Compilation and Approximation of Conjunctive Queries by Concept Descriptions

François Goasdoué and Marie-Christine Rousset
LRI, CNRS & Université Paris-Sud, France
Email: {fg,mcr}@lri.fr

1 Introduction

Conjunctive queries play a central role in database and in knowledge representation. They correspond to the core relational database queries (i.e., SelectProjectJoin queries). They are the basic elements of function-free Horn rule languages that are a decidable subset of first order logic for which several practically efficient inference procedures have been developed and used in many applications. Description logics are another decidable subset which has been extensively studied. They have been designed especially to model rich hierarchies of classes of objects. In particular, they have been found very useful for reasoning on ontologies [15] and for expressing web-site semantics in the next generation "semantic web" [5]. Horn rules and descriptions logics are two orthogonal subsets of first order logic: it is known [6] that neither of these languages can express the other. Their combination has been studied in Al-log [13] and in CARIN [17].

In contrast, in this paper, we focus on the logical overlapping existing between conjunctive queries and concept descriptions. In our comparison, since description logics only deal with unary and binary relations, we restrict the conjunctive queries that we consider to be made of unary and binary atoms only. However, we extend the expressive power of standard conjunctive queries by allowing unary atoms $\mathsf{C}(U)$ such that C is a concept description.

The contribution of this paper is twofold. First, we exhibit a necessary and sufficient condition for the *compilation* of a conjunctive query that is unary (i.e., with a single distinguished variable) into an equivalent concept description in $\mathcal{ALE}$. Second, we provide a necessary and sufficient condition for the *approximation* of unary conjunctive queries by maximally subsumed $\mathcal{ALN}$ concept descriptions.

Compiling or approximating conjunctive queries into concept descriptions enable the transfer of complexity results and algorithms between description logics from one side, and Horn rules or relational database theory from the other side. Characterizing $\mathcal{ALE}$ descriptions that are equivalent to conjunctive queries for which checking containment is polynomial provides a restriction of $\mathcal{ALE}$ for which subsumption is polynomial. Approximating conjunctive queries by $\mathcal{ALN}$ descriptions for which subsumption is polynomial provides an incomplete but efficient procedure for checking containment of conjunctive queries which is in general NP-complete.

The paper is organized as follows. In section 2, we present within a uniform logical framework the basic notions on the description logics and the conjunctive queries that we consider. In section 3, we characterize precisely the conjunctive queries that can be compiled (i.e., equivalently translated) into concept descriptions. Section 4 is devoted to the approximation of conjunctive queries by concept descriptions. Finally, in Section 5, we compare our work with related work and we draw some conclusions.

2 Preliminaries

In this section, we introduce the syntax and logical semantics of (i) the concept descriptions (Section 2.1), and (ii) the conjunctive queries (Section 2.2) which we consider in this paper. The resulting subsumption relation applies indiscriminately to concept descriptions and unary conjunctive queries.

2.1 Concept descriptions

In description logics, concept descriptions are inductively defined, using a set of *constructors*, starting with a set P_c of unary predicates and a set P_r of binary predicates. An element of P_c is called an atomic concept and an element of P_r is called a role. The set of constructors allowed for building concept descriptions varies from one description logic to another. The concept descriptions that we consider in this paper are inductively defined using constructors of the $\mathcal{ALEN}$ description logic as follows:
- any atomic concept $\mathsf{A} \in N_P$ is a concept description,
- $\top$ is a concept description (called *top* or universal concept),
- $\bot$ is a concept description (called *bottom* or empty concept),
- if A is an atomic concept, then $\neg\mathsf{A}$ is a concept description (called *atomic negation*),
- if C and D are concept descriptions, then $(\mathsf{C} \sqcap \mathsf{D})$ is a concept description (called *conjunction* of C and D),
- if C is a concept description and r is a role, then $(\forall \mathsf{r}.\mathsf{C})$ and $(\exists \mathsf{r}.\mathsf{C})$ are concept descriptions (respectively called *value restriction* and *existential restriction*), and
- if n is an integer and r is a role, then $(\geq n\ \mathsf{r})$ and $(\leq n\ \mathsf{r})$ are concept descriptions (called *number restrictions*).

The logical semantics of concept descriptions is defined in terms of *interpretations*. An interpretation I is a pair $(\Delta^I, .^I)$ where Δ^I is a non-empty set of individuals called the domain of interpretation of I, and $.^I$ is an interpretation function, which assigns a subset of Δ^I to every concept description. An interpretation function maps each atomic concept $\mathsf{A} \in N_P$ to a subset A^I of Δ^I and each role $\mathsf{r} \in N_R$ to a subset r^I of $\Delta^I \times \Delta^I$. The interpretation of an arbitrary concept description is inductively defined as follows:
- $\top^I = \Delta^I$, $\bot^I = \emptyset$, $(\neg \mathsf{A})^I = \Delta^I \setminus \mathsf{A}^I$, $(\mathsf{C} \sqcap \mathsf{D})^I = \mathsf{C}^I \cap \mathsf{D}^I$,
- $(\forall \mathsf{r}.\mathsf{C})^I = \{d \in \Delta^I \mid \forall e \in \Delta^I \; ((d,e) \in \mathsf{r}^I \to e \in \mathsf{C}^I)\}$,
- $(\exists \mathsf{r}.\mathsf{C})^I = \{d \in \Delta^I \mid \exists e \in \Delta^I \text{ s.t } \; (d,e) \in \mathsf{r}^I \text{ and } e \in \mathsf{C}^I\}$,
- $(\geq n \; \mathsf{r})^I = \{d \in \Delta^I \mid \#\{e \mid (d,e) \in \mathsf{r}^I\} \geq n\}$ and
- $(\leq n \; \mathsf{r})^I = \{d \in \Delta^I \mid \#\{e \mid (d,e) \in \mathsf{r}^I\} \leq n\}$.

An interpretation I is a model of a concept description C iff C^I is not empty: C is then said to be *satisfiable*.

In this paper, we mention different sublanguages of $\mathcal{ALEN}$:
• $\mathcal{AL}$, in which concept descriptions are restricted to be top, bottom, atomic negation, conjunction and value restriction.
• $\mathcal{ALN}$, which enriches $\mathcal{AL}$ with number restrictions.
• $\mathcal{ALE}$, which enriches $\mathcal{AL}$ with existential restriction.
• We introduce the new sublanguages of $\mathcal{ALEN}$, denoted by $\mathcal{AL}$+$\mathcal{E}$ and $\mathcal{ALN}$+$\mathcal{E}$, limiting the interplay of the existential restriction with the other constructors in the following way:
- If C is in $\mathcal{AL}$ (respectively $\mathcal{ALN}$) then C is in $\mathcal{AL}$+$\mathcal{E}$ (respectively $\mathcal{ALN}$+$\mathcal{E}$).
- If C is in $\mathcal{AL}$+$\mathcal{E}$ (respectively $\mathcal{ALN}$+$\mathcal{E}$) then $\exists \mathsf{r}.\mathsf{C}$ is in $\mathcal{AL}$+$\mathcal{E}$ (respectively $\mathcal{ALN}$+$\mathcal{E}$).
- If C_1 and C_2 are in $\mathcal{AL}$+$\mathcal{E}$ (respectively $\mathcal{ALN}$+$\mathcal{E}$) then $\mathsf{C}_1 \sqcap \mathsf{C}_2$ is in $\mathcal{AL}$+$\mathcal{E}$ (respectively $\mathcal{ALN}$+$\mathcal{E}$).

2.2 Conjunctive queries

The conjunctive queries that we consider in this paper are unary conjunctive queries over unary and binary atoms. They can be purely relational or hybrid with description logics. Given two sets P_c and P_r of unary and binary predicates, a *unary conjunctive query* over $P_c \cup P_r$, denoted by $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$, defines a unary predicate q by a conjunction $\mathcal{CJ}(X, \bar{Y})$ of unary and binary atoms. X is called the *distinguished variable*, and the variables of $\bar{Y}$ are called the *existential variables* of the query. $\mathcal{CJ}(X, \bar{Y})$ is the *body* of the query. The unary conjuncts in $\mathcal{CJ}(X, \bar{Y})$ are atoms of the form $\mathsf{C}(U)$ where U is the distinguished variable or an existential variable, and C is a concept description over $P_c \cup P_r$. The binary conjuncts in $\mathcal{CJ}(X, \bar{Y})$ are atoms of the form $\mathsf{r}(U_1, U_2)$ where U_1 and U_2 are (distinguished or existential) variables and r is a binary predicate of P_r.

For a given variable V, we call its *unary scope* in q, denoted by $UnaryScope_{\mathsf{q}}(V)$, the conjunction of unary atoms of the form $\mathsf{C}(V)$ appearing in the body of q. If there is no unary atom of the form $\mathsf{C}(V)$, the unary scope of V is set to $\top(V)$.

If every unary conjunct $\mathsf{C}(U)$ of a conjunctive query is such that C is an atomic concept (i.e., $\mathsf{C} \in P_c$), this query is said to be *pure*. If a conjunctive query has unary conjuncts $\mathsf{C}(U)$ such that C is a concept description which is not atomic of a description logic LD, then it is said to be *hybrid* with LD.

Example 1 q_1 is a pure conjunctive query that defines flights whose destination is an American city with a stop in an European City: $\mathsf{q}_1(X) : \mathsf{Flight}(X) \wedge \mathsf{Stop}(X, Y_1) \wedge \mathsf{EuropeanCity}(Y_1) \wedge \mathsf{Destination}(X, Y_2) \wedge \mathsf{AmericanCity}(Y_2)$.
q_2 is an hybrid conjunctive query that defines flights with a stop in an European City and atmost 2 stops: $\mathsf{q}_2(X) : (\mathsf{Flight} \sqcap (\leq 2\ \mathsf{Stop}))(X) \wedge \mathsf{Stop}(X, Y_1) \wedge \mathsf{EuropeanCity}(Y_1)$.
Finally q_3 is a pure conjunctive query that defines airlines that share the service of an European city Y_1 with an American airline U: $\mathsf{q}_3(X) : \mathsf{Airline}(X) \wedge \mathsf{Service}(X, Y_1) \wedge \mathsf{EuropeanCity}(Y_1) \wedge \mathsf{Airline}(U) \wedge \mathsf{AmericanCompany}(U) \wedge \mathsf{Service}(U, Y_1)$.

In the same way that an interpretation of atomic concepts and roles determines the interpretation of complex concept descriptions, the interpretation of the unary and binary predicates appearing in the query definition uniquely determines the interpretation of the query.

Definition 1 (Semantics of a conjunctive query) *Let q be a unary predicate defined by a conjunctive query $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$. Given an interpretation I, the* interpretation *q^I of q in I is defined as follows: $o \in \mathsf{q}^I$ iff there exists a mapping α from the query's variables to the domain Δ^I such that: $\alpha(X) = o$, $\alpha(U) \in \mathsf{C}^I$ for every unary atom $\mathsf{C}(U)$ in $\mathcal{CJ}(X, \bar{Y})$, and $\alpha(U_1, U_2) \in \mathsf{r}^I$ for every binary atom $\mathsf{r}(U_1, U_2)$ in $\mathcal{CJ}(X, \bar{Y})$.*

The following definition of subsumption and equivalence applies to unary predicates defined by conjunctive queries or concept descriptions.

Definition 2 (Subsumption and equivalence) *Let q_1 and q_2 be concept descriptions or unary predicates defined by conjunctive queries.*
- q_1 is subsumed *by q_2 (denoted $\mathsf{q}_1 \preceq \mathsf{q}_2$) iff for every interpretation I, $\mathsf{q}_1^I \subseteq \mathsf{q}_2^I$.*
- q_1 is equivalent *to q_2 (denoted $\mathsf{q}_1 \equiv \mathsf{q}_2$) iff for every interpretation I, $\mathsf{q}_1^I = \mathsf{q}_2^I$.*

Example 2 Consider again the query q_1 of Example 1. It is equivalent to $\mathsf{Flight} \sqcap (\exists \mathsf{Stop}.\mathsf{EuropeanCity}) \sqcap (\exists \mathsf{Destination}.\mathsf{AmericanCity})$ and subsumes (without being equivalent to) $\mathsf{Flight} \sqcap (\geq 1\ \mathsf{Stop}) \sqcap (\forall \mathsf{Stop}.\mathsf{EuropeanCity}) \sqcap (\geq 1\ \mathsf{Destination}) \sqcap (\forall \mathsf{Destination}.\mathsf{AmericanCity})$.

3 Translation between conjunctive queries and concept descriptions

We start by introducing the notions of *binding graph* and *restriction* for conjunctive queries, which play a central role in the logical correspondence between conjunctive queries and concept descriptions. The *binding graph* accounts for the connection existing between variables within a conjunctive query. The *restriction* of a query results from equating some variables of the query. We also introduce the notions of *tree*, *forest*, *dag* and *connected queries.*

Definition 3 (Predecessors and successors) *Let* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$ *be a conjunctive query. Let* U_1 *and* U_2 *be two variables of* q.
- U_1 *is a* r-predecessor of U_2 *and* U_2 *is a* r-successor of U_1 *iff the binary atom* $\mathsf{r}(U_1, U_2)$ *is a conjunct of* $\mathcal{CJ}(X, \bar{Y})$.
- U_1 *is a* predecessor of U_2 *(respectively* U_1 *is a* successor of U_2*) iff there exists a binary predicate* r *such that* U_1 *is a* r-predecessor of U_2 *(respectively* U_1 *is a* r-successor of U_2*).*

Definition 4 (Binding graph of a conjunctive query) *Let* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$ *be a conjunctive query. Its* binding graph *(denoted by* $\mathcal{G}(\mathsf{q})$*) is a directed graph defined as follows: the nodes of* $\mathcal{G}(\mathsf{q})$ *are the variables of* q *and there exists an edge labelled by* r *from* U_1 *to* U_2 *in* $\mathcal{G}(\mathsf{q})$ *iff* U_1 *is a* r*-predecessor of* U_2 *in* q.

Definition 5 (Restriction of a conjunctive query) *Let* q *be a conjunctive query defined by* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$. *A* restriction *of* q *is a conjunctive query* q' *defined by* $\mathsf{q}'(X) : \sigma.\mathcal{CJ}(X, \bar{Y})$ *where* σ *is a mapping equating some existential variables of* q *with some (distinguished or existential) variables of* q.

Example 3 Consider the following conjunctive query: $\mathsf{q}'_3(X) : \mathsf{Airline}(X) \wedge \mathsf{Service}(X, Y_1) \wedge \mathsf{EuropeanCity}(Y_1) \wedge \mathsf{AmericanCompany}(X)$. q'_3 is a restriction of the query q_3 in Example 1, which is obtained by equating U with X.

The next proposition results directly from Definition 5.

Proposition 1 *Let* q *and* q' *be two conjunctive queries. If* q' *is a restriction of* q*, then* q' *is subsumed by* q.

Definition 6 (Tree, forest, dag and connected queries) *Let* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$ *be a conjunctive query. It is a* tree query *iff its binding graph is a tree rooted in* X. *It is a* forest query *iff its binding graph is a forest where a tree is rooted in* X. *It is a* dag query *iff its binding graph is acyclic*[1]. *It is a* connected query *iff its binding graph is connected.*

[1]The so-called acyclic conjunctive queries in database theory (e.g., [14]) are different from our dag queries because they are based on non directed graphs. For example, the dag query $\mathsf{q}(X) : \mathsf{r}(X, Y_1) \wedge \mathsf{r}(X, Y_2) \wedge \mathsf{r}(Y_3, Y_1) \wedge \mathsf{r}(Y_3, Y_2)$ is cyclic in the sense of [14].

In Example 1, q_1 and q_2 are tree queries while q_3 is not.
In Section 3.1, we exhibit a one-to-one correspondence between tree queries and concept descriptions, which is exploited to transfer a polynomial subsumption algorithm from tree queries to $\mathcal{AL}$+$\mathcal{E}$ concept descriptions. In Section 3.2, we show that unary conjunctive queries having an equivalent restriction which is a tree query are exactly the conjunctive queries that can be equivalently compiled into concept descriptions.

3.1 Tree queries and concept descriptions

Before defining the *translation* of a tree query into a concept description and the *expansion* of a concept description into a tree query, we start by introducing useful notations.

Notation: For every existential variable Y of a tree query q, we denote by r_Y and V_Y respectively the single binary predicate r and the single (existential or distinguished) variable V such that $\mathsf{r}(V, Y)$ is an atom of q. Given a (distinguished or existential) variable V of a tree query q, we denote by $subtree_{\mathsf{q}}(V)$ the conjunction of the atoms in the body of q inductively defined as follows:
- if V has no successor: $subtree_{\mathsf{q}}(V) = UnaryScope_{\mathsf{q}}(V)$,
- if V has $V_1, \ldots, V_n$ as successors: $subtree_{\mathsf{q}}(V) =$

$$UnaryScope_{\mathsf{q}}(V) \wedge \bigwedge_{i=1}^{n} \left[\mathsf{r}_{V_i}(V, V_i) \wedge subtree_{\mathsf{q}}(V_i)\right].$$

Definition 7 (Translation of a tree query into $\mathcal{ALEN}$) *Let* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$ *be a tree query. The* translation *of* q *in* $\mathcal{ALEN}$*, denoted* $\mathsf{Compil}(\mathsf{q})$*, is the concept description inductively defined as follows:*
- if $\mathcal{CJ}(X, \bar{Y}) = \bigwedge_{i=1}^{k} \mathsf{C}_i(X)$ *(X has no successor) then:* $\mathsf{Compil}(\mathsf{q}) = \bigsqcap_{i=1}^{k} \mathsf{C}_i$ *;*
- if $\mathcal{CJ}(X, \bar{Y}) = \bigwedge_{i=1}^{k} \mathsf{C}_i(X) \wedge \bigwedge_{j=1}^{n} \mathsf{r}_{Y_j}(X, Y_j) \wedge subtree_{\mathsf{q}}(Y_j)$*, where* $Y_1, \ldots, Y_n$ *are the successors of* X*, then:* $\mathsf{Compil}(\mathsf{q}) = \bigsqcap_{i=1}^{k} \mathsf{C}_i \sqcap \bigsqcap_{j=1}^{n} (\exists \mathsf{r}_{Y_j}.\mathsf{Compil}(subtree_{\mathsf{q}}(Y_j)))$.

Example 4 Consider the tree query q_1 in Example 1.
$\mathsf{Compil}(\mathsf{q}_1) = (\mathsf{Flight} \sqcap (\exists \mathsf{Stop.EuropeanCity}) \sqcap (\exists \mathsf{Destination.AmericanCity}))$.

Definition 8 (Expansion of a concept description) *The* expansion *rooted in a variable* X *of an* $\mathcal{ALEN}$ *concept description* C*, denoted by* $\mathsf{Expand}(\mathsf{C}, X)$*, is the conjunction of atoms inductively defined as follows:*
- If C *has the form of an atomic concept, a value restriction or a number restriction:* $\mathsf{Expand}(\mathsf{C}, X) = \mathsf{C}(X)$*.*
- If $\mathsf{C} = \mathsf{C}_1 \sqcap \mathsf{C}_2$*:* $\mathsf{Expand}(\mathsf{C}, X) = \mathsf{Expand}(\mathsf{C}_1, X) \wedge \mathsf{Expand}(\mathsf{C}_2, X)$*.*
- If $C = (\exists \mathsf{r.D})$*:* $\mathsf{Expand}(\mathsf{C}, X) = \mathsf{r}(X, Y) \wedge \mathsf{Expand}(\mathsf{D}, Y)$*, where* Y *is a fresh variable.*

The next proposition is straightforward. It establishes the equivalence between a tree query and its translation on one hand, and a concept description and its expansion (which is a tree query) on the other hand.

Proposition 2 *Let* $\mathsf{q}(X) : \mathcal{CJ}(X, \bar{Y})$ *be a tree query and* $\mathsf{q}'(X) : \mathsf{Expand}(\mathsf{Compil}(\mathsf{q}), X)$ *be the tree query defined by the expansion of its compilation:* $\mathsf{Compil}(\mathsf{q}) \equiv \mathsf{q} \equiv \mathsf{q}'$. *Let* C *be an* $\mathcal{ALEN}$ *concept description and* $\mathsf{q}(X)$:$\mathsf{Expand}(\mathsf{q}, X)$ *be the tree query defined by its expansion:* $\mathsf{C} \equiv \mathsf{q} \equiv \mathsf{Compil}(\mathsf{q})$.

Theorem 1 is based on an extension of the homomophism theorem for relational conjunctive queries [9]. It is a direct adaptation of the extension of the homomophism theorem for conceptual graphs [19, 10] pointed in [3] for establishing the correspondence between conceptual graphs and description logics. The result comes from the combination of two known results [16, 12] : (i) checking whether there exists a homomorphism from a tree onto a graph, and (ii) checking subsumption of two $\mathcal{AL}$ concept descriptions can be done in polynomial time. It is important to note that for guaranteeing in our setting the equivalence between checking query subsumption and checking the existence of an homomorphism between the corresponding binding graphs, we need to saturate the conjunctive queries, hybrid with $\mathcal{AL}$, by applying exhaustively the rule: $\mathsf{r}(U, V) \wedge (\forall \mathsf{r}.\mathsf{C})(U) \rightarrow \mathsf{r}(U, V) \wedge (\forall \mathsf{r}.\mathsf{C})(U) \wedge \mathsf{C}(V)$. The important point is that the size of the saturated queries remains polynomial in the size of the original queries.

Theorem 1 *Let* q *be a tree query and* q' *be any conjunctive query over* $\mathcal{AL}$ *descriptions. Checking whether* q' *is subsumed by* q *can be done in polynomial time.*

The next corollary is a direct consequence of Theorem 1 and Proposition 2, exploiting the fact that any concept description of $\mathcal{AL}$+$\mathcal{E}$ can be expanded in polynomial time into an equivalent tree query over $\mathcal{AL}$ descriptions.

Corollary 1 $\mathcal{AL}$+$\mathcal{E}$ *is a sublanguage of* $\mathcal{ALE}$ *for which subsumption is polynomial.*

3.2 Conjunctive queries compilable into concept descriptions

It has been stated in Proposition 2 that being a tree query is a sufficient condition for a conjunctive query to be compiled into an equivalent concept description. Theorem 2 establishes a necessary condition for a conjunctive query to subsume an $\mathcal{ALEN}$ concept description. Based on that theorem, Theorem 3 then establishes a necessary condition for a conjunctive query to be equivalent to a concept description of $\mathcal{ALE}$ or $\mathcal{ALN}$. As a consequence of Proposition 2 and Theorem 3, Corollary 2 establishes a *necessary and sufficient condition* for a conjunctive query to be compiled into an equivalent $\mathcal{ALE}$ concept description.

Theorem 2 *Let* C *be a satisfiable* $\mathcal{ALEN}$ *concept description, and* q *a conjunctive query. If* $\mathsf{C} \preceq \mathsf{q}$*, then* q *or one of its restriction is a forest query.*

Sketch of proof: The full proof is given in [1]. It is based on the completion calculus and reuses known results from [17]. In particular, since $\mathsf{C} \preceq \mathsf{q}$ is equivalent to $\mathsf{C}(X) \models \mathsf{q}(X)$ then for any clash free completion S of $\mathsf{C}(X)$:
- there exists a mapping α from the variables appearing in q into the variables of S such that $\alpha(X) = X$, if $\mathsf{r}(U_1, U_2)$ is in q then $\mathsf{r}(\alpha(U_1), \alpha(U_2)) \in S$ and if $\mathsf{C}(U)$ is in q then $S \models \mathsf{C}(\alpha(U))$, and
- the binding graph of S is a tree rooted in X.
First, we show that the binding graph of q is a dag: if it had a cycle, then the binding graph of any completion S would have a cycle too. This would contradict that the graph of S is a tree. Then, we show that X cannot have a predecessor in q since X would have a predecessor in the binding graph of S. This would contradict that the graph of S is a tree rooted in X. At this point, it is easy to see that if every (existential) node has atmost one predecessor, then q is a forest query. In the case where an existential node has more than one predecessor, the roles labelling the corresponding incoming edges are necessarily the same. It can then be proved by induction that iteratively equating the predecessors of such nodes defines a restriction of q which is a forest query. $\square$

Theorem 3 *Let* C *be a satisfiable* $\mathcal{ALE}$ *or* $\mathcal{ALN}$ *concept description, and* q *a conjunctive query: if* $\mathsf{C} \equiv \mathsf{q}$*, then* q *or one of its equivalent restriction is a tree query.*

Sketch of proof: The full proof is given in [1]. It is based on the above proof, and on the fact that a unique completion is produced by the standard completion calculus in $\mathcal{ALE}$ and by the non standard completion calculus in $\mathcal{ALN}$ introduced in [4]. Since $\mathsf{C} \equiv \mathsf{q}$ is equivalent to $\mathsf{C}(X) \equiv \mathsf{q}(X)$, there exists two mappings α_1 and α_2 from the clash free completion S_{C} of $\mathsf{C}(X)$ to the clash free completion S_{q} of $\mathcal{CJ}(X, \bar{Y})$ for the former and from the clash free completion S_{q} of $\mathcal{CJ}(X, \bar{Y})$ to the clash free completion S_{C} of $\mathsf{C}(X)$ for the latter, both of these mappings being defined as α is, in the above sketch of proof. According to Theorem 2, q or one of its restriction is a forest query. If q is a forest query, it is easy to see that the binding graph of S_{C} is mapped to a subgraph of the binding graph of S_{q} which is a tree rooted in X. We then prove, using the completion calculus dedicated to CARIN [17], that equating each node v, which does not belong to that tree, to $\alpha_1(\alpha_2(v))$ defines an equivalent restriction of q which is a tree query. If q is not a forest query, according to Theorem 2, there exists a restriction of q which is a forest query and subsumes C. Performing the same reasoning as above, we show that this restriction has an equivalent restriction which is a tree query. $\square$

Corollary 2 *Let* q *be a (pure or hybrid with* $\mathcal{ALE}$*) conjunctive query:* q *is equivalent to an* $\mathcal{ALE}$ *concept description iff* q *or one of its equivalent restriction is a tree query.*

Example 5 Consider the following dag query: $\mathsf{q}_5(X) : \mathsf{Flight}(X) \wedge \mathsf{Destination}(X, Y_1) \wedge \mathsf{AmericanCity}(Y_1) \wedge \mathsf{Stop}(X, Y_2) \wedge \mathsf{Flight}(Y_3) \wedge \mathsf{Stop}(Y_3, Y_2)$. Its equivalent restriction q'_5 obtained by equating Y_3 with X is the following tree query: $\mathsf{q}'_5(X) : \mathsf{Flight}(X) \wedge \mathsf{Destination}(X, Y_1) \wedge \mathsf{AmericanCity}(Y_1) \wedge \mathsf{Stop}(X, Y_2)$. It is equivalent to the $\mathcal{ALE}$ concept description: $\mathsf{Flight} \sqcap (\exists \mathsf{Destination.AmericanCity}) \sqcap (\exists \mathsf{Stop}.\top)$.

As for the conjunctive queries that can be equivalently compiled into $\mathcal{ALN}$ concept descriptions, they are very limited. In particular, each binary atom in the body of those queries must be of the form $\mathsf{r}(X, Y)$ where Y is an existential variable not appearing elsewhere in the body of the query. This results from Theorem 3 and Proposition 2, and from the fact (shown in [1]) that a satisfiable $\mathcal{ALN}$ concept description cannot be equivalent to a description of the form $\bigsqcap_{i=1}^{p}(\exists \mathsf{r}_i.\mathsf{C}_i)$ except if every C_i is equivalent to $\top$.

In the next section, we characterize the conjunctive queries that can be approximated in $\mathcal{ALN}$, i.e., for which we can find a maximally subsumed $\mathcal{ALN}$ concept description.

4 Approximation of conjunctive queries in $\mathcal{ALN}$

We start by defining the (weak and strong) approximation of $\mathcal{ALEN}$ concept descriptions by $\mathcal{ALN}$ concept descriptions.

Definition 9 (Approximation of concept descriptions) *Let* D *be an* $\mathcal{ALEN}$ *concept description. Its* weak *and* strong approximations *in* $\mathcal{ALN}$*, respectively denoted* $\mathsf{Approx}_{\uparrow}(\mathsf{D})$ *and* $\mathsf{Approx}_{\downarrow}(\mathsf{D})$*, are inductively defined as follows:*

- *if* $\mathsf{D} = \top$, $\mathsf{D} = \bot$, $\mathsf{D} = \mathsf{A}$, $\mathsf{D} = \neg\mathsf{A}$, $\mathsf{D} = (\geq n\ \mathsf{r})$ *or* $\mathsf{D} = (\leq n\ \mathsf{r})$*, then:* $\mathsf{Approx}_{\uparrow}(\mathsf{D}) = \mathsf{Approx}_{\downarrow}(\mathsf{D}) = \mathsf{D}$ *;*
- *if* $\mathsf{D} = (\mathsf{D}_1 \sqcap \mathsf{D}_2)$ *then:* $\mathsf{Approx}_{\uparrow}(\mathsf{D}) = \mathsf{Approx}_{\uparrow}(\mathsf{D}_1) \sqcap \mathsf{Approx}_{\uparrow}(\mathsf{D}_2)$
  $\mathsf{Approx}_{\downarrow}(\mathsf{D}) = \mathsf{Approx}_{\downarrow}(\mathsf{D}_1) \sqcap \mathsf{Approx}_{\downarrow}(\mathsf{D}_2)$;
- *if* $\mathsf{D} = (\forall \mathsf{r}.\mathsf{D}_1)$ *then:* $\mathsf{Approx}_{\uparrow}(\mathsf{D}) = (\forall \mathsf{r}.\mathsf{Approx}_{\uparrow}(\mathsf{D}_1))$
  $\mathsf{Approx}_{\downarrow}(\mathsf{D}) = (\forall \mathsf{r}.\mathsf{Approx}_{\downarrow}(\mathsf{D}_1))$;
- *if* $\mathsf{D} = (\exists \mathsf{r}.\mathsf{D}_1)$ *then:* $\mathsf{Approx}_{\uparrow}(\mathsf{D}) = (\geq 1\ \mathsf{r}) \sqcap (\forall \mathsf{r}.\mathsf{Approx}_{\uparrow}(\mathsf{D}_1))$
  $\mathsf{Approx}_{\downarrow}(\mathsf{D}) = (\geq 1\ \mathsf{r})$.

Proposition 3 results trivially from the semantics of the different constructors used in $\mathcal{ALEN}$. Theorem 4 states that the weak approximation is in fact the greatest lower bound for $\mathcal{ALN}$+$\mathcal{E}$ concept descriptions.

Proposition 3 $\mathsf{Approx}_{\uparrow}(\mathsf{D}) \preceq \mathsf{D} \preceq \mathsf{Approx}_{\downarrow}(\mathsf{D})$ *for every* D *in* $\mathcal{ALEN}$*.*

Theorem 4 *Let* D *be an* $\mathcal{ALN}$+$\mathcal{E}$ *concept description and* C *an* $\mathcal{ALN}$ *concept description s.t* $\mathsf{C} \preceq \mathsf{D}$*:* $\mathsf{C} \preceq \mathsf{Approx}_{\uparrow}(\mathsf{D})$*.*

The full proof is given in [1]. It shows that if an $\mathcal{ALN}$ concept description C is subsumed by $\sqcap_{i=1}^{p}(\exists \mathsf{r}_i.\mathsf{C}_i)$ then it is necessarily subsumed by $\sqcap_{i=1}^{p}(\geq 1\ \mathsf{r}_i) \sqcap (\forall \mathsf{r}_i.\mathsf{C}_i)$. It exploits that the canonical interpretation of each clash-free completion of $C(X)$ is a model of every $(\exists \mathsf{r}_i.\mathsf{C}_i)(X)$, and must have been produced by one of the completion rules for $\mathcal{ALN}$.

We now define the approximation of tree queries by $\mathcal{ALN}$ concept descriptions. Corollary 3 results from Proposition 3 and Theorem 4.

Definition 10 (Approximation of tree queries in $\mathcal{ALN}$) *Let* q *be a tree query.* $\mathsf{Approx}_{\uparrow}(\mathsf{q}) = \mathsf{Approx}_{\uparrow}(\mathsf{Compil}(\mathsf{q}))$ *and* $\mathsf{Approx}_{\downarrow}(\mathsf{q}) = \mathsf{Approx}_{\downarrow}(\mathsf{Compil}(\mathsf{q}))$*.*

Corollary 3 *Let* q *be a tree query:* $\mathsf{Approx}_{\uparrow}(\mathsf{q}) \preceq \mathsf{q} \preceq \mathsf{Approx}_{\downarrow}(\mathsf{q})$*. If* q *is hybrid with* $\mathcal{ALN}$ *and* C *is an* $\mathcal{ALN}$ *concept such that* $C \preceq \mathsf{q}$ *then:* $\mathsf{C} \preceq \mathsf{Approx}_{\uparrow}(\mathsf{q})$*.*

Corollary 4 provides a necessary and sufficient condition for a conjunctive query to subsume an $\mathcal{ALN}$ concept description. It results from Theorem 2, Corollary 3 and Proposition 1.

Corollary 4 *Let* q *be a conjunctive query:* q *subsumes a satisfiable* $\mathcal{ALN}$ *concept description iff* q *or one of its restriction is a forest query.*

A particular case concerns the *connected* queries q (possibly hybrid with $\mathcal{ALN}$): q subsumes a satisfiable $\mathcal{ALN}$ concept description iff q or one of its restriction is a tree query ; the greatest lower bound in $\mathcal{ALN}$ is then the weak approximation of the corresponding tree query.

For q and q' being $\mathcal{ALEN}$ concept descriptions or conjunctive queries compilable into $\mathcal{ALEN}$ concept descriptions, their approximations can be used to improve the efficiency of checking whether q is subsumed q' as follows: (1) if $\mathsf{Approx}_{\downarrow}(\mathsf{q}) \preceq \mathsf{Approx}_{\uparrow}(\mathsf{q}')$, the answer is "yes", (2) otherwise if $\mathsf{Approx}_{\uparrow}(\mathsf{q}) \not\preceq \mathsf{Approx}_{\downarrow}(\mathsf{q}')$, the answer is "no". That procedure takes only polynomial time since checking subsumption in $\mathcal{ALN}$ is polynomial. In case no answer is obtained, the procedure could return "don't know" or it could call an (exponential in the worst case) algorithm checking subsumption for $\mathcal{ALEN}$. If q' is (compilable into) an $\mathcal{ALN}$+$\mathcal{E}$ concept description, based on Theorem 4, the previous step (2) can be replaced by: $(2)'$ otherwise if $\mathsf{Approx}_{\uparrow}(\mathsf{q}) \not\preceq \mathsf{Approx}_{\uparrow}(\mathsf{q}')$ the answer is "no".

5 Conclusion and perspectives

This paper is the first work on the approximation of conjunctive queries by concept descriptions. It has been motivated by the problem of rewriting conjunctive queries using views in an information integration setting. In particular, Theorem 4 has been the key ([1]) to prove the completeness of an algorithm computing the maximal rewritings of conjunctive queries (hybrid with $\mathcal{ALN}$) using $\mathcal{ALN}$ views.

Existing work related to approximation in the first-order case has dealt with either approximating concept descriptions by simpler concept descriptions ([8, 2, 7]), or with approximating sets of clauses by tractable sets of clauses ([18, 11]). In [8], concept descriptions of $\mathcal{ALE}$ (respectively of $\mathcal{ALC}$) are approximated by sequences of simpler concept descriptions of the same description logic, i.e., $\mathcal{ALE}$ (respectively $\mathcal{ALC}$). In [2], lower approximations of concept descriptions are defined w.r.t a given description logic *and* a terminology. The problem addressed in [7] is the computation for $\mathcal{ALC}$ concept descriptions of their upper approximation in $\mathcal{ALE}$.
As for the compilation preserving equivalence, in [3], an equivalent translation is provided from the $\mathcal{ELIRO}1$ description logics into a subclass of rooted simple conceptual graphs. However, they do not provide a necessary condition for a rooted simple conceptual graph (i.e., a unary conjunctive query) to be equivalently compilable into a concept description, as we do in Theorem 3.

This current work can be pursued in several directions. We plan to investigate the study of minimal upper bounds for conjunctive queries. We also plan to extend the current work by considering unions of conjunctive queries and their approximation in $\mathcal{ALC}$.

References

[1] F. Goasdoué, 'Réécriture de requêtes en termes de vues dans CARIN et intégration d'informations', PhD, 2001.

[2] F. Baader, R. Küsters, and R. Molitor, 'Rewriting concepts using terminologies', in *Proceedings of KR 2000.*

[3] F. Baader, R. Molitor, and S. Tobies, 'Tractable and decidable fragments of conceptual graphs', in *Proceedings of ICCS'99.*

[4] C. Beeri, A. Y. Levy, and M-C. Rousset, 'Rewriting queries using views in description logics', in *Proceedings of PODS 97.*

[5] T. Berners-Lee, J. Hendler, and O. Lassila, 'The semantic web', *Scientific American, 279*, (2001).

[6] A. Borgida, 'On the relative expressiveness of description logics and predicate logics', *Artificial Intelligence*, **82**(1-2).

[7] S. Brandt, R. Küsters, and A-Y. Turhan, 'Approximation and difference in description logics', LTCS-Report LTCS-01-06.

[8] M. Cadoli and M. Schaerf, 'Approximation in concept description languages', in *Proceedings of KR 92*.

[9] A.K Chandra and P.M Merlin, 'Optimal implementation on conjunctive queries in relational data bases', in *Proceedings ACM Symposium on the Theory of Computing*, (1977).

[10] M. Chein and M-L. Mugnier, 'Conceptual graphs: fundamental notions', *Revue d'Intelligence Artificielle*, **6**(4), (1992).

[11] A. del Val, 'Approximate knowledge compilation: the first order case', in *Proceedings of AAAI 96*.

[12] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, 'Reasoning in description logics', in *Foundation of Knowledge Representation*, ed., Gerhard Brewka, CSLI-Publications, (1996).

[13] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, 'Al-log: integrating datalog and description logics', *Journal of Intelligent Information Systems*, **10**, (1998).

[14] G. Gottlob, N. Leone, and F. Scarcello, 'The complexity of acyclic conjunctive queries', *Journal of the ACM*, **48**, (2001).

[15] I. Horrocks and U. Sattler, 'Ontology reasoning for the semantic web', in *Proceedings of IJCAI 01*.

[16] M-L. Mugnier and M.Chein, 'Polynomial algorithms for projection and matching', *Lecture Notes in Computer Science*, **754**, 239–251, (1993).

[17] A.Y. Levy and M-C. Rousset, 'Combining Horn Rules and Description Logics in Carin', *Artificial Intelligence*, **101**, (1998).

[18] B. Selman and H. Kautz, 'Knowledge compilation and theory approximation', *Journal of the ACM*, **43**(2), (1996).

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