Towards expressive KR systems integrating datalog and description logics: preliminary report

Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
rosati@dis.uniroma1.it

Abstract

We study the problem of extending hybrid knowledge-based systems integrating rules and structured representation of information. Specifically, we define disjunctive $\mathcal{AL}$-LOG, a hybrid formalism which combines two very expressive knowledge representation languages: disjunctive DATALOG and the description logic $\mathcal{ALC}$. First, by exploiting the use of nonmonotonic features of disjunctive DATALOG, we show that disjunctive $\mathcal{AL}$-LOG overcomes some severe representational limitations of analogous proposals. Then, in order to implement a system based on disjunctive $\mathcal{AL}$-LOG, we take advantage of recent advances in the technology of KR systems: in particular, we are able to define a reasoning method that allows for a full implementation of disjunctive $\mathcal{AL}$-LOG, obtained by relying on the optimized deductive services of two state-of-the-art implemented KR systems, i.e. the DL-based system FACT and the disjunctive DATALOG system DLV. We also provide both a computational characterization of reasoning in disjunctive $\mathcal{AL}$-LOG and a general technique for encoding extensional knowledge (ABox) in the system FACT.

1 Introduction

Research in knowledge representation and reasoning has recently shown significant advances in the implementation of reasoning services for very expressive knowledge representation (KR) systems. On the one hand, new techniques for efficient reasoning in very expressive Description Logics have been defined, and prototype KR systems based on such optimized methods have been recently developed (e.g., the systems FACT [Horrocks, 1998] and DLP [Horrocks and Patel-Schneider, 1999]). On the other hand, a similar improvement in deduction techniques has lead to the first implementations of very powerful deductive database systems, which extend the relational language DATALOG with nonmonotonic features like negation as failure and disjunction [Eiter *et al.*, 1997] (e.g., the systems SMODELS [Niemelä and Simons, 1997] and DLV [Eiter *et al.*, 1998]).

Another recent development in knowledge representation is the idea of building hybrid systems combining rules and structured representation of information. In particular, several approaches [Levy and Rousset, 1996, Donini *et al.*, 1998] have studied the problem of integrating DATALOG with Description Logics (DL). Notably, such hybrid formalisms prove very well suited for the construction of tools for accessing heterogeneous information systems. In particular, a decidable fragment of CARIN is used in the Information Manifold, a system developed at AT&T for integrated access to different structured information sources on the World Wide Web [Levy *et al.*, 1996].

However, due to efficiency reasons, such implemented hybrid systems impose severe restrictions on the expressiveness of the representation language, both in the structural and in the relational component. Specifically, in the KR subsystem of Information Manifold the structural component allows for the use of a very restricted fragment of the logic $\mathcal{ALC}$, while the relational component only allows for positive DATALOG programs. Such limitations in the representational power restrict the range of application of hybrid KR systems.

Specifically, the lack of nonmonotonic features in the relational subsystem makes it hard to deal with the relevant notions of *information completeness* [Etzioni *et al.*, 1994, Duschka and Genesereth, 1997] and *disjunctive information sources* [Duschka and Genesereth, 1998] in the mentioned applications to heterogeneous information systems, while the restrictions in the description logic language impose drastic approximations in modeling relationships among structured information sources [Calvanese *et al.*, 1998].

In this paper we define disjunctive $\mathcal{AL}$-LOG, a hybrid formalism which overcomes the above mentioned limitations of existing proposals, and provide a prototype implementation of a KR system based on such a formal framework. Our approach is based on the idea of taking advantage of the recent advances in the technology of KR systems: hence, our formalism integrates the languages used in the most expressive among implemented KR systems, i.e. disjunctive DATALOG and the description logic $\mathcal{ALC}$. In this way we provide a very expressive hybrid formalism for knowledge representation; moreover, we are able to define a reasoning method that allows for a full implementation of disjunctive $\mathcal{AL}$-LOG which relies on the optimized deductive services of the DL-based system FACT and the disjunctive DATALOG system DLV.

As byproducts of this research effort, we obtain both a computational characterization of reasoning in disjunctive $\mathcal{AL}$-LOG, and a general technique for encoding extensional knowledge (ABox) in the system FACT (which does not provide explicit constructs for extensional assertions).

In the next section we briefly recall disjunctive DATALOG. Then, we introduce the system disjunctive $\mathcal{AL}$-LOG, and provide both deductive methods and a computational characterization of reasoning for disjunctive $\mathcal{AL}$-LOG. Finally, we deal with the problem of implementing disjunctive $\mathcal{AL}$-LOG, showing an encoding of extensional knowledge in FACT.

2 Disjunctive $\mathcal{AL}$-LOG

Disjunctive $\mathcal{AL}$-LOG is obtained by adding to disjunctive DATALOG [Eiter *et al.*, 1997] the possibility of using variables which range over the set of instances of a specified concept (or role) of an $\mathcal{ALC}$ knowledge base. This is realized by adding *constraints*, corresponding to concept (role) expressions, in the body of DATALOG rules.

Definition 1 *A* disjunctive $\mathcal{AL}$-LOG knowledge base $\mathcal{H}$ *is a pair* $(\Sigma, \mathcal{P})$*, where* Σ *(also called the* structural component *of* $\mathcal{H}$*) is an* $\mathcal{ALC}$*-knowledge base, and* $\mathcal{P}$ *(also called the* relational component *of* $\mathcal{H}$*) is a hybrid program, i.e. a pair* $(\mathcal{P}_E, \mathcal{P}_I)$ *such that* $\mathcal{P}_E$ *is a set of ground literal predicates of the form* $p(\overline{a})$ *or* $\neg p(\overline{a})$*, where* $\overline{a} = \{a_1, \ldots, a_n\}$ $(n \geq 0)$ *is such that* $a_i \in \mathcal{O}_\Sigma$ *for each* $i = 1, \ldots, n$*, and* $\mathcal{P}_I$ *is a set of* hybrid rules *of the form*

$$p_1(\overline{X_1}) \vee \ldots \vee p_l(\overline{X_l}) \leftarrow q_1(\overline{Y}_1), \ldots, q_l(\overline{Y}_m), \text{not } q_{m+1}(\overline{Y}_{m+1}), \ldots, \text{not } q_n(\overline{Y}_n), t_1(\overline{Z}_1), \ldots, t_k(\overline{Z}_k)$$

such that $l, n, m, k \geq 0$*, the set of* DATALOG *predicate symbols is disjoint from the set of concept and role symbols appearing in* Σ *and* $\mathcal{P}$*, and for each hybrid rule:*

1. *each rule is safe, i.e. each variable appearing in the head of the rule occurs in the body of the rule;*
2. $p_1, \ldots, p_l, q_1, \ldots, q_n$ *are* DATALOG *predicates;*
3. $t_1, \ldots, t_k$ *are all* $\mathcal{ALC}$*-concepts or* $\mathcal{ALC}$*-roles (called* constraints*). Without loss of generality, we assume that each* t_i *is a concept (or role)* name *(a concept expression* C *can be easily represented by means of a new concept name* D*, by adding to the TBox* $\mathcal{T}$ *the two inclusions* $C \sqsubseteq D$ *and* $D \sqsubseteq C$*);*

Observe that the condition $a_i \in \mathcal{O}_\Sigma$ can be easily obtained by adding to $\mathcal{A}$ the set of assertions $\top(a_i)$ for each such a_i.

The semantics of disjunctive $\mathcal{AL}$-LOG is given in the following way. Let $(\Sigma, \mathcal{P})$ (with $\mathcal{P} = (\mathcal{P}_E, \mathcal{P}_I)$) be a disjunctive $\mathcal{AL}$-LOG knowledge base. The *ground instantiation* of $\mathcal{P}$ wrt $\mathcal{O}_\Sigma$, denoted as $gr(\mathcal{P}, \mathcal{O}_\Sigma)$, is the program obtained from $\mathcal{P}$ by substituting each rule in $\mathcal{P}_I$ with the set of rules (one for each instantiation) obtained by substituting the variables in the rule with elements of $\mathcal{O}_\Sigma$.

Definition 2 *Let* $(\Sigma, \mathcal{P})$ *(with* $\mathcal{P} = (\mathcal{P}_E, \mathcal{P}_I)$*) be a disjunctive* $\mathcal{AL}$-LOG *knowledge base, and let* $\mathcal{I}$ *be a* $\mathcal{DL}$*-interpretation. Then,* $gr(\mathcal{P}, \mathcal{O}_\Sigma)|_\mathcal{I}$ *is the ground* DATALOG *program obtained by modifying* $gr(\mathcal{P}, \mathcal{O}_\Sigma)$ *as follows: for each hybrid rule* $r \in gr(\mathcal{P}, \mathcal{O}_\Sigma)$*, if there exists a constraint* $C(a)$ *(resp.* $R(a,b)$*) in* r *such that* $\mathcal{I}$ *does not satisfy* $C(a)$ *(resp.* $R(a,b)$*), then* r *is eliminated, otherwise all constraints are eliminated from* r*.*

Given a $\mathcal{DL}$-interpretation $\mathcal{I}$ and an interpretation I of $gr(\mathcal{P}, \mathcal{O}_\Sigma)$, the pair $(\mathcal{I}, I)$ is a *model* of $\mathcal{H} = (\Sigma, \mathcal{P})$ if $\mathcal{I}$ is a model of Σ and I is a stable model of $gr(\mathcal{P}, \mathcal{O}_\Sigma)$. We say that $q(\overline{a})$ is a *ground query to* $\mathcal{H} = (\Sigma, \mathcal{P})$ if q is a DATALOG predicate symbol appearing in $\mathcal{P}$ and $\overline{a} = a_1, \ldots, a_n$, where each $a_i \in \mathcal{O}_\Sigma$ for $i = 1, \ldots, n$. We write $\mathcal{H} \models q(\overline{a})$ iff, for each model $(\mathcal{I}, I)$ of $\mathcal{H}$, $q(\overline{a})$ is contained in I.

Comparing disjunctive $\mathcal{AL}$-LOG with analogous proposals, namely $\mathcal{AL}$-LOG [Donini *et al.*, 1998] and CARIN[Levy and Rousset, 1996], it turns out that disjunctive $\mathcal{AL}$-LOG generalizes $\mathcal{AL}$-LOG, since the relational component is extended by adding the possibility of expressing classical negation, negation as failure, and disjunction in program rules; moreover, it is possible to use role expressions as constraints. As in the case of $\mathcal{AL}$-LOG, the semantics of disjunctive $\mathcal{AL}$-LOG imposes that variables in the constraints must range over the set $\mathcal{O}_\Sigma$ of individuals explicitly mentioned in the knowledge base. This limitation is not present in CARIN, which also allows binary predicates (corresponding to role expressions) to appear as constraints in the rules. However, the combination of these features leads in the general case to undecidability, as shown in [Levy and Rousset, 1996].

3 Reasoning in disjunctive $\mathcal{AL}$-LOG

In order to provide a deduction method for disjunctive $\mathcal{AL}$-LOG, we give the following preliminary definition.

Definition 3 *Let $\mathcal{H} = (\Sigma, \mathcal{P})$ be a disjunctive $\mathcal{AL}$-LOG knowledge base, let T_C be the set of concept expressions appearing as constraints in $\mathcal{P}$, and let T_R be the set of role expressions appearing as constraints in $\mathcal{P}$. Then, we denote with $\mathcal{V}_\mathcal{H}$ the set of assertions*

$$\{t(a) : t \in T_C \text{ and } a \in \mathcal{O}_\Sigma\} \cup$$
$$\{t(a,b) : t \in T_R \text{ and } a, b \in \mathcal{O}_\Sigma\}$$

The idea behind the above definition is that $\mathcal{V}_\mathcal{H}$ identifies the set of *all* possible instantiations of the constraints of program $\mathcal{P}$ over $\mathcal{O}_\Sigma$. Thus, we can divide the set of all $\mathcal{DL}$-interpretations into equivalence classes, based on the way in which such interpretations "evaluate" (in the sense that will be specified in the following) all possible instantiations of the constraints of program $\mathcal{P}$ over $\mathcal{O}_\Sigma$. Notably, for each pair $\mathcal{I}_1, \mathcal{I}_2$ of $\mathcal{DL}$-interpretations belonging to the same equivalence class, $\{I : (\mathcal{I}_1, I)$ is a model for $\mathcal{H}\} = \{I : (\mathcal{I}_2, I)$ is a model for $\mathcal{H}\}$. Therefore, in order to establish whether $\mathcal{H} \models q(\overline{a})$, we only need to consider one representative of each equivalence class. In particular, we identify each equivalence class with the corresponding partition (V_P, V_N) of $\mathcal{V}_\mathcal{H}$. More precisely, V_P is the set of assertions from $\mathcal{V}_\mathcal{H}$ satisfied by the $\mathcal{DL}$-interpretations of the corresponding class, while V_N is the set of assertions from $\mathcal{V}_\mathcal{H}$ which are not satisfied.

Of course, not all the partitions of $\mathcal{V}_\mathcal{H}$ are compatible with Σ. We say that a partition (V_P, V_N) *is consistent with* Σ iff there exists a $\mathcal{DL}$-interpretation in the equivalence class represented by (V_P, V_N) which satisfies Σ. Now, if (V_P, V_N) is consistent with Σ, then we are able to compute the set

$$\{I : (\mathcal{I}, I) \text{ is a model for } \mathcal{H} \text{ and } \mathcal{I} \text{ induces } (V_P, V_N)\}$$

by identifying the set of stable models of the DATALOG program obtained from $gr(\mathcal{P}, \mathcal{O}_\Sigma)$, substituting all occurrences of the constraints according to the partition (V_P, V_N). We formalize this notion as follows.

Definition 4 *Let $\mathcal{H} = (\Sigma, \mathcal{P})$ be a disjunctive $\mathcal{AL}$-LOG KB, and let (V_P, V_N) be a partition of the set $\mathcal{V}_\mathcal{H}$. The, we denote with $\mathcal{P}(V_P, V_N)$ the disjunctive* DATALOG *program obtained from the hybrid program $\mathcal{P} = (\mathcal{P}_I, \mathcal{P}_E)$ as follows:*

1. *introduce a new extensional* DATALOG *predicate $\tau(t_i)$ for each concept or role t_i appearing as a constraint;*
2. *add to $\mathcal{P}_E$ the set of facts $\{\tau(t_i(\overline{a})) : t_i(\overline{a}) \in V_P\} \cup \{\neg\tau(t_i(\overline{a})) : t_i(\overline{a}) \in V_N\}$;*
3. *for each constraint t_i, replace in $\mathcal{P}$ all occurrences of t_i with $\tau(t_i)$.*

Algorithm D-$\mathcal{AL}$-LOG-Entails$(\mathcal{H}, q(\overline{a}))$
Input: disjunctive $\mathcal{AL}$-LOG-KB $\mathcal{H} = (\Sigma, \mathcal{P})$,
ground query $q(\overline{a})$;
Output: true if $\mathcal{H} \models q(\overline{a})$, false otherwise.
begin
compute $\mathcal{V}_\mathcal{H}$;
if there exists partition (V_P, V_N) of $\mathcal{V}_\mathcal{H}$
such that
(V_P, V_N) is consistent with Σ **and**
DATALOG-Entails$(\mathcal{P}(V_P, V_N), q(\overline{a}))$ = false
then return false
else return true
end

Figure 1: Algorithm D-$\mathcal{AL}$-LOG-Entails.

Roughly speaking, in the construction of $\mathcal{P}(V_P, V_N)$ we eliminate from $\mathcal{P}$ all concept and role occurrences, encoding each constraint in terms of a new extensional predicate, and represent the guess (V_P, V_N) by means of a set of ground literal predicates. We prove that the stable models of $\mathcal{P}(V_P, V_N)$ are strictly related to the models of the hybrid knowledge base $\mathcal{H}$, as stated by the following property.

Lemma 5 *Let $\mathcal{H}$ be a disjunctive $\mathcal{AL}$-LOG KB, let $\mathcal{I}$ be a $\mathcal{DL}$-interpretation and let (V_P, V_N) be the partition induced by $\mathcal{I}$ over $\mathcal{V}_\mathcal{H}$. Then, a pair $(\mathcal{I}, I)$ is a model for $\mathcal{H}$ iff $\mathcal{I}$ satisfies Σ and I is a stable model of the program $\mathcal{P}(V_P, V_N)$.*

Based on the above properties, we are now able to define a method for reasoning in disjunctive $\mathcal{AL}$-LOG KBs. In particular, the algorithm D-$\mathcal{AL}$-LOG-Entails, reported in Figure 2, computes the entailment $\mathcal{H} \models q(\overline{a})$ for a disjunctive $\mathcal{AL}$-LOG knowledge base $\mathcal{H}$ and a ground query $q(\overline{a})$. In the algorithm, DATALOG-Entails denotes a known method for computing query answering in disjunctive DATALOG (see e.g. [Eiter *et al.*, 1998]).

Theorem 6 *Let $\mathcal{H}$ be a disjunctive $\mathcal{AL}$-LOG knowledge base and let $q(\overline{a})$ be a ground query to $\mathcal{H}$. Then, $\mathcal{H} \models q(\overline{a})$ iff D-$\mathcal{AL}$-LOG-Entails$(\mathcal{H}, q(\overline{a}))$ returns* true.

Proof. *If part.* Suppose $\mathcal{H} \not\models q(\overline{a})$. Then, by Lemma 5, it follows that there exists a model $(\mathcal{I}, I)$ for $\mathcal{H}$ such that $\mathcal{I}$ satisfies Σ, $q(\overline{a}) \notin I$ and I is a stable model for $\mathcal{P}(V_P, V_N)$, where (V_P, V_N) is the partition induced by $\mathcal{I}$ over $\mathcal{V}_\mathcal{H}$. Therefore, (V_P, V_N) is consistent with Σ, and DATALOG-Entails$(\mathcal{P}(V_P, V_N), q(\overline{a}))$ returns false. Therefore, D-$\mathcal{AL}$-LOG-Entails$(\mathcal{H}, q(\overline{a}))$ returns false.

Only-if part. Suppose D-$\mathcal{AL}$-LOG-Entails$(\mathcal{H}, q(\overline{a}))$ returns false. Then, correctness of the algorithm Constraints-Sat implies that there exists a partition (V_P, V_N) of $\mathcal{V}_\mathcal{H}$ which is consistent with Σ. Therefore, there exists a $\mathcal{DL}$-interpretation $\mathcal{I}$ such that $\mathcal{I}$ evaluates all constraints according to (V_P, V_N) and $\mathcal{I}$ satisfies Σ. Now, since DATALOG-Entails returns false, it follows that there exists a stable model I for $\mathcal{P}(V_P, V_N)$ such that $q(\overline{a}) \notin I$, and since Lemma 5 implies that $(\mathcal{I}, I)$ is a model for $\mathcal{H}$, it follows that $\mathcal{H} \not\models q(\overline{a})$. □

As for the complexity of reasoning in disjunctive $\mathcal{AL}$-LOG, using the algorithm D-$\mathcal{AL}$-LOG-Entails, we are able to state an upper bound for the computational complexity (combined complexity) of query answering in disjunctive $\mathcal{AL}$-LOG, which follows from the fact that satisfiability in $\mathcal{ALC}$ is in EXPTIME [Buchheit *et al.*, 1993] and query answering in disjunctive DATALOG is in coNEXPTIME^{NP} [Eiter *et al.*, 1997].

Lemma 7 *Let $\mathcal{H}$ be an disjunctive $\mathcal{AL}$-LOG KB, and let $q(\overline{a})$ be a ground query to $\mathcal{H}$. Then, the problem of establishing whether $\mathcal{H} \models q(\overline{a})$ is in coNEXPTIME^{NP}.*

Moreover, since entailment in DATALOG$^{\neg\vee}$ is coNEXPTIME^{NP}-hard [Eiter *et al.*, 1997], the following property holds.

Theorem 8 *Let $\mathcal{H}$ be an disjunctive $\mathcal{AL}$-LOG KB, and let $q(\overline{a})$ be a ground query to $\mathcal{H}$. Then, the problem $\mathcal{H} \models q(\overline{a})$ is coNEXPTIME^{NP}-complete.*

Analogously, it can be shown that if we restrict disjunctive $\mathcal{AL}$-LOG by disallowing the presence of disjunction in the head of hybrid rules, then entailment is coNEXPTIME-complete, which immediately follows from Theorem 6 and from the fact that query answering in DATALOG$^{\neg}$ is coNEXPTIME-complete [Eiter *et al.*, 1997]. In a similar way we can also easily prove that entailment in the original $\mathcal{AL}$-LOG framework is EXPTIME-complete.

Therefore, in disjunctive $\mathcal{AL}$-LOG (and its restrictions) the complexity of reasoning is the same as the complexity of reasoning in the pure relational case. That is, adding an $\mathcal{ALC}$ structural component to DATALOG does not increase the worst-case complexity of reasoning. Notably, such a result holds for the *combined* complexity of reasoning, i.e. when we assume as input of the problem both the knowledge base and the query. Interestingly, $\mathcal{AL}$-LOG exhibits a different behavior in the case of *data* complexity [Cadoli *et al.*, 1997].

4 Implementing disjunctive $\mathcal{AL}$-LOG

We now deal with the problem of implementing disjunctive $\mathcal{AL}$-LOG. To this aim, we exploit the possibility of relying on the implemented KR systems FACT and DLV. Due to lack of space, in the following we only sketch the main ideas underlying the implementation. A naive prototype based on the techniques presented below is currently under construction.

First, we define a new algorithm for the query answering problem $\mathcal{H} \models q(\overline{a})$ in disjunctive $\mathcal{AL}$-LOG. Basically, the algorithm looks for a pair $(\mathcal{I}, I)$ which is a model for $\mathcal{H}$ and such that I does not contain $q(\overline{a})$. The algorithm is tailored for the system DLV, since it exploits the fact that DLV outputs all the stable models of a given DATALOG program. We use this feature of the system together with the possibility of simulating the guess (V_P, V_N) by means of suitable disjunctive DATALOG rules: roughly, the idea is to simulate the guess of each assertion $t_i(\overline{a})$ by means of a disjunctive DATALOG rule of the form

$$\tau(t_i)(\overline{a}) \vee \neg\tau(t_i)(\overline{a}) \leftarrow$$

In this way it is possible to define a (polynomial) transformation of the hybrid program $\mathcal{P}$ in terms of a disjunctive DATALOG program $\mathcal{P}'$ which represents in its stable models all the possible guesses of the assertions of $\mathcal{V}_H$. In other words, a stable model I' of $\mathcal{P}'$ identifies a pair $(\mathcal{I}, I)$. It can be shown that such a pair is a model of $\mathcal{H}$ iff $\mathcal{I}$ satisfies the $\mathcal{ALC}$ KB Σ in $\mathcal{H}$; moreover, this check can be done by proving satisfiability in $\mathcal{ALC}$ of the knowledge base obtained from Σ by augmenting the ABox $\mathcal{A}$ with the set $assert(I')$ of assertions about constraints contained in I'.

Finally, only models that do not contain the query $q(\overline{a})$ are relevant, hence we force such a condition by adding to $\mathcal{P}'$ the rule $\leftarrow q(\overline{a})$. We thus obtain the following algorithm for computing query answering $\mathcal{H} \models q(\overline{a})$ in disjunctive $\mathcal{AL}$-LOG:

begin
  compute $\mathcal{M} = \{I : I \text{ stable model for } \mathcal{P}'\}$;
  while $\mathcal{M} \neq \emptyset$ **do**
    choose I' from $\mathcal{M}$;
    $\mathcal{M} = \mathcal{M} - \{I'\}$;
    if $(\mathcal{T}, \mathcal{A} \cup assert(I'))$ is $\mathcal{ALC}$-satisfiable
    then return false
  end;
  return true
end

Now, in order to implement the above algorithm, we only need to solve the problem of computing satisfiability of an $\mathcal{ALC}$ KB. Unfortunately, no implemented system can be used to this purpose: e.g., as mentioned above, the system FACT is only able to handle the case of empty ABoxes. In order to overcome this limitation, we have defined an encoding of concept and role assertions in terms of intensional inclusions in the language of FACT. That is, given an $\mathcal{ALC}$ KB $\Sigma = (\mathcal{T}, \mathcal{A})$, we define a polynomial transformation $\tau_T(\cdot)$ such that $\tau_T(\Sigma) = (\mathcal{T}', \emptyset)$ and prove that Σ is $\mathcal{ALC}$-satisfiable iff $\tau_T(\Sigma)$ is $\mathcal{ALC}$-satisfiable. Such a transformation is based on analogous encodings defined for propositional dynamic logic [De Giacomo, 1995]. The transformation

$\tau_T(\cdot)$ enables the use of FACT for computing the $\mathcal{ALC}$-satisfiability check in the above algorithm. We are thus able to provide an implementation of disjunctive $\mathcal{AL}$-LOG based on DLV and FACT: we only need to realize auxiliary procedures for computing the program $\mathcal{P}'$ and the transformation $\tau_T(\cdot)$.

References

[Buchheit *et al.*, 1993] Martin Buchheit, Francesco M. Donini, and Andrea Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research*, 1:109–138, 1993.

[Cadoli *et al.*, 1997] Marco Cadoli, Luigi Palopoli, and Maurizio Lenzerini. Datalog and description logics: Expressive power. In *Proc. of the 6th Int. Workshop on Database Programming Languages (DBPL'97)*, 1997.

[Calvanese *et al.*, 1998] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13, 1998.

[De Giacomo, 1995] Giuseppe De Giacomo. *Decidability of Class-Based Knowledge Representation Formalisms.* PhD thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1995.

[Donini *et al.*, 1998] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. AL-log: Integrating datalog and description logics. *J. of Intelligent Information Systems*, 10(3):227–252, 1998.

[Duschka and Genesereth, 1997] Oliver Duschka and Michael Genesereth. Query optimization using local completeness. In *Proc. of the 14th Nat. Conf. on Artificial Intelligence (AAAI'97)*, pages 249–255. AAAI Press/The MIT Press, 1997.

[Duschka and Genesereth, 1998] Oliver Duschka and Michael Genesereth. Query planning with disjunctive sources. In *Proc. of the AAAI Workshop on AI and Information Integration.* AAAI Press/The MIT Press, 1998.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Heikki Mannilla. Disjunctive datalog. *ACM Trans. on Database Systems*, 22(3):364–418, 1997.

[Eiter *et al.*, 1998] Thomas Eiter, Nicola Leone, Cristinel Mateis, Gerald Pfeifer, and Francesco Scarcello. The KR system dlv: Progress report, comparison and benchmarks. In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[Etzioni *et al.*, 1994] O. Etzioni, K. Golden, and D. Weld. Tractable closed-world reasoning with updates. In *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94).* Morgan Kaufmann, Los Altos, 1994.

[Horrocks and Patel-Schneider, 1999] I. Horrocks and P. Patel-Schneider. Optimising description logic subsumption. *J. of Logic and Computation*, 1999. To appear.

[Horrocks, 1998] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[Levy and Rousset, 1996] Alon Y. Levy and Marie-Christine Rousset. The limits on combining recursive Horn rules with description logics. In *Proc. of the 13th Nat. Conf. on Artificial Intelligence (AAAI'96)*, pages 577–584, 1996.

[Levy *et al.*, 1996] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Query answering algorithms for information agents. In *Proc. of the 13th Nat. Conf. on Artificial Intelligence (AAAI'96)*, pages 40–47, 1996.

[Niemelä and Simons, 1997] I. Niemelä and P. Simons. Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In *Proc. of the 4th Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'97)*, volume 1265 of *Lecture Notes In Artificial Intelligence*, pages 420–429. Springer-Verlag, 1997.