as standard rules are involved, finding decidable cases, as expressive as possible, is extremely important. We recall below definitions and results from (BM02), which are adapted to include equality rules (see the section about related works for more details). A set of rules is called a *finite expansion set* if it is guaranteed for any fact that after a finite number of rule applications, all further rule applications will become redundant. More precisely:

Definition 7 (Finite expansion set and full derivation) *A set of standard or equality rules $\mathcal{R}$ is called a* finite expansion set *iff for any standard fact F, there is a $\mathcal{R}$-derivation F' of F s.t. for every rule $R = (H, C) \in \mathcal{R}$, for every homomorphism σ from H to F', there is a homomorphism from $\alpha(F', R, \sigma)$ to F' (called a* full $\mathcal{R}$-derivation of F*).*

If $\mathcal{R} \cup \mathcal{E}$ is a finite expansion set, then a forward chaining algorithm that checks whether a rule has effectively added non redundant information and stops when all rule applications produced redundancy is still complete, and is ensured to stop in finite time. This algorithm effectively produces a standard fact that contains all information that can be deduced from F. However, finite expansion sets are an abstract characterization of such a behavior, that does not help us to predict that forward chaining will effectively stop (note, moreover, that all decidable subclasses of the problem do not rely upon finite expansion sets). As a concrete example of finite expansion sets, let us cite the *range restricted rules*, used in particular in positive Datalog, that do not generate any new variable (i.e. standard rules (H, C) such that *vars*$(C) \subseteq$ *vars*(H)) and *disconnected rules* (standard rules

R such that $fr(R) = \emptyset$). As stated above, a set of equality rules is also a finite expansion set. Note that the union of two finite expansion sets is not necessarily a finite expansion set ((BM02) shows that any Turing machine can be encoded by a set of range restricted and disconnected standard rules).

(BM02) proves that if standard rules are restricted to either range-restricted rules or disconnected rules, $\mathcal{FR}$-DEDUCTION? is NP-complete (provided that the arity of predicates is bounded by a constant). We have shown here that the same result holds for equality rules. It follows that, assuming the same restrictions, $\mathcal{FREC}$-DEDUCTION? and $\mathcal{FREC}$-SATISFIABILITY? are also NP-complete.

The forward chaining paradigm is used to define finite expansion sets and characterize some decidable subclasses of $\mathcal{FREC}$-DEDUCTION?. The backward chaining paradigm that we present now allows for a new characterization of decidable subclasses, namely *finite unification sets*.

Backward Chaining in $\mathcal{FR}$

A backward chaining mechanism is generally based upon a unification operation, that matches part of a current goal with a rule conclusion. This mechanism is typically used in logic programming, where rules have a conclusion restricted to one literal. Since standard rules have a more complex conclusion, the associated unification operation is also more complex. It relies on the *piece* notion (which stems from a graph vision of rules (SM96)).

Definition 8 (Cut points, Pieces) *Let F be a fact and $T \subseteq terms(F)$. A* piece of F according to T *is a subset of F recursively defined as follows: two atoms p and q are in the same piece iff either $(vars(p) \cap vars(q)) \setminus T \neq \emptyset$; or there is an atom $r \in F$ such that p and r are in the same piece and r and q are in the same piece. The* cut points *of a rule $R = (H, C)$ is the set of terms $cutp(R) = fr(R) \cup consts(C)$. A* piece *of R is a piece of C according to $cutp(R)$.*

A piece notion represents a "unit" of knowledge brought by a rule application. As expressed below, a rule R can be decomposed into an equivalent set of rules with exactly one piece in conclusion.

Property 6 *The conclusion of standard rule $R = (H, C)$ can be partitioned into k pieces $C_1, \ldots, C_k$. R is semantically equivalent to the set $\{(H, C_i)\}_{1 \leq i \leq k}$.*

Rewritings in $\mathcal{FR}$

The definitions and results concerning rewritings presented hereafter are a reformulation of those of (SM96).

Definition 9 (Unifier) *Let Q be a standard fact and $R = (H, C)$ be a standard rule. A* unifier *of Q with R is a tuple $\mu = (T_Q, Q', \sigma_Q, \sigma_C)$ where:*

- *T_Q is a subset of $terms(Q)$, and Q' is a fact composed of all atoms in one or more pieces of Q w.r.t. T_Q;*
- *σ_Q is a substitution from the variables of T_Q to $consts(C) \cup T_Q$ and σ_C is a substitution from $fr(R)$ to $consts(C) \cup T_Q$;*
- *there is a homomorphism σ from $\sigma_Q(Q')$ to $\sigma_C(C)$ s.t. for all $t \in T_Q$, there is $t' \in cutp(R)$ with $\sigma(\sigma_Q(t)) = \sigma_C(t')$.*

Definition 10 (Rewriting of a fact) *Let Q be a standard fact, $R = (H, C)$ be a standard rule, and $\mu = (T_Q, Q', \sigma_Q, \sigma_C)$ be a unifier of Q with R. The* rewriting of Q according to R and μ *produces a fact $\beta(Q, R, \mu) = \sigma'(\sigma_C(H)) \cup \sigma_Q(Q \setminus Q')$, where σ' is a safe substitution from $vars(H) \setminus fr(R)$.*

The two following lemmas state the precise relationships between an immediate derivation and a rewriting.

Lemma 1 *Let μ be a unifier of a standard fact Q with a standard rule $R = (H, C)$. Then there is a homomorphism σ from H to $F = \beta(Q, R, \mu)$ such that there is a homomorphism from Q to $\alpha(F, R, \sigma)$.*

Lemma 2 *Let us consider a standard rule R, and F, F' and Q three standard facts such that $F' = \alpha(F, R, \sigma)$ and there is a homomorphism from Q to F' that is not a homomorphism from Q to F. Then there is a unifier μ of Q with R such that there is a homomorphism from $\beta(Q, R, \mu)$ to F.*

These lemmas are used to prove inductively (on the length of the derivation / rewriting) that a derivation sequence and a rewriting sequence encode the same reasoning in $\mathcal{FR}$.

Definition 11 (Rewriting sequence) *Let Q and Q' be two standard facts, and $\mathcal{R}$ be a set of standard rules. We say that Q' is a* $\mathcal{R}$-rewriting of Q *iff there is a finite sequence (called the* rewriting sequence*) $Q_0 = Q, Q_1, \ldots, Q_k = Q'$ such that $\forall 1 \leq i \leq k$, there is a rule $R = (H, C) \in \mathcal{R}$ and a unifier μ of Q_{i-1} with R such that $Q_i = \beta(Q_{i-1}, R, \mu)$.*

Property 7 *Let F and Q be two facts, and $\mathcal{R}$ be a set of standard rules. Then there is a $\mathcal{R}$-rewriting Q' of Q and a homomorphism from Q' to F iff there is a $\mathcal{R}$-derivation F' of F and a homomorphism of Q to F'.*

Then , as a direct consequence of the previous property:

Theorem 3 (Backward Chaining in $\mathcal{FR}$) *Let $K = (F, \mathcal{R})$ be a $\mathcal{FR}$ KB, and Q be a standard fact. Then $F, \mathcal{R} \models Q$ iff there is a $\mathcal{R}$-rewriting Q' of Q such that there is a homomorphism from Q' to F.*

Note that using backward chaining to check the unsatisfiability of a KB $K = (F, \mathcal{R}, \mathcal{C})$ requires to check whether the hypothesis of a constraint can be deduced from $(F, \mathcal{R})$.

Though we could define a unifier that takes equality rules into account, both space requirements as well as the inefficiency of the obtained algorithm led us to take only standard rules into account in this backward chaining framework.

Finite unification sets

Finite expansion sets are used to ensure that all information deducible from a fact in forward chaining can be encoded in a finite standard fact computed in finite time. The *finite unification sets* we present hereafter are used to ensure that only a finite number of rewritings will be necessary in backward chaining.

Definition 12 (Finite unification sets) *A set of standard rules $\mathcal{R}$ is said a finite unification set (f.u.s.) iff for every fact Q, there is a finite set $\mathcal{Q}$ of $\mathcal{R}$-rewritings of Q such that for any $Q' \in \mathcal{Q}$, for any rule $R \in \mathcal{R}$, for any unifier μ of Q' with R, there is a homomorphism from a fact in $\mathcal{Q}$ to $\beta(Q', R, \mu)$. We say that $\mathcal{Q}$ is a full $\mathcal{R}$-rewriting set of Q.*

If the set of standard rules involved is a f.u.s., then the backward chaining algorithm that does not rewrite facts more specific than those that have already been explored is still complete and is ensured to stop in finite time. Similarly to f.e.s., f.u.s. provide an abstract characterization, that should be instantiated with concrete examples. We introduce in this paper two kinds of f.u.s.: the *atomic hypothesis* rules, and the *domain restricted* rules.

Definition 13 (Atomic hypothesis rules) *A standard rule $R = (H, C)$ is called an* atomic hypothesis rule *(a.h.) if H contains only one (standard) atom.*

Property 8 *A set of atomic hypothesis rules is a f.u.s.*

Proof: See that for any standard fact Q, if $R = (H, C)$ is a.h., and $\mu = (T_Q, Q', \sigma_Q, \sigma_C)$ is a unifier of Q with R, then $|Q| \geq |\beta(Q, R, \mu)|$ (since the rewriting removes the non empty Q' and adds a specialization of the unique atom in H). There is a bounded N number of facts (up to a variable renaming) of size $\leq |Q|$ built from the bounded number of constants and predicates appearing in the KB. Thus if $\mathcal{R}$ contains only a.h. rules, the number of $\mathcal{R}$-rewritings of Q is bounded by N. □

Definition 14 (Domain restricted rules) *A standard rule $R = (H, C)$ is called a domain restricted (d.r.) rule if each atom of C contains all or none of the variables of H.*

Note that d.r. rules strictly generalize the class of disconnected rules, since d.r. rules with $fr((H, C)) = vars(H) \neq \emptyset$ are not disconnected.

Property 9 *A set of domain restricted rules is a f.u.s.*

Proof: Let us call k-limited fact, a fact Q such each piece P of Q according to $consts(Q)$ is such as $|vars(P)| \leq k$. There is a finite number N of facts with at most k variables (up to a variable renaming), and a bounded number of constants and predicates. Thus, a k-limited fact containing more than N pieces contains equivalent pieces which can be removed to obtain a fact with at most N pieces. We observe that if a rewriting according to a d.r. rule produces a new variable, then it also produces a new piece which is the only piece containing this variable. Indeed, either the unifier concerns an atom that contains all variables of H and it does not generate new variables, or it creates a new piece. Thus if $\mathcal{R}$ contains only d.r. rules, the number of $\mathcal{R}$-rewritings of Q without duplicate pieces is bounded by N. □

Since the number N involved in the previous proofs is polynomial w.r.t. the size of the knowledge base (provided that the arity of all predicates is bounded by a constant), then the restrictions of $\mathcal{FR}$-SATISFIABILITY? and $\mathcal{FR}$-DEDUCTION? to knowledge bases containing either a.h. or d.r. rules are NP-complete problems.

The graph of rules dependencies

Both f.e.s. and f.u.s. allows to characterize decidable subsets of the rule fragment of FOL by checking independently on each rule if they verify some properties. We now present the graph of rules dependencies as a mean to explore new decidable subsets by studying the structure of the interactions between these rules.

Let us rephrase lemma 1. Its converse means that if R is a standard rule and there is no unifier of a fact Q with R, then the set of homomorphisms from Q to any fact F and the set of homomorphisms from Q to $\alpha(F, R, \sigma)$ are the same. In other words, there is no need to check the applicability of a rule (H, C) at a point of the forward chaining algorithm when there is no unifier of H with a rule that produced new information during a previous step of the algorithm. Moreover, as shown in (Bag04), an off-line computation of unifiers allows us not only to reduce the number of applicability checks during the execution of forward chaining, but also to reduce the search space for the remaining ones.

We define the graph of rules dependencies as a structure that encodes all necessary information to take advantage of the above-mentioned property in forward chaining. It allows us to compute a smaller number of rules application checks, and to compute these checks more efficiently. In this paper, however, we focus the structural properties of this graph leading to new decidability results in $\mathcal{FRESC}$.

F.e.s., f.u.s., and the graph of rule dependencies

Definition 15 (GRD) *Let $\mathcal{R}$ be a set of standard rules. The* graph of rules dependencies *of $\mathcal{R}$, noted GRD($\mathcal{R}$) is a directed labeled graph whose vertices are the rules of $\mathcal{R}$ and whose arcs are the couples (R, R') such that there is a unifier of the hypothesis of R' with R.*

Property 10 *Let $\mathcal{R}$ be a set of standard rules such that GRD($\mathcal{R}$) contains no circuit. Then $\mathcal{R}$ is a both a finite expansion set and a finite unification set.*

Proof: The f.e.s. part of that property has been proven in (Bag04). The f.u.s. is an immediate consequence of a property exposed in (BS06): if there is an unifier μ' of R' with $Q' = \beta(Q, R, \mu)$ such that $\beta(Q', R', \mu')$ is not a rewriting of Q according to R', then there is an unifier of the hypothesis of R with R'. In other terms, to rewrite $\beta(Q, R, \mu)$, we only have to explore the predecessors of R in the GRD. □

Note that a loop (a self-unifiable rule) in the GRD is considered as a circuit, and is sufficient to yield undecidability. Indeed, it has been shown in (Bag01) that a KB containing a single rule can encode a universal Turing machine.

Finite expansion sets and finite unification sets relied upon the structural properties of the rules themselves to ensure decidability. PROP. 10 relies upon the structure of possible interactions between all rules appearing in the KB. The next theorem presents a generalization of both approaches (and is proven in (Bag04) for the f.e.s. part, while the f.u.s. part is a similar extension of PROP. 10):

Theorem 4 *Let $K = (F, \mathcal{R})$ be a KB such that all strongly connected components of the graph GRD($\mathcal{R}$) are finite expansion sets (resp. finite unification sets). Then $\mathcal{R}$ is a finite expansion set (resp. finite unification set).*

Note that a strongly connected component restricted to a single rule, and without loop, is both a f.e.s. and a f.u.s.

F.e.s and f.u.s respectively rely upon forward and backward chaining to achieve finite reasoning mechanisms. Let us now present a combined algorithm that is ensured to stop on more general subclasses of the deduction problem.

Property 11 *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two sets of standard rules such that there is no arc from a rule of $\mathcal{R}_2$ to a rule of $\mathcal{R}_1$ in GRD$(\mathcal{R}_1 \cup \mathcal{R}_2)$. Let F and Q be two standard facts. Then $F, \mathcal{R}_1, \mathcal{R}_2 \models Q$ iff there is a standard fact F' such that $F, \mathcal{R}_1 \models F'$ and $F', \mathcal{R}_2 \models Q$.*

Definition 16 (Finitely combined sets) *A partition $(\mathcal{R}_1, \mathcal{R}_2)$ of standard rules is said to be* finitely combined *if $\mathcal{R}_1$ is a f.e.s., $\mathcal{R}_2$ is a f.u.s., and there is no arc from a rule of $\mathcal{R}_2$ to a rule of $\mathcal{R}_1$ in GRD$(\mathcal{R}_1 \cup \mathcal{R}_2)$. A* finitely combined set *is a set of standard rules that admits a finitely combined partition.*

Theorem 5 ($\mathcal{FR}$ combined scheme) *Let $(\mathcal{R}_1, \mathcal{R}_2)$ be a finitely combined partition of standard rules, and F and Q be two standard facts. Then $F, \mathcal{R}_1, \mathcal{R}_2 \models Q$ iff there is a homomorphism from Q' to F', where F' is a full $\mathcal{R}_1$-derivation of F and Q' belongs to a full $\mathcal{R}_2$-rewriting set of Q.*

As a consequence, $\mathcal{FR}$-DEDUCTION? is decidable when standard rules are restricted to finitely combined sets.

Adding constraints and equalities

It follows from PROP. 3, that $\mathcal{FRC}$-SATISFIABILITY? and $\mathcal{FRC}$-DEDUCTION? are decidable when the problems are restricted to finitely combined sets of standard rules.

Let us now consider a mixed FC/BC scheme that takes equality rules into account in its forward chaining phase.

Theorem 6 ($\mathcal{FRE}$ combined scheme) *Let $(\mathcal{R}_1, \mathcal{R}_2)$ be a finitely combined partition of standard rules, $\mathcal{E}$ be a set of equality rules such that for every equality rule $(H, \{t \doteq t'\}) \in \mathcal{E}$, H only unifies with range restricted rules of $\mathcal{R}_1$, and F and Q be two standard facts. Then $F, \mathcal{R}_1, \mathcal{R}_2, \mathcal{E} \models Q$ iff there is a homomorphism from Q' to F', where F' is a full $\mathcal{R}_1 \cup \mathcal{E}$-derivation of F and Q' belong to a full $\mathcal{R}_2$-rewriting set of Q.*

In that case, $\mathcal{FRE}$-DEDUCTION? and $\mathcal{FRE}$-SATISFIABILITY? are decidable, and it follows, thanks to PROP. 3, that $\mathcal{FREC}$-SATISFIABLITY? and $\mathcal{FREC}$-DEDUCTION? are also decidable.

Relationships to other formalisms

This section details the relationships between our rules and other notions in other formalisms. It indicates how our results extend known results and explains what the potential benefit of our results could be for these related formalisms.

Clauses and rules

A *clause* (without function symbols) is a logical formula of form $\forall x_1 \dots \forall x_k (N^\vee \vee P^\vee)$ where $P^\vee$ is a disjunction of atoms, $N^\vee$ is a disjunction of negated atoms, and $x_1, \dots, x_k$ are the variables appearing in $P^\vee$ and $N^\vee$. Such a formula is trivially equivalent to the formula: $\forall n_1 \dots \forall n_k (\overline{N}^\wedge \rightarrow (\forall p_1 \dots p_q P^\vee))$ where $\overline{N}^\wedge$ is the conjunction of the atoms appearing in $N\vee$, the n_i are the x_q appearing in $N^\vee$ and the p_j are the x_q appearing in $P^\vee$ but not in $N^\vee$. This rewriting points out the difference between clauses and rules: they have the same "hypothesis", but the conclusion of our rules is an existentially quantified conjunction, while the "conclusion" of a clause is an universally quantified disjunction. Note that a formula $F = \forall x_1 \dots \forall x_q (H^\wedge \rightarrow C)$ is both a clause and a rule when C contains only 0 or 1 positive atoms, and all variables of C appear in $H^\wedge$. The intersection of rules and clauses can thus be characterizes as "*range restricted Horn clauses*".

Another interesting subset of rules, that we name *atomic conclusion rules*, is restricted to rules having at most one atom in their conclusion. This subset has the same expressivity as the general rule fragment: any standard rule $R = (H, C)$ can be encoded by an equivalent set of atomic conclusion rules $\{(H, A = \{R(t_1, \dots, t_k)\}), (A, \{A_q\})\}_{A_q \in C}$ where the predicate R is associated to the rule R and the t_i are the terms of C. An immediate question arises: why study these general rules, since unifiers with atomic conclusion rules are *(i)* well known, and *(ii)* much simpler. Our answer is that the GRD of these equivalent formulas contains more arcs, encoding "fake" unifications, and thus leads to more restricted decidable fragments, as shown by the following example. Consider the standard rule $R = (H, \{p(x, y), q(y, z)\})$ where $(R) = \{x\}$ and the fact $Q = \{s(x, y), q(y, z)\}$. There is no unifier, according to our definition, of Q with R. Let us consider now the equivalent set of rules $\{R_1, R_2, R_3\}$ where $R_1 = (H, \{R(x, y, z)\})$, $R_2 = (\{R(x, y, z)\}, \{p(x, y)\})$ and $R_3 = (\{R(x, y, z)\}, \{q(y, z)\})$. In the GRD, there is an arc from R_1 to R_2 and R_3, and there exists a unifier of Q with R_3. Indeed, we have lost the information that R_2 and R_1 cannot be applied independently.

Tuple-Generating Dependencies in Databases

The rules studied in this paper correspond to the logical translation of generalized dependencies in databases, namely Tuple Generating Dependencies (TGDs) and Equality Generating Dependencies (EGDs), which provide of uniform way of expressing most of the database dependencies (AHV95). More precisely, the logical translation of a TGD is a standard rule, and the logical translation of an EGD is an equality rule. The only notable difference is that TGDs and EGDs have no constants.

A fundamental problem for these generalized dependencies is the implication problem. This problem takes a set of generalized dependencies S and a generalized dependency t as input, and asks if every database instance that satisfies S also satisfies t, i.e., in logical terms, if $S \models t$. It is processed by the so-called *chase* procedure, which was proven sound and complete with respect to semantics of dependencies in (BV84). This procedure can be identified with our forward chaining mechanism when t is a TGD. The backward chaining can be seen as an alternative to the chase (as proposed in (Cou03)). The rule dependency graph studied in this paper can be used to improve the chase. It would be interesting to study the properties of the dependency graph depending on the specific kinds of dependencies taken into account and compare the decidability results obtained to those already known.

Conceptual Graphs

Conceptual graphs have been presented in (Sow76) as a graphical interface for databases. Indeed, the formulas we have studied here admit a very intuitive graph encoding, where terms are labeled vertices of the graph and atoms are represented by labeled hyperarcs whose ends are the argument of the predicate. Semantic consequence between facts (called *simple CGs*) is computed by a *graph homomorphism*, hence the name of our elementary operation. We have developed most results presented in this paper under this graph paradigm, and the family of conceptual graphs languages presented in (BM02) is closely related to the family presented here. Indeed, the basic CG language $\mathcal{SG}$ encodes standard facts (without equality) and standard rules of form $(\{p(x_1, \ldots, x_n\}, \{p'(x_1, \ldots, x_n\})$ that are used to encode *hierarchies of types*. The language $\mathcal{SGC}$ further adds constraints, that can be positive or negative. Negative constraints correspond to the constraints presented here, while the essentially non-monotonic positive constraints cannot be represented by rules. Finally, the language $\mathcal{SRC}$ also consider rules that are equivalent to the standard rules presented here. The rule fragment of logic $\mathcal{FRC}$ can thus be identified with the CG language $\mathcal{SRC}^-$ restricted to negative constraints. Note that $\mathcal{FREC}$ can also be encoded in $\mathcal{SRC}^-$, a language that does not include the unique name assumption.

Finally, note that the main difference between these CG languages and the logic of rules is that CGs consider rules encoding type hierarchies (typing rules) as different objects, with a algorithmic specific role. By example, our definition of homomorphism becomes: "a substitution σ such that for every atom $p(t_1, \ldots, t_k) \in Q$, there is an atom $p'(t'_1, \ldots, t'_k) \in F$ with $p' \leq p$ in the type hierarchy and $\forall 1 \leq i \leq k$, $\sigma(t_i) = t'_i$". This allows for a compilation of type hierarchies, optimizing the search for substitutions. Type hierarchies are integrated in the same way in unifiers, effectively removing from the GRD circuits formed of typing rules, that do not correspond to an increase of the complexity of the deduction problem. Similarly, *forbidden types* encode constraints of form $\neg\{p_1(x), \ldots, p_k(x)\}$. They are used to reduce the number of unifiers of a fact with a rule.

Description logics

We intend to use our results in the rules fragment of FOL to study the deduction problem in logic-based knowledge representation languages. Our general approach can be stated as follows: consider a language $\mathcal{L}$, such that there exists a transformation $\tau_{\mathcal{L}}$ from the expressions of $\mathcal{L}$ to standard facts and constraints, and a fixed set of standard and equality rules $\mathcal{R}_{\mathcal{L}}$ such that for all expressions $L, L_1, \ldots, L_k$ of $\mathcal{L}$, $L_1, \ldots, L_k \models_{\mathcal{L}} L$ iff $\tau_{\mathcal{L}}(L_1), \ldots, \tau_{\mathcal{L}}(L_k), \mathcal{R}_{\mathcal{L}} \models \tau_{\mathcal{L}}(L)$. Assuming we can obtain such an encoding of the language $\mathcal{L}$, we can study the properties of that language by studying the properties of *GRD*$(\mathcal{R}_{\mathcal{L}})$. Sometimes such an encoding cannot be obtained, but a weaker form may still be found, when there exists a transformation $\tau'_{\mathcal{L}}$ from the expressions of $\mathcal{L}$ to rules such that $L_1, \ldots, L_k \models_{\mathcal{L}} L$ iff $\tau'_{\mathcal{L}}(L_1), \ldots, \tau_{\mathcal{L}}(L_k), \models \tau'_{\mathcal{L}}(L)$. In that case, the GRD does not provide us with properties of the language, but of the specific instance we have transformed.

This latter method was recently applied in (BLMS08) to study particular subsets of description logics (BCM$^+$03), namely DL-Lites (CGL$^+$05; CGL$^+$07). In particular, the GRD obtained was sufficient to prove the *FOL-reducibility* of a specific DL-Lite. That FOL-reducibility property can be stated as follows: there exists a transformation ρ that maps each standard fact to a *finite* set of database conjunctive queries and a *linear* transformation δ that maps each KB to a relational database instance such that, for any KB K, for any standard fact Q, $K \models Q$ if and only if $\delta(K)$ contains an answer to one of the queries in $\rho(Q)$. Since homomorphisms in $\mathcal{F}$ are equivalent to the answering of positive, conjunctive queries in database, it follows that if the set of rules encoding the semantics of our problem is always a finite unification set (or, more generally, a finitely combined set whose f.e.s. subset of rule generates a full graph in linear time), then the deduction problem in the studied language is FOL-reducible. In order to study more expressive DL-Lites, equality rules were required: they will allow, by example, to encode *functional roles* by adding the rule $(\{R(x, y), R(x, z)\}, \{y \doteq z\})$ for each functional rule R.

References

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

J.-F. Baget. *Représenter des connaissances et raisonner avec des hypergraphes: de la projection à la dérivation sous contraintes*. PhD thesis, Université Montpellier II, Nov. 2001.

J.-F. Baget. Improving the forward chaining algorithm for conceptual graphs rules. In *KR*, pages 407–414, 2004.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

J.-F. Baget, M. Leclère, M.-L. Mugnier, and E. Salvat. Characterizing FOL-Reducibility of Some DL-Lites by Structural Properties of $\mathcal{SR}$ Rules. In *Proceedings of DL'08*, 2008. To appear.

J.-F. Baget and M.-L. Mugnier. The Complexity of Rules and Constraints. *JAIR*, 16:425–465, 2002.

J.-F. Baget and E. Salvat. Rules dependencies in backward chaining of conceptual graphs rules. In *ICCS*, pages 102–116, 2006.

C. Beeri and M.Y. Vardi. A proof procedure for data dependencies. *Journal of the ACM*, 31(4):718–741, 1984.

D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Dl-lite: Tractable description logics for ontologies. In *AAAI*, pages 602–607, 2005.

D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *l-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

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

Stéphane Coulondre. A top-down proof procedure for generalized data dependencies. *Acta Inf.*, 39(1):1–29, 2003.

G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions: A survey. *MFCS'01*, 2136:37–57, 2001.

P. G. Kolaitis and M. Y. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

E. Salvat and M.-L. Mugnier. Sound and Complete Forward and Backward Chainings of Graph Rules. In *Proc. of ICCS'96*, volume 1115 of *LNAI*, pages 248–262. Springer, 1996.

J. F. Sowa. Conceptual Graphs. *IBM Journal of Research and Development*, 1976.