# TBox and ABox Reasoning in Expressive Description Logics

**Giuseppe De Giacomo**
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
degiacomo@dis.uniroma1.it

**Maurizio Lenzerini**
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
lenzerini@dis.uniroma1.it

## Abstract

A Description Logic (DL) system is characterized by four fundamental aspects: the set of constructs used in concept and role expressions, the kind of assertions allowed in the TBox (assertions on concepts) and the ABox (assertions on individuals), and the inference mechanisms for reasoning on both the TBox and the ABox. Most of the research done in the last decade made several simplifying assumptions on the above aspects. However, the recent interest in DLs exhibited in many application areas (databases, software engineering, intelligent access to the network, planning, etc.) calls for investigating DL systems with full capabilities. The work presented in this paper represents a step in this direction. We present a sound, complete, and terminating (in worst-case EXPTIME) inference procedure that solves the problem of reasoning in a DL system with the following characteristics: it comes equipped with a very expressive language, it allows the most general form of TBox assertions, and it takes into account instance assertions on both concepts and roles in the ABox.

## 1 INTRODUCTION

The research on Knowledge Representation has always paid attention to languages for the representation of classes and relationships. Description Logics (DLs) have been studied in the last decade as a formalization of these languages (see (Woods & Schmolze, 1992; Donini, Lenzerini, Nardi, & Schaerf, 1996; Borgida & Patel-Schneider, 1994; Baader, Hollunder, Nebel, Profitlich, & Franconi, 1992)). They allow one to represent a domain of interest in terms of *concepts* and *roles*, where concepts model classes of individuals, and roles model relationships between classes. Starting with atomic concepts and atomic roles, which are simply described by a name, complex concepts and roles can be denoted by expressions built using suitable constructs. Concepts and roles are given Tarskian semantics in terms of sets and binary relations, respectively.

A knowledge base expressed in a DL is constituted by two components, traditionally called TBox and ABox. The TBox stores a set of universally quantified assertions (inclusion assertions) stating general properties of concepts and roles. For example, an assertion of this kind is the one stating that a concept represents a specialization of another concept. The ABox comprises assertions on individual objects (instance assertions). A typical assertion in the ABox is the one stating that an individual is an instance of a certain concept.

Several reasoning tasks can be carried out on a knowledge base of the above kind. The simplest form of reasoning involves computing the subsumption relation between two concept expressions, i.e. verifying whether one expression always denotes a subset of the objects denoted by another expression. A more complex reasoning task consists in checking whether a certain assertion (either an inclusion or an instance assertion) is logically implied by a knowledge base.

A DL system is then characterized by four aspects:

1. The set of constructs constituting the language used for building the concepts and the roles mentioned in the TBox and in the ABox.
2. The kind of assertions that may appear in the TBox.
3. The kind of assertions that may appear in the ABox.
4. The inference mechanisms provided for reasoning on the knowledge bases expressible in the system.

It follows that the expressive power and the deduction capabilities of a DL system depends on the various choices and assumptions that the system adopts with regard to the above aspects. As to the fourth aspect, we concentrate in this paper on inference mechanisms that are sound and complete with respect to the standard semantics, although other choices are possible (see (Patel-Schneider, 1989)).

Most of the basic research work on the computational complexity of DLs has been carried out in a simplified context where both the TBox and the ABox are empty (see (Donini, Lenzerini, Nardi, & Nutt, 1991a, 1991b; Nebel, 1988)). This is not surprising, since these works aimed at studying the language constructs in isolation, with the goal of singling out their impact on the complexity of subsumption between concept expressions.

Other papers dealt with logical implication of ABox assertions under the simplifying assumption of an empty TBox, again with the goal of studying how the various language constructs influence the reasoning on individuals (see (Donini, Lenzerini, Nardi, & Schaerf, 1994; Schaerf, 1994)).

More recently, there has been a strong interest in the problem of reasoning with TBox assertions in isolation (see (De Giacomo & Lenzerini, 1994a; Calvanese, De Giacomo, & Lenzerini, 1995; Nebel, 1991; Baader, 1991; Schild, 1994)). One important outcome of this line of research is that, limiting the expressive power of the language with the goal of gaining tractability is useless in this setting, because the power of TBox assertions alone (when no limitations on cycles in the TBox are imposed) generally leads to high complexity in the inference mechanisms. For this reason, these investigations often refer to very powerful languages for expressing concepts and roles.

The complete setting has been the subject of some investigations only recently. For example, in (Buchheit, Donini, & Schaerf, 1993) a DL system with both the TBox and the ABox is studied with a relatively powerful language (not including inverse roles) . However, results about reasoning on knowledge bases with both the TBox and the ABox are still rare.

We observe that such results would be very important in the light of the renewed interest in DLs that we find in disparate application areas. Indeed, DL systems are now advocated as suitable knowledge representation systems in many contexts, such as information systems (Catarci & Lenzerini, 1993), databases (Borgida, 1995; Bergamaschi & Sartori, 1992; Sheth, Gala, & Navathe, 1993), software engineering (Devambu, Brachman, Selfridge, & Ballard, 1991), intelligent access to the network (Levy, Rajaraman, & Ordille, 1996; Blanco, Illarramendi, & Goni, 1994), action representation (Artale & Franconi, 1994), and planning (Weida & Litman, 1992). Many of the above papers point out that the whole capabilities of a DL system (expressive language, TBox and ABox assertions) are often required in the corresponding application fields (see also (Doyle & Patil, 1991)).

The work presented in this paper represents a fundamental step in this direction. We present a sound, complete, and terminating inference procedure that solves the problem of reasoning in a DL system with the following characteristics:

1. It comes equipped with a very expressive language, comprising all classical concept forming constructs, plus several role forming constructs (including inverse roles), and the most general form of number restrictions.
2. It allows the most general form of TBox assertions, without any limitations on the presence of cycles.
3. It allows expressing instance assertions on both concepts and roles in the ABox.

The most important contributions of our work can be summarized as follows:

- We present the first decidability result for a DL system combining inverse roles, number restrictions, and TBox and ABox assertions simultaneously.
- We present the first technique for reasoning on ABox assertions in a DL system that does *not* enjoy the finite model property (a knowledge base in our system may have only models with infinite domains).
- Our technique is optimal with respect to the complexity class of the inference problem (EXPTIME), and has the same computational complexity (in the worst case) as the procedure for reasoning in a TBox expressed in the basic language $\mathcal{ALC}$ (Schmidt-Schauß & Smolka, 1991).

The paper is organized as follows. In Section 2, we present the DL language we are interested in, called $\mathcal{CIQ}$. In Section 3 we illustrate the various features of $\mathcal{CIQ}$ by means of some examples. In Section 4 we briefly discuss the correspondence between DLs and propositional dynamic logics (PDLs) which is at the base of our results on the reasoning procedures for

$\mathcal{CIQ}$. In Section 5 we introduce some technical notions that will be needed to get our results. In Section 6, we describe our technique for computing logical implication over knowledge bases built using $\mathcal{CIQ}$. Finally, in Section 7 we draw some conclusions.

## 2 $\mathcal{CIQ}$ SYNTAX AND SEMANTICS

In the following, we focus on the description logic $\mathcal{CIQ}$ which has been studied in (De Giacomo & Lenzerini, 1995; De Giacomo, 1995). The available constructs for concept and role expressions in $\mathcal{CIQ}$ are specified in Figure 1.

Note that $\mathcal{CIQ}$ is a very expressive language, comprising all usual concept constructs, including the most general form of number restrictions, the so called *qualified number restrictions*, and a rich set of role constructs, namely: *union* of roles $R_1 \sqcup R_2$, *chaining* of roles $R_1 \circ R_2$, *reflexive-transitive* closure of roles $R^*$, *inverse* roles $R^-$, and the identity role $id(C)$ projected on $C$ also called *test* in the following.

The semantics of $\mathcal{CIQ}$ interprets concepts as subsets of a domain, and roles as binary relations over such a domain. Formally, an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a *domain of interpretation* $\Delta^{\mathcal{I}}$, and an *interpretation function* $\cdot^{\mathcal{I}}$ mapping every atomic concept $A$ to a subset of $\Delta^{\mathcal{I}}$, and every atomic role $P$ to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is systematically extended to complex concepts and roles according to the semantics of the constructs given in Figure 1. In the figure, $\sharp S$ denotes the cardinality of the set $S$, and $(R^{\mathcal{I}})^i$ stands for $i$ repetitions of $R^{\mathcal{I}}$ – i.e., $(R^{\mathcal{I}})^0 = (id(\top))^{\mathcal{I}}$, and $(R^{\mathcal{I}})^i = R^{\mathcal{I}} \circ (R^{\mathcal{I}})^{i-1}$.

A $\mathcal{CIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is constituted by two components: a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

The TBox is a finite set of *inclusion assertions* of the form:

$$C_1 \sqsubseteq C_2$$

where $C_1$ and $C_2$ are concepts. In the following we use $C \equiv D$ as an abbreviation of $C \sqsubseteq D$ and $D \sqsubseteq C$.

The ABox is a finite set of *instance assertions* of the form:

$$C(\alpha)$$

where $C$ is a concept, and $\alpha$ is an individual name, or of the form:

$$P(\alpha_i, \alpha_j)$$

where $P$ is a primitive role and $\alpha_i, \alpha_j$ two individuals names. An interpretation $\mathcal{I}$ maps individual names to individuals in $\Delta^{\mathcal{I}}$, in such a way that different individual names denote different individuals. Therefore we do not make any distinction between individuals and their names in the following.

An interpretation $\mathcal{I}$ is a model of an inclusion assertion $C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of an instance assertion $C(\alpha)$ if $\alpha \in C^{\mathcal{I}}$, and is a model of $P(\alpha_i, \alpha_j)$ if $(\alpha_i, \alpha_j) \in P^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of knowledge base $\mathcal{K}$ if $\mathcal{I}$ is a model of each inclusion and instance assertion in $\mathcal{K}$. $\mathcal{K}$ is satisfiable if it has a model. $\mathcal{K}$ logically implies an (inclusion or instance) assertion $\sigma$, written $\mathcal{K} \models \sigma$, if $\sigma$ is satisfied by every model of $\mathcal{K}$. A concept $C$ is satisfiable in $\mathcal{K}$ if there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $C^{\mathcal{I}} \neq \emptyset$. Observe that satisfiability of a concept $C$ in a knowledge base $\mathcal{K}$ can be reformulated in terms of logical implication as $\mathcal{K} \not\models C \sqsubseteq \bot$, and in terms of satisfiability of a knowledge base as the satisfiability of $\mathcal{K} \cup \{C(\alpha_{new})\}$, where $\alpha_{new}$ in an individual not mentioned in $\mathcal{K}$.

## 3 EXAMPLES

Figure 2 shows a $\mathcal{CIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, concerning directories and files. The TBox $\mathcal{T}$ is made by four assertions.

The first inclusion assertion states that every `dir_child` of an instance $d$ of `Directory` is either a directory or a file, and has exactly one `dir_child`$^-$ predecessor, which is $d$ itself. In other words, the fragment of `dir_child` starting from an instance of `Directory` has a structure similar to a tree, except that cycles are not prevented.

The second inclusion assertion states that instances of `File` have no children, and that are distinct from instances of `Directory`.

The third assertion states that the instances of `FileSysRoot` are directories which have no `dir_child`-predecessor.

The forth assertion states that every instance of `FileSysElement` reaches an instance of `FileSysRoot` in a finite number of steps through a chain of `dir_child`$^-$. It can be seen that such constraint, together with the first assertion, prevents cycles to appear in `dir_child`-chains involving instances of `FileSysElement`.

The ABox $\mathcal{A}$ can be thought of as divided into parts.

The first part is made of instance assertions concerning the individuals `a`, `b` and `c`, and their `dir_child` relationships. It expresses that `a` has two children, namely `b` and `c`, and that `a` is in turn a child of `c` (i.e. there is a cycle involving `a` and `b`).

The second part concerns the individuals `MyDir`,

| Construct Name | Syntax | Semantics |
|---|---|---|
| atomic concept | $A$ | $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ |
| top | $\top$ | $\Delta^{\mathcal{I}}$ |
| bottom | $\bot$ | $\emptyset$ |
| conjunction | $C_1 \sqcap C_2$ | $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$ |
| disjunction | $C_1 \sqcup C_2$ | $C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$ |
| negation | $\neg C$ | $\Delta^{\mathcal{I}} - C^{\mathcal{I}}$ |
| existential quantification | $\exists R.C$ | $\{s \mid \exists s'.(s,s') \in R^{\mathcal{I}} \text{ and } s' \in C^{\mathcal{I}}\}$ |
| universal quantification | $\forall R.C$ | $\{s \mid \forall s'.(s,s') \in R^{\mathcal{I}} \text{ implies } s' \in C^{\mathcal{I}}\}$ |
| qualified number restrictions | $(\geq n\, Q.C)$ | $\{s \mid \sharp\{s'.(s,s') \in Q^{\mathcal{I}} \text{ and } s' \in C^{\mathcal{I}}\} \geq n\}$ |
| | $(\leq n\, Q.C)$ | $\{s \mid \sharp\{s'.(s,s') \in Q^{\mathcal{I}} \text{ and } s' \in E^{\mathcal{I}}\} \leq n\}$ |
| atomic role | $P$ | $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ |
| union | $R_1 \sqcup R_2$ | $R_1^{\mathcal{I}} \cup R_2^{\mathcal{I}}$ |
| chaining | $R_1 \circ R_2$ | $R_1^{\mathcal{I}} \circ R_2^{\mathcal{I}}$ |
| reflexive-transitive closure | $R^*$ | $\bigcup_{i>0}(R^{\mathcal{I}})^i$ |
| test | $id(C)$ | $\{(s,s) \mid s \in C^{\mathcal{I}}\}$ |
| inverse | $R^-$ | $\{(s,s') \mid (s',s) \in R^{\mathcal{I}}\}$ |
| basic role | $Q = P \mid P^-$ | |

Figure 1: Syntax and semantics of $\mathcal{CIQ}$ concept and role constructs.

| TBox: | ABox: |
|---|---|
| $\texttt{Directory} \sqsubseteq \forall \texttt{dir\_child}.((\texttt{Directory} \sqcup \texttt{File}) \sqcap (\leq 1\, \texttt{dir\_child}^-.\top))$<br>$\texttt{File} \sqsubseteq (\forall \texttt{dir\_child}.\bot) \sqcap \neg \texttt{Directory}$<br><br>$\texttt{FileSysRoot} \sqsubseteq \texttt{Directory} \sqcap \forall \texttt{dir\_child}^-.\bot$<br>$\texttt{FileSysElement} \equiv \exists(\texttt{dir\_child}^-)^*.\texttt{FileSysRoot}$ | dir_child(a, b)<br>dir_child(a, c)<br>dir_child(c, a)<br><br>dir_child(MyDir, Research)<br>dir_child(MyDir, Teaching)<br>dir_child(Research, CIQ.tex)<br>FileSysElement(CIQ.tex)<br>File(CIQ.tex) |

Figure 2: Example: directories and files.

**Teaching**, **Research** and **CIQ.tex**. It expresses that **MyDir** has **Teaching** and **Research** as children, and that **CIQ.tex** is a child of **Research**. Moreover **CIQ.tex** is both a **FileSysElement** and a **File**.

From $\mathcal{K}$ we can make the following inference:

$$\mathcal{K} \models \texttt{FileSysElement} \sqsubseteq \texttt{Directory} \sqcup \texttt{File}.$$

Let us prove the above logical implication. By the fourth assertion in the TBox, every instance $s$ of **FileSysElement** reaches an instance $s'$ of **FileSysRoot** in a finite (but indeterminate) number $n$ of $\texttt{dir\_child}^-$ steps. We proceed by *induction* on $n$. If $n = 0$, then $s = s'$. Hence $s$ is an instance of **FileSysRoot** and so is an instance of **Directory**. If $n = k+1 > 0$, then let $s''$ be the immediate **dir_child**-predecessor along the chain. By induction hypothesis, $s''$ either a directory or a file. Since it has a **dir_child**-successor, namely $s$, it must be a directory and hence all is children, including $s$ are either a directory or a file.

With minimal modification, this proof applies to the following logical implication as well:

$$\mathcal{K} \models \texttt{FileSysElement} \sqsubseteq \forall(\texttt{dir\_child}^-)^*.\texttt{Directory}.$$

The knowledge base $K$ logically implies also that a, b, and c are not **FileSysElement**, i.e. for $\alpha = \texttt{a}, \texttt{b}, \texttt{c}$:

$$\mathcal{K} \models \neg\texttt{FileSysElement}(\alpha).$$

To prove the above logical implication we may reason as follows. First observe that none of a, b, and c

can be an instance of **FileSysRoot**, since all of them have an immediate **dir_child**-predecessor. Now let, for example, $\alpha$ = **b**. Suppose that **b** is an instance of **FileSysElement**. By the fourth assertion in the TBox, **b** must be connected by a finite chain of $\mathtt{dir\_child}^-$ to an instance of **FileSysRoot**. Also, as we saw before, the fact that **b** is an instance of **FileSysElement** implies that all **dir_child**-predecessors of **b** are instances of **Directory**. It follows from the first assertion in the TBox that each of the individuals **a**, **b**, and **c** has at most one **dir_child** immediate predecessor, and therefore, there are no **dir_child**-predecessors of **b** other than **a** and **c**. Since neither **a** nor **c** can be an instance of **FileSysRoot**, we have a contradiction. Hence we can conclude that neither **a**, **b**, or **c** are instances of **FileSysElement**.

With similar reasoning we can prove that, for $\beta$ = **MyDir**, **Teaching**, **Research**:

$$\mathcal{K} \models \mathtt{FileSysElement}(\beta)$$

and also that **MyDir** and **Research** are instances of **Directory**, while **Teaching** is either an instance of **Directory** or an instance of **File**. In addition, we can prove that it is consistent that **MyDir** is an instance of **FileSysRoot**, though it is not logically implied.

Observe that in the proofs above, the use of induction is essential. Thus, the automatic reasoning procedure for $\mathcal{CIQ}$ must include either implicitly or explicitly such form of induction. The need for induction comes, as shown in the examples, from the presence of the reflexive-transitive closure of roles, which allows the specification of properties of objects that are distant a finite but indeterminate number of steps away (through a chain of roles). This ability testifies the non-first-order nature of our logic. $\mathcal{CIQ}$ is indeed a subset of *first order logic + fixpoints* (see (De Giacomo & Lenzerini, 1994b) and not of the pure *first order logic*, as most description logics are.

Finally, note that the knowledge base described in Figure 2 enjoys the finite model property. However, it easy to build a $\mathcal{CIQ}$ knowledge base with only infinite models. For example consider the following knowledge base:

| | |
|---|---|
| **TBox:** | $\top \sqsubseteq (\leq 1\,\mathtt{succ}.\top) \sqcap (\leq 1\,\mathtt{succ}^-.\top)$<br>$\mathtt{InfSeq} \sqsubseteq \exists \mathtt{succ}.\mathtt{InfSeq}$ |
| **ABox:** | $(\mathtt{InfSeq} \sqcap \forall \mathtt{succ}^-.\bot)(\mathtt{Init})$ |

The first assertion constrains the role **succ** and its inverse to be functional.

The second assertion constrains the instances of **InfSeq** to have *its* immediate successor in **InfSeq** as well, i.e. each instance of **InfSeq** either is a (not necessarily immediate) successor of itself, or has an infinite chain of successors.

The instance assertion states that the individual **Init** is an instance of **InfSeq** but does not have any predecessor.

Now, since **Init** has no predecessor, it cannot be a successor of itself, so being an **InfSeq** it must have an infinite chain of successor. Hence all models of the knowledge base are infinite.

Observe that the existence of knowledge bases that admit only infinite models makes the reasoning methods based on the direct search and construction of a model (as the tableaux-based method in (Donini et al., 1994)) infeasible for $\mathcal{CIQ}$. Any reasoning procedure for $\mathcal{CIQ}$ based on the construction of a model, may at most construct a finite structure that represents a model (a pseudo-model), in the sense that it contains enough information so that, in principle, it can be expanded (maybe not univocally) to a model.

# 4 CORRESPONDENCE WITH PDLs

In the next sections, we will describe the procedure for reasoning in $\mathcal{CIQ}$-knowledge bases. Such procedure is based on the inference technique that the authors developed for the description logic $\mathcal{CIQ}$, mainly based on the correspondence between DLs and Propositional Dynamic Logics (PDLs) (Schild, 1991; De Giacomo & Lenzerini, 1994a; De Giacomo, 1995). PDLs are modal logics developed to specify and reason about program schemas in terms of states and state transitions caused by (running) a program (Fischer & Ladner, 1979; Kozen & Tiuryn, 1990).

The correspondence between DLs and PDLs is due to a substantial similarity between the interpretation structure of the two kinds of logics: individuals in DLs correspond to states in PDLs, links between individuals correspond to state transitions, concepts correspond to formulae, and roles correspond to programs. In fact, most constructs in DLs have a counterpart in known PDLs as shown in Figure 3. In (Schild, 1991), using the correspondence, many new results on DLs were obtained from known results on PDLs. In particular, from the decision procedures for Converse PDL, the first reasoning procedures for DLs that include inverse and TBoxes were devised.

Notably, neither qualified number restrictions nor ABoxes have a counterpart in PDLs.

Indeed the only form of number restrictions known in PDLs is that of assuming all atomic programs (not

| DLs | | PDLs | |
|---|---|---|---|
| atomic concept | $A$ | atomic proposition | $A$ |
| top | $\top$ | true | tt |
| bottom | $\bot$ | false | ff |
| conjunction | $C_1 \sqcap C_2$ | conjunction | $\phi_1 \wedge \phi_2$ |
| disjunction | $C_1 \sqcup C_2$ | disjunction | $\phi_1 \vee \phi_2$ |
| negation | $\neg C$ | negation | $\neg\phi$ |
| existential quantification | $\exists R.C$ | diamond ("some runs ...") | $<r>\phi$ |
| universal quantification | $\forall R.C$ | box ("all runs ...") | $[r]\phi$ |
| qualified number restrictions | $(\geq n\, Q.C)$ $(\leq n\, Q.C)$ | (deterministic PDLs) | **assumption:** deterministic atomic programs |
| atomic role | $P$ | atomic program | $P$ |
| union | $R_1 \sqcup R_2$ | choice | $r_1 \cup r_2$ |
| chaining | $R_1 \circ R_2$ | sequence | $r_1; r_2$ |
| reflexive-transitive closure | $R^*$ | reflexive-transitive closure | $r^*$ |
| test | $id(C)$ | test | $\phi?$ |
| inverse | $R^-$ | converse | $r^-$ |
| basic role | $Q$ | — | — |
| inclusion assertions | $C_1 \sqsubseteq C_2$ | axioms (valid formulae) | $\phi$ |
| instance assertions | $C(\alpha) \mid P(\alpha_1, \alpha_2)$ | — | — |

Figure 3: Correspondence between DLs and PDLs.

their inverse) to be deterministic, thus getting the so call Deterministic PDLs. As an aside, Deterministic PDLs that include also the converse operator do not have the finite model property and indeed the reasoning procedures developed for these logics are based on the construction of automata on infinite trees.

As for ABoxes, in PDLs, they would roughly correspond to a partial specification of an actual evaluation of a program. However, such kind of specification have not been studied yet.

The research in (De Giacomo & Lenzerini, 1994a, 1995; De Giacomo, 1995) has tackled these two aspects.

In (De Giacomo & Lenzerini, 1994a) the EXPTIME-decidability of $\mathcal{CIF}$, i.e. $\mathcal{CIQ}$ with number restrictions limited to unqualified functional restrictions (on both atomic roles and their inverse), was established. In (De Giacomo & Lenzerini, 1995; De Giacomo, 1995) this result was extended to $\mathcal{CIQ}$. The reasoning procedures developed in these works do not construct automata on infinite trees, but are based on a polynomial encoding of a $\mathcal{CIQ}$ TBox into a $\mathcal{CIF}$ TBox, which is in turn encoded into a $\mathcal{CI}$-concept (corresponding to a Converse PDL formula. Observe that, from $\mathcal{CIQ}$ to $\mathcal{CI}$, we go from a logic which does not have the finite model property to a logic that does have it.

As for ABoxes, the best known results about reasoning on knowledge bases constituted by both a TBox and an ABox, are two EXPTIME reasoning procedures presented in (De Giacomo & Lenzerini, 1994a; De Giacomo, 1995) for $\mathcal{CI}$ and $\mathcal{CQ}$ (the logic obtainded from $\mathcal{CIQ}$ by disallowing inverse roles), respectively. Notice that both logics have the finite model property.

Finally we remark that in $\mathcal{CIQ}$, qualified number restrictions are allowed only for basic roles (i.e. atomic roles and their inverse). This is a disign choice due to the fact that allowing a generic role to appear in a qualified number restriction would have made the logic undecidable. Indeed it suffices to observe that the unqualified functional restriction $(\leq 1\,(R_1 \sqcup R_2).\top)$ is in fact a form of role value map[1], which leads to undecidability (e.g. see (Schmidt-Schauß, 1989)).

## 5 TECHNICAL PRELIMINARIES

We assume, without loss of generality, $\sqcup, \forall, \leq$ to be expressed by means of $\neg, \sqcap, \exists, \geq$, and the inverse role operator to be applied to atomic roles only[2].

The *Fisher-Ladner closure* (Fischer & Ladner, 1979) of a $\mathcal{CIQ}$ concept $C$ is denoted by $CL(C)$ and is defined

[1] This observation is originally due to Franz Baader.

[2] We recall that the following equations hold: $(R_1 \circ R_2)^- = R_2^- \circ R_1^-$, $(R_1 \sqcup R_2)^- = R_1^- \sqcup R_2^-$, $(R_1^*)^- = (R_1^-)^*$, $id(C)^- = id(C)$.

inductively as the smallest set of concepts $S$ containing $C$ and such that:

$$
\begin{array}{ll}
C_1 \sqcap C_2 \in S & \text{implies } C_1, C_2 \in S \\
\neg C' \in S & \text{implies } C' \in S \\
C' \in S & \text{implies } \neg C' \in S \text{ (if } C' \neq \neg C'') \\
(\geq n\, Q.C') \in S & \text{implies } C' \in S \\
\exists R.C' \in S & \text{implies } C' \in S \\
\exists R_1 \circ R_2.C' \in S & \text{implies } \exists R_1.\exists R_2.C' \in S \\
\exists R_1 \sqcup R_2.C' \in S & \text{implies } \exists R_1.C', \exists R_2.C' \in S \\
\exists R^*.C' \in S & \text{implies } \exists R.\exists R^*.C' \in S \\
\exists id(C'').C' \in S & \text{implies } C'' \in S.
\end{array}
$$

Intuitively, $CL(C)$ is analogous to the set of subconcepts in simpler logics: It comprises the concepts that play a direct role in establishing the interpretation of $C$. The size of $CL(C)$ is linearly bounded by the size of $C$ (cf. (Fischer & Ladner, 1979)). By definition, if $C' \in CL(C)$, then $CL(C') \subseteq CL(C)$.

We can extend the above notion to that of the Fisher-Ladner closure of a knowledge base by simply taking the union of the Fisher-Ladner closures of all concepts appearing in the knowledge base.

Let us denote the *empty sequence of roles* by the role $\varepsilon$, and define $\exists\varepsilon.C \doteq C$ and $\forall\varepsilon.C \doteq C$. Given a role $R$, we call $Pre(R)$ and $Post(R)$ the two sets of roles defined inductively as follows ($Q = P \mid P^-$):

$$
\begin{array}{ll}
Pre(Q) & = \{\varepsilon, Q\} \\
Pre(R_1 \circ R_2) & = \{R_1 \circ R_2' \mid R_2' \in Pre(R_2)\} \cup Pre(R_1) \\
Pre(R_1 \sqcup R_2) & = Pre(R_1) \cup Pre(R_2) \\
Pre(R_1^*) & = \{R_1^* \circ R_1' \mid R_1' \in Pre(R_1)\} \\
Pre(id(C)) & = \{\varepsilon, id(C)\}
\end{array}
$$

$$
\begin{array}{ll}
Post(Q) & = \{\varepsilon, Q\} \\
Post(R_1 \circ R_2) & = \{R_1' \circ R_2 \mid R_1' \in Post(R_1)\} \cup Post(R_2) \\
Post(R_1 \sqcup R_2) & = Post(R_1) \cup Post(R_2) \\
Post(R_1^*) & = \{R_1' \circ R_1^* \mid R_1' \in Post(R_1)\} \\
Post(id(C)) & = \{\varepsilon, id(C)\}.
\end{array}
$$

Roughly speaking, $Pre(R)$ and $Post(R)$ are the sets formed by those roles that are "prefix" and "postfix" of the role $R$, respectively. The size of both $Pre(R)$ and $Post(R)$ is polynomial in the size of $R$.

For the roles in $Post(R)$ the following two properties can be easily proven (see (De Giacomo, 1996, 1995)):

- Let $\exists R.C$ be a concept. For all roles $R' \in Post(R)$, $\exists R'.C \in CL(\exists R.C)$.
- Let $\exists R_1 \ldots \exists R_l.C$ be a concept. For all roles $R' \in Post(R_1 \circ \ldots \circ R_l)$, there is a formula $D \in CL(\exists R_1 \ldots \exists R_l.C)$ such that $D$ is equivalent to $\exists R'.C$.

A *path* in an interpretation $\mathcal{I}$ is a sequence $(s_0, \ldots, s_q)$ of elements of $\Delta^{\mathcal{I}}$ ($q \geq 0$), such that for each $i = 1, \ldots, q$, $(s_{i-1}, s_i) \in Q^{\mathcal{I}}$, for some $Q = P \mid P^-$. The length of $(s_0, \ldots, s_q)$ is $q$. Intuitively a path describes the sequence of individuals which are met by following a role (or the inverse of a role) in a given interpretation. We inductively define the set of paths $Paths_{\mathcal{I}}(R)$ of a role $R$ in an interpretation $\mathcal{I}$, as follows, where ($Q = P \mid P^-$):

$$
\begin{array}{ll}
Paths_{\mathcal{I}}(Q) & = Q^{\mathcal{I}} \\
Paths_{\mathcal{I}}(R_1 \sqcup R_2) & = Paths_{\mathcal{I}}(R_1) \cup Paths_{\mathcal{I}}(R_2) \\
Paths_{\mathcal{I}}(R_1 \circ R_2) & = \{(s_0, \ldots, s_u, \ldots, s_q) \mid \\
 & \qquad (s_0, \ldots, s_u) \in Paths_{\mathcal{I}}(R_1) \text{ and} \\
 & \qquad (s_u, \ldots, s_q) \in Paths_{\mathcal{I}}(R_2)\} \\
Paths_{\mathcal{I}}(R^*) & = \{(s) \mid s \in \Delta^{\mathcal{I}}\} \cup (\bigcup_{i>0} Paths_{\mathcal{I}}(R^i)) \\
Paths_{\mathcal{I}}(id(C)) & = \{(s) \mid s \in C^{\mathcal{I}}\}.
\end{array}
$$

We say that a path $(s_0)$ in $\mathcal{I}$ *satisfies* a concept $C$ which is not of the form $\exists R.C$ if $s_0 \in C^{\mathcal{I}}$. We say that a path $(s_0, \ldots, s_q)$ in $\mathcal{I}$ *satisfies* a concept $C$ of the form $\exists R_1. \cdots \exists R_l.C'$, where $C'$ is not of the form $\exists R'.C''$, if $(s_0, \ldots s_q) \in Paths_{\mathcal{I}}(R_1 \circ \cdots \circ R_l)$ and $s_q \in C'^{\mathcal{I}}$.

The following two propositions describe the basic properties of paths and can be proven by induction on the structure of the role $R$ (see (De Giacomo, 1996, 1995)).

**Proposition 1** *Let $\mathcal{I}$ be an interpretation and $\exists R.C$ a concept such that: $s \in (\exists R.C)^{\mathcal{I}}$, $(s) \in Paths_{\mathcal{I}}(R)$, and $s \in C^{\mathcal{I}}$. Then there exists a concept $\exists id(C_1) \circ \ldots \circ id(C_g).C$, with $g \geq 0$, such that:*

- *all tests $id(C_i)$ occur in $R$, and hence $C_i \in CL(\exists R.C)$;*
- *$s \in (\exists id(C_1) \circ \ldots \circ id(C_g).C)^{\mathcal{I}}$;*
- *$\exists id(C_1) \circ \ldots \circ id(C_g).C \sqsubseteq \exists R.C$ is valid.*

**Proposition 2** *Let $\mathcal{I}$ be a structure, and $\exists R.C$ a formula such that: $s \in (\exists R.C)^{\mathcal{I}}$, $(s = s_0, \ldots, s_q) \in Paths_{\mathcal{I}}(R)$ with $q > 0$, $s_q \in C^{\mathcal{I}}$. Then there exists a formula $\exists id(C_1) \circ \ldots \circ id(C_g) \circ Q.\exists R'.C$, with $g \geq 0$, such that:*

- *all tests $id(C_i)$ occur in $R$, and hence $C_i \in CL(\exists R.C)$;*
- *$R' \in Post(R)$ and hence $\exists R'.C$ is equivalent to $D$ for some $D \in CL(\exists R.C)$;*

- $(s_0, s_1) \in Q^{\mathcal{I}}$;
- $s_1 \in (\exists R'.C)^{\mathcal{I}}$;
- $(s_1, \ldots, s_q) \in Paths_{\mathcal{I}}(R')$;
- $\exists id(C_1) \circ \ldots \circ id(C_g) \circ Q.(\exists R'.C) \sqsubseteq \exists R.C$ *is valid.*

# 6 REASONING IN $\mathcal{CIQ}$ KNOWLEDGE BASES

In this section, we illustrate the technique for reasoning on $\mathcal{CIQ}$ knowledge bases. The basic idea underlying our method is as follows: checking the satisfiability of a $\mathcal{CIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is polynomially reduced to checking the satisfiability of a $\mathcal{CIQ}$ knowledge base $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$, whose ABox $\mathcal{A}'$ is made of a single instance assertion $C(\alpha)$. In other words, the satisfiability of $\mathcal{K}$ is reduced to the satisfiability of the concept $C$ wrt the TBox $\mathcal{T}'$ of the resulting knowledge base. The latter reasoning service can be realized by means of the method presented in (De Giacomo & Lenzerini, 1995; De Giacomo, 1995), and is known to be EXPTIME-complete. Thus, by means of the reduction, we get an EXPTIME algorithm for satisfiability of $\mathcal{CIQ}$ knowledge base, and hence for all reasoning services on $\mathcal{CIQ}$ knowledge bases.

**Definition** Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{CIQ}$ knowledge base. We call *the reduced form* of $\mathcal{K}$ the $\mathcal{CIQ}$ knowledge base $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ defined as follows (a new atomic concept $A_i$ is introduced for each individual $\alpha_i$ (i=1,...,m) occurring in $\mathcal{A}$).

- $\mathcal{A}' = \{(\exists create.A_1 \sqcap \ldots \sqcap \exists create.A_m)(g)\}$, where $g$ is a new individual (the only one present in $\mathcal{A}'$) and *create* is a new atomic role:
- $\mathcal{T}'$ is formed by $\mathcal{T}'_{\mathcal{K}}$ and $\mathcal{T}'_{aux}$:
  - $\mathcal{T}'_{\mathcal{K}} = \mathcal{T}'_{\mathcal{T}} \cup \mathcal{T}'_{\mathcal{A}}$, where $\mathcal{T}'_{\mathcal{T}} = \mathcal{T}$, and $\mathcal{T}'_{\mathcal{A}}$ is made of one inclusion assertion:

    $$A_i \sqsubseteq C$$

    for each instance assertion $C(\alpha_i) \in \mathcal{A}$, two inclusion assertions:

    $$A_i \sqsubseteq \exists P.A_j \sqcap (\leq 1\, P.A_j)$$
    $$A_j \sqsubseteq \exists P^-.A_i \sqcap (\leq 1\, P^-.A_i)$$

    for each instance assertion $P(\alpha_i, \alpha_j) \in \mathcal{A}$, and one inclusion assertion:

    $$A_i \sqsubseteq \sqcap_{i \neq j} \neg A_j$$

    for each individual $\alpha_i$ occurring in $\mathcal{A}$.
  - $\mathcal{T}'_{aux}$ is made of one inclusion assertion ($\mathbf{u}$ stands for $(P_1 \sqcup \ldots \sqcup P_n \sqcup P_1^- \sqcup \ldots \sqcup P_n^-)^*$, where $P_1, \ldots, P_n$ are all the atomic roles in $\mathcal{T}'_{\mathcal{K}}$):

    $$(A_i \sqcap C) \sqsubseteq \forall \mathbf{u}.(\neg A_i \sqcup C)$$

    for each $A_i$ occurring in $\mathcal{T}'_{\mathcal{K}}$ and $C$ such that:
    1. $C \in CL(\mathcal{T}'_{\mathcal{K}})$
    2. $C = \exists \overline{R}.C'$ with $\exists R.C' \in CL(\mathcal{T}'_{\mathcal{K}})$
    3. $C = \exists(\overline{R'} \circ Q).A_j$ with $R' \in Pre(R)$, $Q = P \mid P^-$, and $R, P, A_j$ occurring in $CL(\mathcal{T}'_{\mathcal{K}})$

    where, $\overline{R}$ is defined inductively as follows ($Q = P \mid P^-$):
    - $\overline{Q} = Q \circ id(\sqcap_i \neg A_i)$;
    - $\overline{R_1 \circ R_2} = \overline{R_1} \circ \overline{R_2}$;
    - $\overline{R_1 \sqcup R_2} = \overline{R_1} \cup \overline{R_2}$;
    - $\overline{R_1^*} = \overline{R_1}^*$;
    - $\overline{id(C)} = id(C)$.

□

**Lemma 3** *Let $\mathcal{K}$ be a $\mathcal{CIQ}$ knowledge base, and $\mathcal{K}'$ its reduced form. Then the size of $\mathcal{K}'$ is polynomial with respect to the size of $\mathcal{K}$.*

Let us comment on how the reduced form $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ relates to the original knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. First, observe that the ABox $\mathcal{A}'$ is used to force the existence of the only individual $g$, connected by the role *create* to one instance of each $A_i$. It can be shown that this allows us to restrict the attention to models of $\mathcal{K}'$ that represent graphs connected to $g$, i.e. models $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{K}'$ such that $\Delta^{\mathcal{I}} = \{g\} \cup \{s' \mid (g, s') \in create^{\mathcal{I}} \circ (\bigcup_P (P^{\mathcal{I}} \sqcup P^{\mathcal{I}^-})^*$.

The TBox $\mathcal{T}'$ consists of two parts $\mathcal{T}'_{\mathcal{K}}$ and $\mathcal{T}'_{aux}$. $\mathcal{T}'_{\mathcal{K}}$ is made of the original inclusion assertions in $\mathcal{T}$ plus what we may call a "naive encoding" of the original ABox $\mathcal{A}$ as inclusion assertions, which form $\mathcal{T}'_{\mathcal{A}}$. Indeed, each individual $\alpha_i$ is represented in $\mathcal{T}'_{\mathcal{A}}$ as a new atomic concept $A_i$ (disjoint from the other $A_j$'s), and the instance assertions in the original ABox $\mathcal{A}$ are represented as inclusion assertions in $\mathcal{T}'_{\mathcal{A}}$ involving such new atomic concepts. However $\mathcal{T}'_{\mathcal{K}}$ alone does not suffice to represent faithfully (wrt the reasoning services we are interested in) the original knowledge base, because an individual $\alpha_i$ in $\mathcal{K}$ is represented by the set of instances of $A_i$ in $\mathcal{K}'$. In order to relate the satisfiability of $\mathcal{K}'$ to the satisfiability of $\mathcal{K}$, we must be able to single out, for each $A_i$, one instance of $A_i$ representative of $\alpha_i$. For this purpose, we need to add a new part, called $\mathcal{T}'_{aux}$, to $\mathcal{T}'$. Roughly speaking, $\mathcal{T}'_{aux}$ contains inclusion assertions of the form:

$$(A_i \sqcap C) \sqsubseteq \forall \mathbf{u}.(\neg A_i \sqcup C)$$

which say that if an instance of $A_i$ is also an instance of $C$,a new then every instance of $A_i$ is an instance of $C$. Observe that if we could add an infinite set of assertions of this form, one for each possible concept of the language (e.g. by a kind of axiom schema), we could safely restrict our attention to models of $\mathcal{K}'$ with just one instance for every concept $A_i$ $(i = 1, \ldots, m)$, since there would be no way in the logic to distinguish two instances of $A_i$ one from the other. What we show below is that in fact we do need only a finite (polynomial) number of such inclusion assertions (as specified by $\mathcal{T}'_{aux}$) in order to be able to identify, for each $i$, an instance of $A_i$ as representative of $\alpha_i$. This allows us to prove that the existence of a model of $\mathcal{K}'$ implies the existence of a model of $\mathcal{K}$.

The individuals $t$ of a model $\mathcal{I}$ of $\mathcal{K}'$ such that $t \in A_i^{\mathcal{I}}$ are called *aliases* of the individual $\alpha_i$ in $\mathcal{I}$. The assertions in $\mathcal{T}'_{aux}$ allow us to prove the lemma below.

**Lemma 4** *Let $\mathcal{K}$ be a $\mathcal{CIQ}$ knowledge base, $\mathcal{K}'$ its reduced form, and $\mathcal{I}$ a model of $\mathcal{K}'$. Let $t$ be an alias of $\alpha_i$ in $\mathcal{I}$, and let $\exists R.C \in CL(\mathcal{T}'_{\mathcal{K}})$. If there is a path from $t$ that satisfies $\exists R.C$ and contains $N$ aliases $t = t_1, \ldots, t_N$, of $\alpha_i = \alpha_{i_1}, \ldots, \alpha_{i_N}$ respectively, then from every alias $t'$ of $\alpha_i$ in $\mathcal{I}$, there is a path that satisfies $\exists R.C$ and contains $N$ aliases $t' = t'_1, \ldots, t'_N$ of $\alpha_{i_1}, \ldots, \alpha_{i_N}$, in the same order as $t_1, \ldots, t_N$.*

**Proof** By induction on the number $N$ of aliases, making use of the inclusion assertions in $\mathcal{T}'_{aux}$ with $C$ of the form (2) and (3). □

We further restrict our attention to tree-like models only, without loss of generality. Indeed, any model $\mathcal{I}$ of $\mathcal{K}'$ can be easily transformed into a tree-like model, by simply unfolding $\mathcal{I}$ as follows: Put $g$ as the root of the tree; for each $Q$-successor $(Q = P \mid P^-)$ of $g$ add it to the tree as a $Q$-child of the node; continue recursively to process the children of $g$, and so on. Observe that the tree-like model obtained may be infinite.

Note that, by virtue of $\mathcal{A}'$, in the tree-like model, for each $\alpha_i$ occurring in $\mathcal{K}$, $g$ has one *create*-successor $s_{\alpha_i}$, as a child, such that $s_{\alpha_i} \in A_i$. Moreover, each $s_{\alpha_i}$ has a single $P$-successor $s \in A_j^{\mathcal{I}}$ for each $P(\alpha_i, \alpha_j) \in \mathcal{K}$ and a single $P$-predecessor $s' \in A_j^{\mathcal{I}}$ for each $P(\alpha_j, \alpha_i) \in \mathcal{K}$, by $\mathcal{T}'_{\mathcal{A}}$.

Given a tree-like model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{K}'$, we define a new interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ of $\mathcal{K}'$ as follows

- $\Delta^{\mathcal{I}'} = \{g\} \cup \{s \in \Delta^{\mathcal{I}} \mid (g, s) \in \mathcal{R}_{create} \circ (\bigcup_P (\mathcal{R}_P \cup \mathcal{R}_P^-))^*\}$
- $create^{\mathcal{I}} = \mathcal{R}_{create}$ and $P^{\mathcal{I}} = \mathcal{R}_P \cap (\Delta^{\mathcal{I}'} \times \Delta^{\mathcal{I}'})$ for each atomic role $P$ occurring in $\mathcal{K}'$
- $A^{\mathcal{I}'} = A^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$ for each atomic concept $A$ occurring in $\mathcal{K}'$

where

- $\mathcal{R}_{create} = \{(g, s_{\alpha_i}) \in create^{\mathcal{I}} \mid \alpha_i \text{ for } i = 1, \ldots, m\}$
- $\mathcal{R}_P = (P^{\mathcal{I}} - (\{(s_{\alpha_i}, s) \in P^{\mathcal{I}} \mid s \in A_j^{\mathcal{I}} \text{ and } P(\alpha_i, \alpha_j) \in \mathcal{K}\} \cup \{(s_{\alpha_j}, s') \in (P^-)^{\mathcal{I}} \mid s' \in A_i^{\mathcal{I}} \text{ and } P(\alpha_i, \alpha_j) \in \mathcal{K}\})) \cup \{(s_{\alpha_i}, s_{\alpha_j}) \mid P(\alpha_i, \alpha_j) \in \mathcal{K}\}$.

Observe that in $\mathcal{I}'$, for every atomic role $P$, the number of $P$-successors of all individuals in $\Delta^{\mathcal{I}'}$, is the same as in $\mathcal{I}$. The following lemma holds for $\mathcal{I}'$.

**Lemma 5** *Let $\mathcal{K}$ be a $\mathcal{CIQ}$ knowledge base and $\mathcal{K}'$ its reduced form. Let $\mathcal{I}$ be a model of $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$, and $\mathcal{I}'$ be the interpretation obtained from $\mathcal{I}$ as above. Then, for every $C \in CL(\mathcal{T}'_{\mathcal{K}})$ and for every $x \in \Delta^{\mathcal{I}'}$:*

$$x \in C^{\mathcal{I}'} \text{ if and only if } x \in C^{\mathcal{I}}.$$

**Proof** By induction on the formation of $C$ (called concept induction in the following). The only complex case is $C = \exists R.C'$. Here we show the if-direction of such a case (the only-if-direction is similar, yet slightly simpler).

If $x \in (\exists R.C')^{\mathcal{I}}$, then there is a path $(x = x_0, \ldots, x_q) \in Paths_{\mathcal{I}}(R)$ such that $x_q \in C'^{\mathcal{I}}$. We prove $x \in (\exists R.C')^{\mathcal{I}'}$, by induction on the number $k$ of aliases along the path $(x_0, \ldots, x_q)$, different from $s_{\alpha_i}$ for any $i$ (we call this induction, path induction).

Case $k = 0$. In this case, for all the states $x_i$ along the path, $x_i \in \Delta^{\mathcal{I}'}$. By applying Proposition 2 $q$ times and Proposition 1 once, we can conclude that there exists a concept $\exists((id(C_{0,1}) \circ \ldots \circ id(C_{0,g_0})) \circ Q_1 \circ \ldots \circ (id(C_{q-1,1}) \circ \ldots \circ id(C_{q-1,g_{q-1}})) \circ Q_q \circ (id(C_{q,1}) \circ \ldots \circ id(C_{q,g_q}))).C'$ with $g_i \geq 0$, such that:

- all tests $id(C_{i,j})$ occur in $R$, and hence $C_{i,j} \in CL(\exists R.C') \subseteq CL(\mathcal{T}'_{\mathcal{K}})$;
- $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}}$, for $i = 1, \ldots, q$;
- $\exists((id(C_{0,1}) \circ \ldots \circ id(C_{0,g_0})) \circ Q_1 \circ \ldots \circ (id(C_{q-1,1}) \circ \ldots \circ id(C_{q-1,g_{q-1}})) \circ Q_q \circ (id(C_{q,1}) \circ \ldots \circ id(C_{q,g_q}))).C' \sqsubseteq \exists R.C'$ is valid.

By concept induction hypothesis we have that, for all $C_{i,j}$, $x_i \in C_{i,j}^{\mathcal{I}}$ iff $x_i \in C_{i,j}^{\mathcal{I}'}$, and $x_q \in C'^{\mathcal{I}}$ iff $x_q \in C'^{\mathcal{I}'}$. By construction of $\mathcal{I}'$, $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}}$ implies $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}'}$. Hence $x \in (\exists R.C')^{\mathcal{I}'}$.

Case $k > 0$. Let $(x_0, \ldots, x_q) = (x_0, \ldots, x_u, \ldots x_q)$ where $x_u$, such that $x_u \in A_j^{\mathcal{I}}$, is the first alias, different from $s_{\alpha_i}$ for any $i$, along the path $(x_0, \ldots, x_q)$. By applying Proposition 2 $u$ times only, we can conclude that, there exists a formula $\exists((id(C_{0,1}) \circ \ldots \circ id(C_{0,g_0})) \circ Q_1 \circ \ldots \circ (id(C_{u-1,1}) \circ \ldots \circ id(C_{q-1,g_{u-1}})) \circ Q_u).(\exists R'.C')$ with $g_i \geq 0$, such that:

- all tests $id(C_{i,j})$ occur in $R$, and hence $C_{i,j} \in CL(\exists R.C') \subseteq CL(\mathcal{T}'_{\mathcal{K}})$;
- $R' \in Post(R)$, and hence the concept $\exists R'.C'$ is equivalent to $D$ for some $D \in CL(\exists R.C') \subseteq CL(\mathcal{T}'_{\mathcal{K}})$;
- $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}}$, for $i = 1, \ldots u$;
- $(x_u, \ldots, x_q) \in Paths_{\mathcal{I}}(R')$;
- $\exists((id(C_{0,1}) \circ \ldots \circ id(C_{0,g_0})) \circ Q_1 \circ \ldots \circ (id(C_{q-1,1}) \circ \ldots \circ id(C_{q-1,g_{q-1}})) \circ Q_q).(\exists R'.C') \sqsubseteq \exists R.C'$ is valid.

Since the path $(x_u, \ldots, x_q)$ contains $k$ aliases, by Lemma 4, from each alias of $\alpha_j$ there is a path satisfying $\exists R'.C'$ which goes through exactly the "same" $k$ aliases in the same order. Let $(s_{\alpha_i} = x'_u, \ldots, x'_{q'})$ be such a path. This path contains less than $k$ aliases, excluding $x'_u$. Thus, by path induction hypothesis, $s_{\alpha_i} \in (\exists R'.C')^{\mathcal{I}'}$.

Now, by construction of $\mathcal{I}'$, $(x_{u-1}, x_u) \in Q_u^{\mathcal{I}}$ implies $(x_{u-1}, s_{\alpha_i}) \in Q_u^{\mathcal{I}'}$ thus $x_{u-1} \in (\exists Q_u.(\exists R'.C'))^{\mathcal{I}'}$. Whereas, by formula induction hypothesis, for all $C_{i,j}$, $x_i \in C_{i,j}^{\mathcal{I}}$ iff $x_i \in C_{i,j}^{\mathcal{I}'}$. Hence considering that for $i = 1, \ldots, u-1$, $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}}$ implies $(x_{i-1}, x_i) \in Q_i^{\mathcal{I}'}$, we get $x \in (\exists R.C')^{\mathcal{I}'}$. □

We can now state the main theorem on reasoning in $\mathcal{CIQ}$ knowledge bases.

**Theorem 6** *A $\mathcal{CIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfiable iff its reduced form $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ is satisfiable. Thus, satisfiability of $\mathcal{CIQ}$ knowledge bases is EXPTIME-complete.*

**Proof** $\Rightarrow$ We can extend a model $\mathcal{I}$ of $\mathcal{K}$ to a model of $\mathcal{K}'$ by adding the individual $g$ to $\Delta^{\mathcal{I}}$, and letting $create^{\mathcal{I}} = \{(g, \alpha_i) \mid i = 1, \ldots, m\}$.

$\Leftarrow$ If there exists a model $\mathcal{I}$ of $\mathcal{K}'$ then by Lemma 5 we can construct an interpretation $\mathcal{I}'$ such that (1) $\mathcal{I}'$ satisfies all inclusion assertions in $\mathcal{T}$; (2) to each individual $\alpha_i$ occurring in $\mathcal{K}$, it corresponds exactly one individual $s_{\alpha_i}$ of $\mathcal{I}'$, and for such $s_{\alpha_i}$ we have $s_{\alpha_i} \in C^{\mathcal{I}}$ for each instance assertion $C(\alpha_i)$ in $\mathcal{K}$, and $(\alpha_i, \alpha_j) \in P^{\mathcal{I}'}$ for each instance assertion $P(\alpha_i, \alpha_j)$. Hence $\mathcal{I}'$ satisfies $\mathcal{K}$.

Thus, the satisfiability of $\mathcal{K}$ is polynomially reducible to satisfiability of its reduced form $\mathcal{K}'$, i.e. to satisfiability of a concept (namely the concept in $\mathcal{A}'$) in a $\mathcal{CIQ}$ TBox (namely $\mathcal{T}'$), which is known to be EXPTIME-complete (De Giacomo & Lenzerini, 1995; De Giacomo, 1995). □

# 7 DISCUSSION AND CONCLUSION

We have presented a new technique for reasoning in a DL system with full capabilities, showing that reasoning in this logic is EXPTIME-complete. The technique is based on a careful encoding of instance assertions into special TBox assertions, that treat individuals as mutually disjoint atomic concepts, and add suitable constraints by exploiting the capability of $\mathcal{CIQ}$ to express complex properties of role chains. We stress the importance of these additional constraints: indeed, a naive translation of individuals into atomic concepts, like the one implicitly done in CLASSIC (Borgida & Patel-Schneider, 1994), would not be sufficient for our purposes.

Consider the knowledge base $\mathcal{K}$ constituted by the TBox and the ABox shown in Figure 4.

The first inclusion assertion constrains the role **succ** and its inverse to be functionl.

The second inclusion assertion states that every individual is linked by a ($\mathbf{succ} \sqcup \mathbf{succ}^-$)-chain to some instance of C. In fact the existence of a single instance of C for each ($\mathbf{succ} \sqcup \mathbf{succ}^-$)-connected part of the model is sufficient to satisfy the above constraint.

The third inclusion assertion states that the instances of C have a **succ**-successor in D and a **succ**-predecessor in $\neg$D.

The assertions in the ABox express that a has b as **succ**-successor, and b has a as **succ**-successor.

The knowledge base $\mathcal{K}$ is unsatisfiable. Indeed, both a and b must be connected by a ($\mathbf{succ} \sqcup \mathbf{succ}^-$)-chain to an instance of C, hence either a or b must be an instance of C. Suppose that b is an instance of C. Then its **succ**-successor, which is a, is an instance of D, and

| TBox: | ABox: | naive ABox encoding |
|---|---|---|
| $\top \sqsubseteq (\leq 1\,\mathtt{succ}.\top) \sqcap (\leq 1\,\mathtt{succ}^-.\top)$<br>$\top \sqsubseteq \exists(\mathtt{succ} \sqcup \mathtt{succ}^-)^*.\mathtt{C}$<br>$\mathtt{C} \sqsubseteq \exists\mathtt{succ}.\mathtt{D} \sqcap \exists\mathtt{succ}.\neg\mathtt{D}$ | succ(a, b)<br>succ(b, a) | $\mathtt{A} \sqsubseteq \exists\mathtt{succ}.\mathtt{B}$<br>$\mathtt{B} \sqsubseteq \exists\mathtt{succ}^-.\mathtt{A}$<br>$\mathtt{B} \sqsubseteq \exists\mathtt{succ}.\mathtt{A}$<br>$\mathtt{A} \sqsubseteq \exists\mathtt{succ}^-.\mathtt{B}$ |

Figure 4: Example: naive ABox encoding fails.

its succ-predecessor, which is again a, is an instance of $\neg\mathtt{D}$. But this is a contradiction. The conclusion is reached if we assume that a is an instance of C.

Now consider the knowledge base $\mathcal{K}'$ obtained from $\mathcal{K}$ by substituting the assertions in the ABox with their naive encodings (see Fig. 4, where A and B are disjoint). It is easy to see that $\mathcal{K}'$ is satisfiable. Indeed, to see this, it suffices to consider the following interpretation: $\Delta^{\mathcal{I}} = \mathtt{a}, \mathtt{a}', \mathtt{b}, \mathtt{b}',\ \mathtt{a}, \mathtt{a}' \in \mathtt{A}^{\mathcal{I}},\ \mathtt{b}, \mathtt{b}' \in \mathtt{B}^{\mathcal{I}}$, $(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a}'), (\mathtt{a}', \mathtt{b}'), (\mathtt{b}', \mathtt{a}) \in \mathtt{succ}^{\mathcal{I}}$, and such that $\mathtt{b} \in \mathtt{C}^{\mathcal{I}}$, $\mathtt{a} \in \mathtt{D}^{\mathcal{I}}$ and $\mathtt{a}' \in \neg\mathtt{D}^{\mathcal{I}}$.

Observe that if we include the assertions:

$$(\mathtt{A} \sqcap \mathtt{D}) \sqsubseteq \forall \mathtt{u}.(\neg\mathtt{A} \sqcup \mathtt{D})$$

for all $\mathtt{D} \in CL(\mathcal{K})$, then the above interpretation is not a model anymore.

By virtue of the characteristics of the encoding presented in this paper, it can be shown that the technique can be extended to even more powerful DLs, such as $\mathcal{CATS}$ and $\mathcal{CVL}$ (see (De Giacomo & Lenzerini, 1995; Calvanese et al., 1995)), which include role conjunction and a limited form of role-value map. Here, we restricted our attention to $\mathcal{CIQ}$ for the sake of simplicity.

In the future, we aim at extending our analysis to the ONE-OF construct (by which we can form a concept as a set of individuals). Although we know that reasoning is still EXPTIME decidable if we add ONE-OF and get rid off of either inverse roles or number restrictions (De Giacomo & Lenzerini, 1994a; De Giacomo, 1995), the decidability of reasoning on $\mathcal{CIQ}$ knowledge bases extended with ONE-OF is still an open problem.

## References


Artale, A., & Franconi, E. (1994). A computational account for a description logic of time and action. In Doyle, J., Sandewall, E., & Torasso, P. (Eds.), *Proc. of KR-94*, pp. 3–14 Bonn. Morgan Kaufmann, Los Altos.

Baader, F. (1991). Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of IJCAI-91* Sydney, Australia.

Baader, F., Hollunder, B., Nebel, B., Profitlich, H.-J., & Franconi, E. (1992). An empirical analysis of optimization techniques for terminological representation systems. In *Proc. of KR-92*, pp. 270–281. Morgan Kaufmann, Los Altos.

Bergamaschi, S., & Sartori, C. (1992). On taxonomic reasoning in conceptual design. *ACM Trans. on Database Systems, 17*(3), 385–422.

Blanco, J. M., Illarramendi, A., & Goni, A. (1994). Building a federated database system: An approach using a knowledge based system. *J. of Intelligent and Cooperative Information Systems, 3*(4), 415–455.

Borgida, A. (1995). Description logics in data management. *IEEE Trans. on Knowledge and Data Engineering, 7*(5), 671–682.

Borgida, A., & Patel-Schneider, P. F. (1994). A semantics and complete algorithm for subsumption in the CLASSIC description logic. *J. of Artificial Intelligence Research, 1*, 277–308.

Buchheit, M., Donini, F. M., & Schaerf, A. (1993). Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research, 1*, 109–138.

Calvanese, D., De Giacomo, G., & Lenzerini, M. (1995). Structured objects: Modeling and reasoning. In *Proc. of DOOD-95*, No. 1013 in LNCS, pp. 229–246. Springer-Verlag.

Catarci, T., & Lenzerini, M. (1993). Representing and using interschema knowledge in cooperative information systems. *J. of Intelligent and Cooperative Information Systems, 2*(4), 375–398.

De Giacomo, G. (1995). *Decidability of Class-Based Knowledge Representation Formalisms.* Ph.D. thesis, Dip. di Inf. e Sist., Univ. di Roma "La Sapienza".

De Giacomo, G. (1996). Eliminating converse from Converse PDL. *J. of Logic, Language and Information.* To appear.

De Giacomo, G., & Lenzerini, M. (1994a). Boosting the correspondence between description logics and propositional dynamic logics. In *Proc. of AAAI-94*, pp. 205–212. AAAI Press/The MIT Press.

De Giacomo, G., & Lenzerini, M. (1994b). Concept language with number restrictions and fixpoints, and its relationship with $\mu$-calculus. In *Proc. of ECAI-94*, pp. 411–415.

De Giacomo, G., & Lenzerini, M. (1995). What's in an aggregate: Foundations for description logics with tuples and sets. In *Proc. of IJCAI-95*, pp. 801–807.

Devambu, P., Brachman, R. J., Selfridge, P. J., & Ballard, B. W. (1991). LASSIE: A knowledge-based software information system. *Comm. of the ACM, 34*(5), 36–49.

Donini, F. M., Lenzerini, M., Nardi, D., & Nutt, W. (1991a). The complexity of concept languages. In Allen, J., Fikes, R., & Sandewall, E. (Eds.), *Proc. of KR-91*, pp. 151–162. Morgan Kaufmann, Los Altos.

Donini, F. M., Lenzerini, M., Nardi, D., & Nutt, W. (1991b). Tractable concept languages. In *Proc. of IJCAI-91*, pp. 458–463 Sydney.

Donini, F. M., Lenzerini, M., Nardi, D., & Schaerf, A. (1994). Deduction in concept languages: From subsumption to instance checking. *J. of Logic and Computation, 4*(4), 423–452.

Donini, F. M., Lenzerini, M., Nardi, D., & Schaerf, A. (1996). Reasoning in description logics. In Brewka, G. (Ed.), *Foundation of Knowledge Representation*. Cambrige University Press. To appear.

Doyle, J., & Patil, R. S. (1991). Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *AIJ, 48*, 261–297.

Fischer, M. J., & Ladner, R. E. (1979). Propositional dynamic logic of regular programs. *J. of Computer and System Sciences, 18*, 194–211.

Kozen, D., & Tiuryn, J. (1990). Logics of programs. In Leeuwen, J. V. (Ed.), *Handbook of Theoretical Computer Science - Formal Models and Semantics*, pp. 789–840. Elsevier.

Levy, A. Y., Rajaraman, A., & Ordille, J. J. (1996). Query answering algorithms for information agents. In *Proc. of AAAI-96*.

Nebel, B. (1988). Computational complexity of terminological reasoning in BACK. *AIJ, 34*(3), 371–383.

Nebel, B. (1991). Terminological cycles: Semantics and computational properties. In Sowa, J. F. (Ed.), *Principles of Semantic Networks*, pp. 331–361. Morgan Kaufmann, Los Altos.

Patel-Schneider, P. F. (1989). A four-valued semantics for terminological logic. *AIJ, 38*(1), 319–351.

Schaerf, A. (1994). Reasoning with individuals in concept languages. *Data and Knowledge Engineering, 13*(2), 141–176.

Schild, K. (1991). A correspondence theory for terminological logics: Preliminary report. In *Proc. of IJCAI-91*, pp. 466–471 Sydney, Australia.

Schild, K. (1994). Terminological cycles and the propositional $\mu$-calculus. In Doyle, J., Sandewall, E., & Torasso, P. (Eds.), *Proc. of KR-94*, pp. 509–520 Bonn. Morgan Kaufmann, Los Altos.

Schmidt-Schauß, M. (1989). Subsumption in KL-ONE is undecidable. In Brachman, R. J., Levesque, H. J., & Reiter, R. (Eds.), *Proc. of KR-89*, pp. 421–431. Morgan Kaufmann, Los Altos.

Schmidt-Schauß, M., & Smolka, G. (1991). Attributive concept descriptions with complements. *AIJ, 48*(1), 1–26.

Sheth, A., Gala, S., & Navathe, S. (1993). On automatic reasoning for schema integration. *J. of Intelligent and Cooperative Information Systems, 2*(1), 23–50.

Weida, R., & Litman, D. (1992). Terminological reasoning with constraint networks and an application to plan recognition. In *Proc. of KR-92*, pp. 282–293. Morgan Kaufmann, Los Altos.

Woods, W. A., & Schmolze, J. G. (1992). The KL-ONE family. In Lehmann, F. W. (Ed.), *Semantic Networks in Artificial Intelligence*, pp. 133–178. Pergamon Press. Published as a special issue of *Computers & Mathematics with Applications*, Volume 23, Number 2–9.