DL-$\mathcal{SR}$, a Lite DL Extended with Expressive Rules: Preliminary Results

Jean-François Baget[1,2], Michel Leclère[2],
Marie-Laure Mugnier[2], and Éric Salvat[3]

[1] INRIA, 2004 route des lucioles, 06902 Sophia Antipolis - France
jean-francois.baget@inria.fr
[2] LIRMM (CNRS & U. Montpellier II), 161 rue Ada, 34392 Montpellier - France
mugnier@lirmm.fr, leclere@lirmm.fr
[3] IMERIR, Avenue Pascot, 66004 Perpignan - France
salvat@imerir.com

Abstract. Simple conceptual graphs can be seen as a very basic description logic, allowing however for answering conjunctive queries. In the first part of this paper, we translate some results obtained for conceptual graph rules of form "if A then B" into an equivalent DL-based formalism. Then we show that, our algorithms can automatically decide in some cases whether a given DL has the FOL-reducibility property, provided that the semantics of its constructors can be expressed by rules.

1 Introduction

The families of languages known as Description Logics and Conceptual Graphs answer criticisms addressed to their common ancestor, semantic networks, by distinguishing between factual (existential) and ontological (universal) knowledge (the TBox and ABox in DLs, and the support and graphs in CGs), as well as by providing these languages with model-theoretical semantics.

These shallow similarities must not hide fundamental differences: while DLs offer expressive TBox constructors, conjunctive queries generally have an important computational cost [2]. On the other hand, while the most simple CGs can answer such queries, their ontological component offers little more than inclusion of atomic types. Relationships between DLs and CGs have been precisely pointed out (*e.g.* [1] identifies their intersection as the weakly expressive $\mathcal{ELIRO}^1$).

The recent DL-Lite approach [8, 9] (reducing the TBox expressivity to answer conjunctive queries) meets our approach, which consists of adding more expressive ontological constructs to simple CGs (*e.g.* type conjunction [4], disjointness of types [10], atomic negation [12]). The goal of this paper is to show that the theoretical tools we have developed for CG rules can be used for lite DLs.

Sect. 2 presents DL-$\mathcal{SG}$, a lite DL inspired by simple CGs (called $\mathcal{SG}$ in [6]), allowing only atomic inclusion assertions, but with variables in the ABox. sect. 3 presents DL-$\mathcal{SR}$, an extension of DL-$\mathcal{SG}$ with rules (called $\mathcal{SR}$ in [6]), allowing assertions of form "if *hypothesis* H then *conclusion* C". Decidable subclasses of DL-$\mathcal{SR}$ are detailed in sect. 4. Finally, in sect. 5, we apply these results to study the FOL-reducibility of the DLR-Lite$_{\mathcal{R}}$ language.

2 DL-$\mathcal{SG}$: conjunctive queries, but lite TBox ...

2.1 DL-$\mathcal{SG}$ syntax and semantics

A *vocabulary* $\mathcal{V}$ is the union of pairwise disjoint sets $\mathcal{N}$, $\mathcal{R}_1$, ..., $\mathcal{R}_k$ where $\mathcal{N}$ contains *individual names* and $\mathcal{R}_i$ contains *role names of arity* i. Role names of arity 1 are called *concept names* ($\mathcal{C} = \mathcal{R}_1$). The set of concept names contains a distinguished element, $\bot$ (the absurd concept name). We also consider a set $\mathcal{X}$ of *variables*, disjoint from *names* of $\mathcal{V}$. The set $\mathcal{T} = \mathcal{N} \cup \mathcal{X}$ is the set of *terms*.

Definition 1 (Primitive TBox). *A* primitive TBox *is a set of* inclusion assertions *of form $R \sqsubseteq R'$ where R and $R' \neq \bot$ are role names with the same arity.*

Definition 2 (Unrestricted ABox). *An* unrestricted ABox *is a set of* membership assertions *of form $r(t_1, \ldots, t_p)$ where r is a role name of arity p (i.e. a concept name if $p = 1$) and the t_i are terms.*

Definition 3 (DL-$\mathcal{SG}$ knowledge bases and queries). *A* DL-$\mathcal{SG}$ knowledge base *is an ordered pair $K = (T, A)$ where T is a primitive TBox and A is an unrestricted ABox. A* query *is an unrestricted ABox.*

We note $\mathcal{V}(K)$, $\mathcal{N}(K)$, $\mathcal{R}_i(K)$, $\mathcal{X}(K)$, $\mathcal{T}(K)$ the sets of names, individual names, role names, variables and terms contained in a KB K.

An *interpretation* is an ordered pair $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set called the *interpretation domain*, and the *interpretation function* $.^{\mathcal{I}}$ maps each individual name $n \in \mathcal{N}$ to an element $n^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and each role name r of arity k of $\mathcal{R}_k$ to a subset $r^{\mathcal{I}}$ of $(\Delta^{\mathcal{I}})^k$ (a concept name is mapped to a subset of $\Delta^{\mathcal{I}}$, and $\bot^{\mathcal{I}} = \emptyset$).

Definition 4 (Model of a primitive TBox). *Let T be a primitive TBox. An interpretation $\mathcal{I}$ is a* model *of T iff for every $R \sqsubseteq R'$ of T, $R^{\mathcal{I}} \subseteq R'^{\mathcal{I}}$.*

Definition 5 (Model of an unrestricted ABox). *Let A be an unrestricted ABox. An interpretation $\mathcal{I}$ is a model of A iff there is a mapping $\mu : \mathcal{T}(A) \rightarrow \Delta^{\mathcal{I}}$ (called a proof of A in $\mathcal{I}$) such that (i) $\forall n \in \mathcal{N}(A)$, $\mu(n) = n^{\mathcal{I}}$; and (ii) $\forall r(t_1, \ldots, t_k) \in A$, $(\mu(t_1), \ldots, \mu(t_k)) \in r^{\mathcal{I}}$.*

An unrestricted ABox is indeed the linear encoding, with the same semantics, of a positive, conjunctive, existential FOL formula without function symbols, while the TBox encodes implications of form $\forall x_1 \ldots \forall x_k (r(x_1, \ldots, x_k) \rightarrow r'(x_1, \ldots, x_k))$. A primitive TBox has the same semantics as a CG support, while an unrestricted ABox has the same semantics as a simple CG. This is why the soundness and completeness results presented in this paper can be stated as a direct translation of the CG versions in which they have been proven. Note that (T, A) can also be seen as a relational database by freezing its variables. An interpretation is a model of a KB (T, A) iff it is a model of T and A.

Definition 6 (Satisfiability, validity, semantic consequence). *Let K and K' be two KBs (of any language presented here). K is said* satisfiable *iff there is an interpretation that is a model of K (otherwise, it is said* unsatisfiable*); K is said* valid *iff every interpretation is a model of K; and K' is called a* semantic consequence *of K (we note $K \models K'$) iff every model of K is also a model of K'.*

Property 1 (Satisfiability and validity). A DL-$\mathcal{SG}$ KB K is unsatisfiable iff its ABox contains a membership assertion of form $\bot(t)$ and it is valid iff $K = (\emptyset, \emptyset)$.

2.2 DL-$\mathcal{SG}$ inferences

Definition 7 (T-Homomorphism). *Let $K = (T, A)$ be a DL-$\mathcal{SG}$ KB and Q be a query. A T-homomorphism from Q to K is a mapping π from $\mathcal{T}(Q)$ to $\mathcal{T}(A)$ such that (i) $\forall n \in \mathcal{N}(Q)$, $\pi(n) = n$; and (ii) $\forall r(t_1, \dots, t_k) \in Q$, $\exists r' \in \mathcal{R}_k$ such that $r' \leq_T r$ and $r'(\pi(t_1), \dots, \pi(t_k)) \in A$.*

A T-homomorphism is classically called *projection* (a graph homomorphism) in CGs. Note that the set of all T-homomorphisms of Q to A corresponds to the answer set of Q in the relational database associated to (T, A).

Deciding if there is a T-homomorphism from Q to A is an NP-complete problem (actually polynomial in data complexity), that becomes polynomial when the graph Q has a tree-like structure [13, 11, 4].

Theorem 1. *Let $K = (T, A)$ be a DL-$\mathcal{SG}$ KB and Q be a query. Then $K \models Q$ iff either K is unsatisfiable or there is a T-homomorphism from Q to A.*

3 DL-$\mathcal{SR}$: adding expressive rules

The extension of DL-$\mathcal{SG}$ with rules yields a formalism equivalent to the CG fragment $\mathcal{SR}$. Results of this section not associated with a reference are from [6].

3.1 DL-$\mathcal{SR}$ syntax and semantics

Definition 8 (Rules). *A* rule *is an ordered pair (H, C) of unrestricted A-boxes, where H is called the* hypothesis *of the rule, and C its* conclusion.

An *RBox* is a set of rules. A DL-$\mathcal{SR}$ knowledge base is a triple $K = (T, A, R)$ where T is a primitive TBox, A an unrestricted ABox and R is an RBox.

Definition 9 (Models of a rule). *Let (H, C) be a rule and $\mathcal{I}$ be an interpretation. Then $\mathcal{I}$ is a model of (H, C) iff for every proof μ of H in $\mathcal{I}$, there is a proof μ' of $H \cup C$ in $\mathcal{I}$ such that $\forall t \in \mathcal{T}(H), \mu'(t) = \mu(t)$. An interpretation is a model of an RBox iff it is a model of each of its rules.*

The FOL formula associated with (H, C) is $\forall x_1 \dots \forall x_q (\wedge_H \rightarrow (\exists y_1 \dots \exists y_p \wedge_C))$ where $\{x_1, \dots, x_q\} = \mathcal{X}(H)$ and $\{y_1, \dots, y_p\} = \mathcal{X}(C) \setminus \mathcal{X}(H)$, and $\wedge_H$, $\wedge_C$ are the conjunctions of atoms in H and C.

A model of a DL-$\mathcal{SR}$ KB (T, A, R) is a model of T, A and R. The DL-$\mathcal{SR}$-DEDUCTION problem is as follows: "given K and Q, does $K \models Q$ hold?"

3.2 DL-$\mathcal{SR}$ inferences: Forward Chaining of rules

Definition 10 (Application of a rule). *Let $K = (T, A)$ be a DL-$\mathcal{SG}$ KB. A rule (H, C) is said* applicable *to K if there is a T-homomorphism π from H to A. In this case, the* application of (H, C) on K according to π *produces an ABox $A \cup \rho(C, \pi)$ where $\rho(C, \pi)$ is obtained as follows from a copy of C: associate to each variable $x \in \mathcal{X}(C)$ the term $\pi(x)$ if defined, a new variable otherwise, and replace each occurrence of a variable x in C by its associated term.*

Definition 11 (Deriving a KB). *Let $K = (T, A, R)$ be a DL-$\mathcal{SR}$ KB. Then $K' = (T, A', R)$ is an* immediate derivation of K *iff there is a rule $(H, C) \in R$ and a T-homomorphism π from H to A such that $A' = A \cup \rho(C, \pi)$. K' is a* derivation *of K iff there is a finite sequence of KBs $K = K_0, K_1, \ldots, K_n = K'$ such that, $\forall 1 \leq i \leq n$, K_i is an immediate derivation of K_{i-1}.*

Theorem 2. *Let K be a DL-$\mathcal{SR}$ KB and Q be a query. Then $K \models Q$ iff there is a derivation $K' = (T, A', R)$ of K such that either A' contains an assertion of form $\bot(t)$ (K is unsatisfiable) or there is a T-homomorphism from Q to A'.*

According to this latter theorem, and since the derivation mechanism is confluent (*i.e.* the Church-Rosser property holds), if $K \models Q$, any sequence of breadth-first applications of rules will lead to a KB that is either unsatisfiable or contains an answer to Q. More precisely, let us consider a DL-$\mathcal{SR}$ KB $K = (T, A, R)$. An immediate saturation of K is a KB noted $\Sigma_1(K) = (T, A \cup_{1 \leq i \leq q} \rho(C_{j_i}, \pi_i), R)$ obtained by computing the set of all pairs $\{(\pi_i, (H_{j_i}, C_{j_i}))\}_{1 \leq i \leq q}$ where π_i is a T-homomorphism from H_{j_i} to A. We call a *saturation of K at rank k* the KB $\Sigma_k(K) = \Sigma_1(\Sigma_{k-1}(K))$. Forward Chaining builds successive saturations, checking if they contain an answer to Q or are unsatisfiable, until some halting condition is achieved. However, since DL-$\mathcal{SR}$-DEDUCTION is undecidable, no universal halting condition can be found. A sufficient condition is when there is a T-homomorphism from the ABox of $\Sigma_k(K)$ to the ABox of $\Sigma_{k-1}(K)$. In this case, $\Sigma_k(K)$ contains no new knowledge w.r.t. $\Sigma_{k-1}(K)$. This leads to the following abstract characterization of decidable subclasses of DL-$\mathcal{SR}$-DEDUCTION.

Definition 12 (Finite expansion sets). *An RBox R is a* finite expansion set *(f.e.s.) iff for any KB K, there is a rank k such that there is a T-homomorphism from the ABox of $\Sigma_k(K)$ to the ABox of $\Sigma_{k-1}(K)$.*

Let us give two concrete examples of f.e.s.: *range restricted rules*, which have no new variable in their conclusion, i.e. $\mathcal{X}(C) \subseteq \mathcal{X}(H)$, and *disconnected rules*, where $\mathcal{X}(H) \cap \mathcal{X}(C) = \emptyset$. In both cases, DL-$\mathcal{SR}$-DEDUCTION is decidable, and even NP-complete (and polynomial in data complexity). Note that some decidable subclasses of the deduction problem do not correspond to f.e.s., and that the union of two f.e.s. is not necessarily a f.e.s.

4 Efficient reasonings in DL-$\mathcal{SR}$ using unifiers

The optimizations of Forward Chaining as well as the Backward Chaining presented in this section require a complex operation in order to find *unifiers*. Proving that there is no unifier of a query Q with a rule R means that applying R does not generate new answers to Q. Unifiers are thus used in forward chaining to reduce the number of rule applicability checks, as well as in backward chaining. 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 our rules have a more complex conclusion, the associated unification operation is also more complex. For the sake of completeness, a precise translation of its conceptual graph definition [14] into DL-$\mathcal{SR}$ is provided here.

Definition 13 (T-Unifier). *Let Q be a query, T be a primitive TBox, and (H, C) be a rule. A T-*unifier *of Q with (H, C) is a tuple $(Q', (T_Q, T_{Q_1}, \ldots, T_{Q_k}), (T_{C_1}, \ldots, T_{C_k}), \mu)$ where Q' is a non-empty subset of Q, T_Q and the T_{Q_i} form a partition of $\mathcal{T}(Q')$, the T_{C_j} are pairwise disjoint non-empty subsets of $(\mathcal{T}(H) \cap \mathcal{T}(C)) \cup \mathcal{N}(C)$ and $\mu : T_Q \rightarrow \mathcal{T}(C) \setminus (\cup_{1 \leq j \leq k} T_{C_j})$ is a mapping such that: (i) $\forall n \in \mathcal{T}_Q \cap \mathcal{N}$, $\mu(n) = n$; (ii) $\forall 1 \leq i \leq k$, $T_{Q_i} \cup T_{C_i}$ contains at most one individual name; (iii) $\forall r(t_1, \ldots, t_k) \in Q'$, $\exists r'(t'_1, \ldots, t'_k) \in C$ such that $r' \leq_T r$ and $\forall 1 \leq i \leq k$, either $\mu(t_i) = t'_i$ or $t_i \in T_{Q_j}$ and $t'_i \in T_{C_j}$; and (iv) $\forall r(t_1, \ldots, t_k) \in Q \setminus Q'$, $t_i \notin T_Q, 1 \leq i \leq k$.*

Note that it is not mandatory for a unifier to contain any of the sets T_{Q_i} and T_{C_i}. In this case, the unifier corresponds to a T-homomorphism of connected components (according to its graph encoding) of Q to C.

4.1 Unifiers for Forward Chaining

Property 2 ([5]). Let Q be a query, T be a primitive TBox, and (H, C) be a rule. Then there is no unifier of Q with (H, C) iff for every ABox A, for every immediate derivation $(T, A', \{(H, C)\})$ of $(T, A, \{(H, C)\})$, every T-homomorphism from Q to A' is also from Q to A.

Let (H_1, C_1) and (H_2, C_2) be two rules. Assume that we have computed all applications of (H_2, C_2) to an ABox A, obtaining A'. Lets us now apply (H_1, C_1) to A', yielding A''. Then, if there is no T-unifier of H_2 with (H_1, C_1), we know that there is no new T-homomorphism from H_2 to A''.

Let us consider a DL-$\mathcal{SR}$ KB $K = (T, A, R)$. We build (this can be done offline with $|R|^2$ calls to an NP-hard problem) the *graph of rule dependencies* GRD(T, R). The nodes of this graph are the rules of R, and there is an arc from the rule (H, C) to the rule (H', C') if and only if there is a T-unifier of H' with (H, C). In this case, this arc is labelled with all such unifiers. Forward Chaining is then modified to benefit from this graph. At first step, all rules have to be checked for applicability, then at rank $k > 1$, only successors (in the GRD) of rules applied at rank $k - 1$ have to be checked. Moreover, the unifiers labelling arcs can be used to reduce the search space of applicability checks [5]. Though this modified Forward Chaining effectively improves the runtime efficiency, the graph of rule dependencies can also be used as a theoretical tool to prove that Forward Chaining will be able to halt on a given RBox:

Property 3 ([5]). Let $K = (T, A, R)$ be a DL-$\mathcal{SR}$ KB such that the graph GRD(T, R) has no circuit. Define k as the size of the longest path in this graph. Then $\Sigma_k(K)$ and $\Sigma_{k+1}(K)$ are equivalent.

Note that a loop (a self-unifiable rule) in the GRD is considered as a circuit, and is sufficient to yield the undecidability of the deduction problem. Indeed, [3] shows that a KB containing a single rule can encode a universal Turing machine.

Finite expansion sets rely on the structural properties of rules to ensure decidability. PROP. 3 relies upon the structure of possible interactions between them. The next theorem presents a generalization of both approaches:

Theorem 3 ([5]). *Let $K = (T, A, R)$ be a DL-$\mathcal{SR}$ KB such that all strongly connected components of the graph GRD(T, R) are finite expansion sets. Then there is a finite integer k such that $\Sigma_k(K)$ and $\Sigma_{k+1}(K)$ are equivalent.*

Cutting circuits in the GRD (by removing arcs or rules) may be achieved through the following methods, that preserve completeness:

Method 1 [5]: Consider a DL-$\mathcal{SR}$ KB $K = (T, A, R)$. Then A can be considered as a rule $A_R = (\emptyset, A)$. Build the graph $G = \text{GRD}(T, R \cup \{A_R\})$. Then any rule of R that is not on a path whose origin is a rule with empty hypothesis (*i.e.* equivalent to an ABox) can be safely removed. This operation can be done offline. In the same way, let us add the rule $(Q, \emptyset)$ to the GRD and compute the T-unifiers of $(Q, \emptyset)$ with rules of G. Then rules that are not on a path whose destination is $(Q, \emptyset)$ can also be removed. This operation can be done at runtime.

Method 2: Rules can also be simplified to find less unifiers: if some information added by the application of (H, C) is necessarily present in the ABox A, then this information can be safely removed from C. This is the case when C can be partitioned into two sets C_1 and C_2 such that there is a T-homomorphism π from C_2 to $H \cup C_1$, with $\forall t \in \mathcal{T}(C_1) \cap \mathcal{T}(C_2)$, $\pi(t) = t$ (it is called a *folding* in graph theory). Then (H, C) can be safely replaced by (H, C_1).

Method 3: Consider two rules (H, C) and (H', C'). Suppose there is a *complete* T-unifier U of H' with (H, C), *i.e.* such that U corresponds to a T-homomorphism π of the whole H' to C. It means that, whenever (H, C) is applied to an ABox A, it is also be possible to add the conclusion of (H', C'). The rule (H, C) can thus be replaced by the rule $(H, C \cup \rho(C', \pi))$, and since U has been taken into account in this new rule, it can be safely removed from the labels of the arc $((H, C), (H', C'))$ in the GRD. If all T-unifiers labelling an arc are complete, it is thus possible to remove this arc entirely, up to a rewriting of its origin. Moreover, one has to compute, for every rule (H'', C'') such that H'' is unifiable with (H, C) or (H', C'), the new T-unifiers with $(H, C \cup \rho(C', \pi))$.

4.2 Unifiers for Backward Chaining

Definition 14 (Rewriting a query according to a unifier). *Let* $U = (Q', (T_Q, T_{Q_1}, \ldots, T_{Q_k}), (T_{C_1}, \ldots, T_{C_k}), \mu)$ *be a* T*-unifier of a query* Q *with a rule* (H, C)*. The rewriting of* Q *according to* $(U, (H, C))$ *is the query* $\sigma_Q(Q \setminus Q') \cup \sigma_C(H)$*, where the effect of* σ_X *is to specialize an ABox* A *by replacing its variables as follows: if* x *is a variable of* T_{X_i}*, replace each occurrence of* x *in* A *by the individual name* a *if* $a \in T_{C_i} \cup T_{Q_i}$*, otherwise by the new variable* x_i *associated with* $T_{Q_i} \cup T_{C_i}$*; if* x *does not belong to any* T_{C_i}*, replace each of its occurrences by the same new variable.*

Definition 15 (Rewriting sequence). *Let* Q *be a query,* R *be an RBox, and* T *be a TBox. A query* Q^R *is an* immediate (T, R)-rewriting of Q *iff there is* $(H, C) \in R$ *and a* T*-unifier* U *of* Q *with* (H, C) *such that* $Q^R = \sigma_Q(Q \setminus Q') \cup \sigma_C(H)$*.* Q^R *is a* (T, R)-rewriting of Q *iff there is a finite sequence* $Q = Q_0, \ldots, Q_k = Q^R$ *s.t.* $\forall 1 \leq i \leq k$*,* Q_i *is an immediate* (T, R)*-rewriting of* Q_{i-1}*.*

Theorem 4. *Let* Q *be a query and* $K = (T, A, R)$ *be a DL-*$\mathcal{SR}$ *KB. Then* $K \models Q$ *if and only if* $\emptyset$ *is a* $(T, R \cup \{(\emptyset, A)\})$*-rewriting of* Q *or a* $(T, R \cup \{(\emptyset, A)\})$*-rewriting of* $\{\bot(x)\}$ *(this latter condition meaning that* K *is unsatisfiable).*

This latter theorem is the basis for the sound and complete Backward Chaining procedure of [14], that develops in a breadth-first manner all possible rewritings of the query Q. Another halting condition can be added: never rewrite a query semantically equivalent to a query that has already been explored. The GRD can also be used to optimize reasonings, as shown by the next property.

Property 4. ([7]) Let G be the GRD of a DL-$\mathcal{SR}$ KB $K = (T, A, R)$. Let $\mathcal{U}$ be the set of T-unifiers of a query Q with the rules in R. Let Q^R be an immediate (T, R)-rewriting of Q using $(H, C) \in R$. Then the T-unifiers of Q^R with the rules in R is the union of a subset of $\mathcal{U}$ and of the T-unifiers of H with the rules in R.

As an immediate consequence of this property, Backward Chaining is ensured to halt (as does Forward Chaining) when the GRD has no circuit:

Property 5. Let Q be a query, T be a TBox and R be an RBox such that $GRD(T, R)$ has no circuit. Then the set of (T, R)-rewritings of Q is finite (up to an isomorphism).

Let us now introduce the new notion of *finite unification sets*, that plays the same role in Backward Chaining than f.e.s. in Forward Chaining.

Definition 16 (Finite Unification Sets). *An RBox* R *is a* finite unification set *(f.u.s.) of rules iff for every TBox* T*, for every query* Q*, the set of* (T, R)*-rewritings of* Q *is finite (up to an isomorphism).*

If we prove that, for any query Q, all immediate (T, R)-rewritings of Q produce a query Q^R such that $|Q^R| \leq |Q|$, then the RBox R is a f.u.s. If, moreover, the size of the query strictly decreases ($|Q^R| < |Q|$), then DL-$\mathcal{SR}$-DEDUCTION becomes an NP-complete problem. Note, however, that though we have exhibited kinds of f.e.s. (*e.g.* range restricted, disconnected, ...) that can be automatically checked by a computer program, no concrete case of f.u.s. is proposed here. As in Forward Chaining, structural restrictions on rules as well as on their interactions can be combined for more interesting decidable subclasses:

Theorem 5. *Let T be a TBox and R be an RBox such that all strongly connected components of the graph $GRD(T, R)$ are f.u.s. Then for every query Q, the set of (T, R)-rewritings of Q is finite (up to an isomorphism).*

4.3 Combining Forward and Backward Chaining

Finally, both results in TH. 3 and TH. 5 can be generalized to obtain a larger decidable subclass, for which we exhibit a mixed Forward/Backward Chaining.

Definition 17 (Finite expansion/unification). *Let T be a TBox and R be an RBox. The graph $G = GRD(T, R)$ has the* finite expansion/unification (f.e/u.) property *if all strongly connected components of G are either f.e.s. or f.u.s., and there is no path from a rule in a f.u.s. to a rule in a f.e.s. If a strongly connected component contains only one rule without self-loop, then it can be considered either as a f.e.s. or a f.u.s.*

Property 6. Let $K = (T, A, R_1 \cup R_2)$ be a DL-$\mathcal{SR}$ KB such that there is no path from a rule of R_2 to a rule of R_1 in GRD(K). Then $K \models Q$ iff there is an ABox A' such that $(T, A, R_1) \models A'$ and $(T, A', R_2) \models Q$.

Theorem 6. *DL-$\mathcal{SR}$-DEDUCTION is decidable for KBs whose GRD has the f.e/u. property.*

Proof. We use a sound and complete algorithm that mixes Forward and Backward Chainings. Let $G = \text{GRD}(T, R)$. If G has the f.e/u. property, let $R = R_e \cup R_u$, where R_e and R_u are composed of rules belonging respectively to a f.e.s. and to a f.u.s. Since there is no path from a rule in R_u to a rule in R_e, by PROP. 6, we have $(T, A, R) \models Q$ iff there is an ABox A' such that $(T, A, R_e) \models A'$ and $(T, A', R_u) \models Q$. Let G_e (resp. G_u) be the subgraph of G induced by rules in R_e (resp. R_u). Then all strongly connected components of G_e are f.e.s., and there exists a finite rank k for which all knowledge that can be added with rules in R_e is present in $A^S = \Sigma_k((T, A, R_e))$ (TH. 3). Then $(T, A, R) \models Q$ iff $(T, A^S, R_u) \models Q$. Since all strongly connected components of G_u are f.u.s., Q can be rewritten as a finite set of queries for which the union of answers in A^S form the answers of Q in A^S (TH. 5). □

5 Relationships with FOL-reducibility of DL-Lites

5.1 FOL-reducibility and the f.e/u. property

We are interested in DLs whose semantics can be encoded in a DL-$\mathcal{SR}$ KB (and more generally in extensions of $\mathcal{SR}$ such as [6]). Given a specific description logic

DL_X, our goal is to propose a transformation τ_X from assertions of DL_X into assertions or rules of DL-$\mathcal{SR}$, preserving essential semantic properties. Ideally, this transformation should preserve models ($\mathcal{I}$ is a model of the DL_X KB (T, A) iff it is a model of the DL-$\mathcal{SR}$ KB $\tau_X(T, A) = (T', A', R)$). This case also has the added benefit that we can immediately and safely extend DL_X with expressive rules of the form presented in this paper. Since such a strong property cannot always be ensured, we can rely on weaker semantic properties such that "(T, A) and $\tau_X(T, A) = (T', A', R)$ provide the same answers to conjunctive queries", used in the following example. Note that such a transformation is not always possible in DL-$\mathcal{SR}$, since general negation and disjunction of concepts or roles, for instance, are out of the scope of our framework. Assume now that such a transformation holds for a description logic (and generally for a KR formalism) DL_X. We believe that the theoretical tools presented here provide a solid framework to investigate combinatorial properties of DL_X-DEDUCTION. According to the transformation τ_X used, the rules obtained from the translation of a DL_X TBox will have a particular structure, and specific interactions that can be analyzed using the GRD. We hope this analysis will allow to develop new complexity results, new and efficient algorithms for the source language, as well as provide us with an original, structural insight on the reasonings involved. On the other hand, the specific algorithms developed for the source language can lead to new structural results for DL-$\mathcal{SR}$: finite unification sets introduced in this paper stemmed from our understanding of DL-Lite algorithms.

As an example of combinatorial properties of a language that can be analyzed through the theoretical tools developed for DL-$\mathcal{SR}$, consider now the FOL-reducibility property in the DL-Lite family [9]. Let us consider a DL-Lite KB $K = (T, A)$ and a conjunctive query Q. Basically, the FOL-reducibility property states that there are a relational database D obtained from a linear translation of A, and a finite set of conjunctive queries $\{Q_1, \ldots, Q_p\}$ obtained from T and Q, such that the answers to Q in K is the union of the (database) answers of the Q_i in D. Now, let us consider a linear transformation τ associating to any DL KB $K = (T, A)$ a DL-$\mathcal{SR}$ KB $\tau(K) = (T', A', R')$ that preserves answers. Let us now consider an arbitrary KB of this DL, and its translation into DL-$\mathcal{SR}$. If (T', R') has the f.e/u. property, then for any query Q, there is a finite set of conjunctive queries $\{Q_1, \ldots, Q_k\}$ such that the answers to Q are the union of the answers to the Q_i in a finite saturation of A'. If, moreover, the saturation of A' can be built in linear time (using very specific f.e.s containing rules such as the rule $A_1[R, i]$ presented below), then the FOL-reducibility property holds.

5.2 Example: DLR-Lite$_{\mathcal{R}}$

As an example of the above-mentioned approach, consider now the description logic DLR-Lite$_{\mathcal{R}}$ (*i.e.* the DL-Lite$_{\mathcal{R}}$ of [9] extended with n-ary relations). We translate TBox assertions of this DL into rules or primitive TBox inclusion assertions of DL-$\mathcal{SR}$ as shown in the table. The first column presents all possible forms of TBox assertions in DLR-Lite$_{\mathcal{R}}$, the second the name of its associated rule in DL-$\mathcal{SR}$, and the third the form of this rule. In the fourth column, we

DLR-Lite$_\mathcal{R}$	Rule name	Form of the rule	Notes
	$A_1[R,i]$	$R(x_1,\ldots,x_i,\ldots,x_k) \rightarrow \exists_i R(x_i)$	r.r. (f.e.s.)
	$A_2[R,i]$	$\exists_i R(x_i) \rightarrow R(x_1,\ldots,x_i,\ldots,x_k)$	n.i. (f.u.s.)
$A \sqsubseteq A'$		$A \sqsubseteq A'$	p.i.a.
$A \sqsubseteq \exists_i R$		$A \sqsubseteq \exists_i R$	p.i.a.
$\exists_i R \sqsubseteq A$		$\exists_i R \sqsubseteq A$	p.i.a.
$\exists_i R \sqsubseteq \exists_j R'$		$\exists_i R \sqsubseteq \exists_j R'$	p.i.a.
$A \sqsubseteq \neg A'$	$C_1[A,A']$	$A(x), A'(x) \rightarrow \bot(x)$	r.r. (f.e.s.)
$A \sqsubseteq \neg\exists_i R$	$C_2[A,R,i]$	$A(x), \exists_i R(x) \rightarrow \bot(x)$	r.r. (f.e.s.)
$\exists_i R \sqsubseteq \neg A$	$C_3[A,R,i]$	$A(x), \exists_i R(x) \rightarrow \bot(x)$	r.r. (f.e.s.)
$\exists_i R \sqsubseteq \neg\exists_j R'$	$C_4[R,i,R',j]$	$\exists_i R(x), \exists_j R'(x) \rightarrow \bot(x)$	r.r. (f.e.s.)
$R \sqsubseteq R'$		$R \sqsubseteq R'$	p.i.a.
$R \sqsubseteq \neg R'$	$C_5(R,R')$	$R(x_1,\ldots,x_k), R'(x_1,\ldots,x_k) \rightarrow \bot(x_1)$	r.r. (f.e.s.)

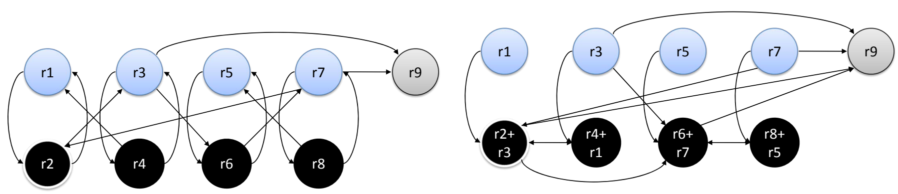

Fig. 1. The initial GRD and its simplification

indicate when the rule is range restricted (r.r.), non increasing (n.i., meaning that no rewrite of a query using that rule can increase its size), or that the rule is a primitive inclusion assertion (p.i.a.), and thus remains in the DL-$\mathcal{SR}$ TBox. The two first lines of this table show rules that translate the existential role constructor of DLR-Lite$_R$, where: *(i)* "$\exists_i R$" has to be considered as a unary predicate symbol, and *(ii)* for each role name in the KB with arity k, there are k different pairs of such rules. This translation preserves answers to conjunctive queries, as shown by the reasonings involved in [9].

Let us now consider the example of a specific DLR-Lite$_\mathcal{R}$ KB, which comes from [9]. The ABox A contains the atom $ht(J,M)$, while the TBox contains the assertions $a_1 = (pr \sqsubseteq \exists tt)$, $a_2 = (st \sqsubseteq \exists ht)$, $a_3 = (\exists tt^- \sqsubseteq st)$, $a_4 = (\exists ht^- \sqsubseteq pr)$, $a_5 = (pr \sqsubseteq \neg st)$. This KB is translated into a DL-$\mathcal{SR}$ KB $K = (T,A,R)$ where $T = \{a_1,a_2,a_3,a_4\}$ and the RBox R contains the rules $r_1 = A_1[ht,1]$, $r_2 = A_2[ht,1]$, $r_3 = A_1[ht,2]$, $r_4 = A_2[ht,2]$, $r_5 = A_1[tt,1]$, $r_6 = A_2[tt,1]$, $r_7 = A_1[tt,2]$, $r_8 = A_2[tt,2]$, and $r_9 = C_1[pr,st]$. The graph GRD(T,R) can be seen on the left side of FIG. 1, it contains two strongly connected components, the first one containing r_9, and the second all other rules. Since this component contains both f.e.s. and f.u.s. (in black), it does not even satisfy our most general decidability characterization. However, its simplification according to the methods 2 and 3 of SECT. 4.1 generates the graph on the right side of FIG. 1, which has the f.e/u property. Note that, for any DLR-Lite$_\mathcal{R}$ KB, the associated GRD will contain the same circuits as the graph presented here. Then we have proven the f.e/u property for any instance of the language: DLR-Lite$_\mathcal{R}$ is FOL-reducible.

6 Conclusion

We have presented in this paper DL-$\mathcal{SR}$, an extension of a lite DL with expressive rules, inspired by the CG fragment $\mathcal{SR}$. We have investigated some combinatorial properties of this language, and have extended the results of [6, 5, 7] with the new notion of finite unification sets of rules. This notion is combined with previous ones to yield a new decidable subclass using a mixed Forward/Backward Chaining procedure. On any instance of a DLR-Lite$_{\mathcal{R}}$ KB, our algorithms will automatically devise a strategy allowing to answer conjunctive queries in a way consistent with the FOL-reducibility property of the language. We intend now to use this framework to define new lite DLs, and propose efficient algorithms computing DEDUCTION in these languages.

References

1. F. Baader, R. Molitor, and S. Tobies. Tractable and Decidable Fragments of Conceptual Graphs. In *Proc. of ICCS'99*, volume 1640 of *LNAI*, pages 480–493. Springer, 1999.
2. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
3. J.-F. Baget. *Représenter des connaissances et raisonner avec des hypergraphes: de la projection à la dérivation sous contraintes.* PhD thesis, U. Montpellier II, 2001.
4. J.-F. Baget. Simple Conceptual Graphs Revisited: Hypergraphs and Conjunctive Types for Efficient Projection Algorithms. In *Proc. of ICCS'03*, volume 2746 of *LNAI.* Springer, 2003.
5. J.-F. Baget. Improving the forward chaining algorithm for conceptual graphs rules. In *KR*, pages 407–414, 2004.
6. J.-F. Baget and M.-L. Mugnier. The Complexity of Rules and Constraints. *JAIR*, 16:425–465, 2002.
7. J.-F. Baget and E. Salvat. Rules dependencies in backward chaining of conceptual graphs rules. In *ICCS*, pages 102–116, 2006.
8. 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.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
10. Michel Chein and Marie-Laure Mugnier. Concept types and coreference in simple conceptual graphs. In K. E. Wolff et al, editor, *ICCS*, volume 3127 of *LNAI*, pages 303–318. Springer, 2004.
11. G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *PODS'99*, 1999.
12. M.-L. Mugnier and M. Leclère. On querying simple conceptual graphs with negation. *Data Knowl. Eng.*, 60(3):468–493, 2007.
13. M.L. Mugnier and M. Chein. Polynomial algorithms for projection and matching. In *Proc. the 7th Workshop on Conceptual Graphs*, LNAI. Springer.
14. 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.