Decision Procedures for Expressive Description Logics with Intersection, Composition, Converse of Roles and Role Identity*

Fabio Massacci
Dip. di Ingegneria dell'Informazione – Università di Siena
via Roma 56 – 53100 Siena – Italy
massacci@dii.unisi.it

Abstract

In the quest for expressive description logics for real-world applications, a powerful combination of constructs has so far eluded practical decision procedures: intersection and composition of roles.

We propose tableau-based decision procedures for the satisfiability of logics extending $\mathcal{ALC}$ with the intersection $\sqcap$, composition $\circ$, union $\sqcup$, converse $\cdot^{-}$ of roles and role identity $id(\cdot)$. We show that

1. the satisfiability of $\mathcal{ALC}(\sqcap,\circ,\sqcup)$, for which a 2-EXPTIME upper bound was given by tree-automata techniques, is PSPACE-complete;
2. the satisfiability of $\mathcal{ALC}(\sqcap,\circ,\sqcup,\cdot^{-},id(\cdot))$, an open problem so far, is in NEXPTIME.

1 Introduction and Motivations

Description Logics (DLs) are a popular knowledge representation formalism based on *concepts* and *roles*, where concepts model classes of individuals, and roles model relationships between individuals [Baader *et al.*, 2001, Chap. 1-2].

In the last years, the investigation on DLs has been driven by the modeling needs of applications ranging from semi-structured data to planning [Baader *et al.*, 2001, Part III], which have stimulated the usage of expressive constructs. For instance, to model semi-structured data one needs to represent arbitrary relations (graphs with labeled edges), and use constructs for stating irreflexivity of relations, their transitive closure, or their intersection. For reasoning about actions, one may want to express the fact that two long sequences of actions (roles) must be completed in parallel.

These applications have called for decision procedures to make reasoning services up to the modeling duties (see [Baader *et al.*, 2001, Chap. 2] or Sec. 6). Yet, a powerful combination of constructs has so far eluded this quest: composition and intersection of roles. There is only an involved automata-based algorithm by Danecki [1984] giving a 2-EXPTIME upper bound.

*This work was done while the author was on leave at IRIT - Toulouse partly supported by CNR grant 203-7-27. I am greatly indebted to P. Balbiani for directing my attention to logics with intersection and for countless suggestions and discussions. Discussions with F. M. Donini, I. Horrocks and U. Sattler were helpful.

Decision procedures are hard to find because DLs with intersection and composition (even without converse and role identity[1]) haven't the tree model property. With composition and intersection, we can write concepts whose models are directed acyclic graphs. Adding role identity, we can force a relation to be well-founded or a model to be a cyclic graph.

For instance, suppose we want to model the tangled web of corporate ownerships by using the roles owns, app-board, app-CEO (denoting that a corporation owns another company, appoints the company's board of directors, or its CEO). We can represent the corporations having a doubly indirectly controlled subsidiary with the concept

$$\mathsf{corp} \sqcap \exists(\mathsf{app\text{-}board} \circ \mathsf{owns}) \sqcap (\mathsf{owns} \circ \mathsf{app\text{-}CEO}).\mathsf{corp}$$

If regulators forbid corporations to be owners of themselves, we can forbid it too, without ad-hoc well-founded constructs:

$$\forall \mathsf{owns} \sqcap id(\mathsf{corp}).\bot$$

We can also model, self-owned "Chinese-box" corporations:

$$\exists(\mathsf{owns} \circ \mathsf{has\text{-}shares} \circ \mathsf{app\text{-}board}) \sqcap id(\mathsf{corp}).\top$$

These models are not representable with the "classical" expressive DLs such as $\mathcal{ALC}_{reg}$ or $\mathcal{DLR}$.

Here, we consider the DL $\mathcal{ALC}(\sqcap,\circ,\sqcup,\cdot^{-},id(\cdot))$ which, in addition to the constructs of $\mathcal{ALC}$, provides *intersection, composition, union, converse of roles* and *role identity*. $\mathcal{ALC}(\sqcap,\circ,\sqcup)$ is the fragment without inverse and identity.

These logics are siblings of three extensions of $\mathcal{ALC}$: $\mathcal{ALC}_{reg}$ with composition and transitive/reflexive closure of roles but without intersection [Baader *et al.*, 2001, Chap.2]; $\mathcal{ALB}$ with intersection, union and negation [Lutz and Sattler, 2000], and possibly converse [Hustadt and Schmidt, 2001]; $\mathcal{DLR}$ [Calvanese *et al.*, 1998] with intersection and role difference but only on atomic roles and with additional limitations. In the correspondence with PDL by Schild [1991], $\mathcal{ALC}(\sqcap,\circ,\sqcup)$ corresponds to the the *-free and test-free fragment of IPDL and $\mathcal{ALC}(\sqcap,\circ,\sqcup,\cdot^{-},id(\cdot))$ is the *-free fragment of Converse-IPDL [Harel, 1984].

In the next sections we recall some preliminaries. Then we present our tableau calculus (Sec.3), transform it into algorithms (Sec.4), sketch the complexity results (Sec.5), and discuss related works (Sec.6).

[1] Role identity is the construct which, given a concept C, allows one to build a role connecting each instance of C to itself.

2 Preliminaries

Let A and P denote atomic concepts and atomic roles respectively. Concepts C, D and roles R, S are formed as follows:

$$\begin{array}{rcl} C, D & ::= & A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \\ R, S & ::= & P \mid R \sqcup S \mid R \circ S \mid R \sqcap S \mid id(C) \mid R^- \end{array}$$

A *TBox* $\mathcal{T}$ is a finite set of *inclusions* $C \sqsubseteq D$. In the sequel we focus on the satisfiability of concepts wrt empty TBoxes. However, we will build non-empty TBoxes of inclusions with a special form during the proof search itself.

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$, the *domain* of $\mathcal{I}$ — whose members are called *elements* — and a function $\cdot^{\mathcal{I}}$, the *interpretation function* of $\mathcal{I}$, that maps every concept to a subset of $\Delta^{\mathcal{I}}$ and every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. We refer to Borgida [1996] or Baader et al. [2001, Chap. 2] for details.

An interpretation $\mathcal{I}$ *satisfies a concept* C if there exists an element $d \in \Delta^{\mathcal{I}}$ such that $d \in C^{\mathcal{I}}$, i.e. if $C^{\mathcal{I}} \neq \emptyset$.

3 A Tableau Calculus

Taming intersection and composition requires the novel combination of a number of intuitions from the literature and some new features. They are highlighted here, for the reader familiar with the literature.

- We use the *graph-based representation of rules* for tableau-structures originally due to Kripke and used by Castillo et al. [1997] and Horrocks and Sattler [1999].
- We borrow the idea by Danecki [1984] of *pebble games* to mark elements linked by both R and S roles so that the *labeling an element with both pebbles marks the intersection*, triggering the insertion of suitable concepts;
- We *internalize pebbles in the calculus by introducing new propositional constants* as done by De Giacomo and Massacci [1997] for eventualities in CPDL.
- We use *skolemization for generating "new" nodes and atomic concepts* in the proof search, exploiting the results from DL translations into first-order logics [Hustadt and Schmidt, 2001].
- We *exploit some semantical properties of intersection* by Balbiani and Vakarelov [2001] to show we are sound.
- We use the idea of *on-the fly modification of TBoxes* proposed by Massacci [1998] for the universal modality.
- We borrow the idea of *lazy unfolding of axioms* proposed by Horrocks and Tobies [2000] for plain $\mathcal{ALC}$.
- We use a *new role name to denote equivalence of individuals* but only for the proof search.

We use *tableau-structures* i.e. labeled graphs:

$$\mathcal{S} = \langle \mathcal{N}, \mathcal{E}, \mathcal{C}(\cdot), \mathcal{R}(\cdot, \cdot) \rangle$$

where $\mathcal{N}$ is a finite nonempty set of nodes denoted by x, y, z possibly with indices; $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of oriented edges; $\mathcal{C}(\cdot)$ maps nodes to sets of concepts, and $\mathcal{R}(\cdot, \cdot)$ maps edges to sets of roles. We also use an initially empty TBox $\mathcal{T}$. Inclusions of the form $A_{x,R} \sqcap A_{x,S} \sqsubseteq C$ are added on-the-fly.

Axiom: If $x{:}A_1 \in \mathcal{S}$, $x{:}A_2 \in \mathcal{S}$, and $A_1 \sqcap A_2 \sqsubseteq C \in \mathcal{T}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}C\}$.

Conjunction: If $x{:}C \sqcap D \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}C, x{:}D\}$.

Disjunction: If $x{:}C \sqcup D \in \mathcal{S}$ then non-deterministically either $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}C\}$ or $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}D\}$.

Universal atomic roles: If $x : \forall P.C \in \mathcal{S}$, $\langle x, y \rangle : P \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{y{:}C\}$.

Universal role concatenation: If $x : \forall R \circ S.C \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}\forall R.\forall S.C\}$.

Universal role intersection: If $x : \forall R \sqcap S.C \in \mathcal{S}$ then let $A_{x,R}$ and $A_{x,S}$ be new atomic concepts and $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}\forall R.A_{x,R}, x{:}\forall S.A_{x,S}\}$ and $\mathcal{T} \Rightarrow \mathcal{T} \cup \{A_{x,R} \sqcap A_{x,S} \sqsubseteq C\}$.

Universal role union: If $x : \forall R \sqcup S.C \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}\forall R.C, x{:}\forall S.C\}$.

Existential restriction: If $x{:}\exists R.C \in \mathcal{S}$ then let y be a new node and $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle x, y \rangle{:}R, y{:}C\}$.

Role concatenation: If $\langle x, y \rangle : R \circ S \in \mathcal{S}$ then let z be a new node and $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle x, z \rangle{:}R, \langle z, y \rangle{:}S\}$.

Role intersection: If $\langle x, y \rangle : R \sqcap S \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle x, y \rangle{:}R, \langle x, y \rangle{:}S\}$.

Role union: If $\langle x, y \rangle : R \sqcup S \in \mathcal{S}$ then non-deterministically either $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle x, y \rangle{:}R\}$ or $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle x, y \rangle{:}S\}$.

Figure 1: The reduction rules for $\mathcal{ALC}(\sqcap, \circ, \sqcup)$.

We start with a structure with a single node labeled with an input concept C, and apply the tableau rules to build a reduced structures without contradictions that can be used as a model for C. If none can be build, C is unsatisfiable.

The *reduction rules* in Fig. 1 are applicable to a tableau-structure $\mathcal{S}$ and a set of inclusions $\mathcal{T}$ for $\mathcal{ALC}(\sqcap, \circ, \sqcup)$. W.l.o.g. we assume that negation and converse are pushed down to atomic concepts and roles. We also abuse the DL syntax for tableau systems [Buchheit *et al.*, 1993]: we write $x : C \in \mathcal{S}$ to indicate that $C \in \mathcal{C}(x)$, and $\langle x, y \rangle : R \in \mathcal{S}$ when $R \in \mathcal{R}(x, y)$. Thus, by $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}C\}$ we mean that we obtain a new structure $\mathcal{S}'$ such that $\mathcal{N}' = \mathcal{N} \cup \{x\}$, $\mathcal{E}' = \mathcal{E}$, $\mathcal{C}'(y) = \mathcal{C}'(y)$ if $y \neq x$ and $\mathcal{C}'(x) = \mathcal{C}(x) \cup \{C\}$, $\mathcal{R}'(\cdot, \cdot) = \mathcal{R}(\cdot, \cdot)$. Similarly for roles.

Most rules are close to other tableau approaches [Buchheit *et al.*, 1993; Castilho *et al.*, 1997; De Giacomo and Massacci, 1997; Horrocks and Sattler, 1999]. A difference is that they only label edges with atomic roles whereas we also use complex roles. So, we have rules for reducing these roles.

The role of *pebbles* is played by the new atoms introduced by the universal role intersection rule. The idea is that since $A_{x,R}$ is new, it can label a node y only by reducing $\forall R.A_{x,R}$. thus y must be an R-successors of x. If both $A_{x,R}$ and $A_{x,S}$ label node y, then y is linked to x by an $R \sqcap S$ role. Then we can add C to y.

Since we cannot forecast where $A_{x,R}$ and $A_{x,S}$ will appear, we must potentially check every node. To internalize this check into the calculus, we use the idea of *on-the-fly modification of TBoxes* by [Massacci, 1998]. With the intersection

rule we "update" $\mathcal{T}$ with the inclusion $A_{x,R} \sqcap A_{x,S} \sqsubseteq C$ and then we use the Axiom rule to add C to nodes on demand.

Notice that we apply the Axiom rule only when both $A_{x,R}$ and $A_{x,S}$ are present in the same node. This technique, borrowed from the lazy unfolding of Horrocks and Tobies [2000], works because our inclusions are acyclic.

Notice also that the new concept $A_{x,R}$ introduced by the reduction of $x : \forall R \sqcap S.C$ must depend on the node where the concept is located. Following Danecki [1984, Pag. 44], one may think that it suffices to make them dependent only on the subformula (e.g. $A_{R,\forall R \sqcap S.C}$), possibly distinguishing between occurrences. This would be unsound. The concept

$$(\forall R.\forall R \sqcap S.C) \sqcap (\exists (R \circ R) \sqcap (R \circ S).\neg C)$$

is satisfiable but introducing the same "new" concepts at different nodes for the reduction of the only occurrence of $\forall R \sqcap S.C$ would result in a "clash". By applying the rules of Fig. 1, one can see that the pebbles $A_{R,\forall R \sqcap S.C}$ and $A_{S,\forall R \sqcap S.C}$ would propagate along the "wrong" path.

The generation of new nodes and new atomic concept symbols may clearly lead to redundant rule applications or even to a non-terminating process. However, we do not need to generate really "new" nodes or atomic concepts. We can use the *equivalent of skolemization* (or Hilbert's ϵ-terms) introduced by Ohlbach and later refined by Hustadt and Schmidt [2001] for the translations of DLs into first-order logic.

The generation of skolem functions is standard and we assume that for "new" concepts we use the function GENCONCEPT(x, R), and for "new" nodes we use GENNODECONCEPT(x, C) and GENNODEROLE(x, y, R). So, when reducing $x : \exists R.C$, we call $y = \text{GENNODECONCEPT}(x, \exists R.C)$ and add $y : C$ and $\langle x, y \rangle : R$ to the structure.

Definition 1 *A tableau structure $\mathcal{S}$ and a set of inclusions $\mathcal{T}$ are* reduced for a rule R *if the application of* R *maps $\mathcal{S}$ and $\mathcal{T}$ into themselves. A structure and a set of inclusions are* reduced *if they are reduced for all rules.*

Definition 2 *A tableau-structure contains a clash if there is a node x and a concept C such that $\{C, \neg C\} \subseteq \mathcal{C}(x)$.*

Theorem 1 *A concept C of $\mathcal{ALC}(\sqcap, \circ, \sqcup)$ is satisfiable iff there is a non-deterministic application of the rules in Fig. 1 to an initially empty TBox and a structure with only one node labeled with C leading to a reduced and clash-free structure.*

For every feature, the proof cleverly combines the techniques from the cited works in the literature. The twist is the soundness of the universal role intersection rule. To this extent, we set $A^{\mathcal{I}}_{x,R} = \{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\}$ and similarly for S. In words, the newly introduced concepts are fulfilled by the appropriate individuals R-reachable, or S-reachable, from x. Due to lack of space, details are left to the full paper.

Next, we define the tableau rules for the full language. Beside converse, the tricky bit is the presence of the intersection between role identity and complex roles: $id(C) \sqcap R$ is a self-loop describing individuals in the class C that are in relation R with themselves (e.g. the self-owned companies). To this extent we employ an atomic role symbol (not occurring in the input concept) for equality of nodes $\cdot \approx \cdot$.

Universal converse of roles: If $y : \forall R^{-}.C \in \mathcal{S}$, $\langle x, y \rangle : R \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}C\}$.

Universal role identity: If $x : \forall id(D).C \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x{:}\neg D \sqcup C\}$.

Role converse: If $\langle x, y \rangle{:}R^{-} \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{\langle y, x \rangle{:}R\}$.

Role identity: If $\langle x, y \rangle{:}id(C) \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x \approx y, x{:}C\}$

Role identity distribution: If $x \approx y \in \mathcal{S}$ and $x : C \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{y{:}C\}$.

Role identity symmetry: If $x \approx y \in \mathcal{S}$ then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{y \approx x\}$.

Role identity transitivity: If $x \approx y \in \mathcal{S}$ and $y \approx z \in \mathcal{S}$ for then $\mathcal{S} \Rightarrow \mathcal{S} \cup \{x \approx z\}$.

Figure 2: The additional reduction rules for $\cdot^{-}$ and $id(\cdot)$

Algorithm WORLDS;
input node $\underline{x}_0$; set of concepts $\mathcal{C}$; set of inclusions $\mathcal{T}$;
output *sat* if $\mathcal{C}$ is satisfiable, *unsat* otherwise;
variables node x; labeled graph $\mathcal{S}$;
begin
  $\mathcal{N}$= $\{\underline{x}_0\}$; $\mathcal{E}$= $\emptyset$; $\mathcal{C}(\cdot)$= $\{\langle \underline{x}_0, \mathcal{C} \rangle\}$; $\mathcal{R}(\cdot, \cdot)$ = $\emptyset$;
  if CLASH($\mathcal{S}$) **then return**(*unsat*);
  while x=CHOOSENODE($\mathcal{S}$,$\mathcal{T}$)=/=*none* **do**
    R=CHOOSERULE(x,$\mathcal{S}$,$\mathcal{T}$);
    "Apply R to x, $\mathcal{S}$, $\mathcal{T}$ updating $\mathcal{S}$ and $\mathcal{T}$"
    if CLASH($\mathcal{S}$) **then return**(*unsat*);
  forall $x \in \mathcal{N} \setminus \{\underline{x}_0\}$ **do**
    if WORLDS(x,$\mathcal{C}(x)$,$\mathcal{T}$)==*unsat*
    then return(*unsat*);
  return *sat*
end;

Figure 3: The algorithm for $\mathcal{ALC}(\sqcap, \circ, \sqcup)$

The *reduction rules* in Figure 1 and the additional rules in Figure 2 are applicable to a tableau-structure $\mathcal{S}$ and a set of inclusions $\mathcal{T}$ for the logic $\mathcal{ALC}(\sqcap, \circ, \sqcup, \cdot^{-}, id(\cdot))$.

The rules for $\cdot \approx \cdot$ might be eliminated if the rule for $id(\cdot)$ collapses nodes. However, this would complicate the algorithms, as we would need a unique way for collapsing nodes.

Theorem 2 *A concept C of $\mathcal{ALC}(\sqcap, \circ, \sqcup, \cdot^{-}, id(\cdot))$ is satisfiable iff there is a non-det. application of the rules in Fig. 1,2 to an initially empty TBox and a structure with only one node labeled with C yielding a reduced and clash-free structure.*

4 From Calculi to Algorithms

The PSPACE-algorithm for $\mathcal{ALC}(\sqcap, \circ, \sqcup)$ is called WORLDS after Ladner's WORLD for modal logic K. It is plural because each call works on a fragment of a Kripke model and not just on a set of concepts true for an individual. It is in Fig. 3.

To check the satisfiability of a concept C one calls WORLDS(0,$\{C\}$,$\emptyset$) and applies the rules from Fig. 1.

In the algorithms, we use the symbol "=" for assignment, and "==" for equality testing. We assume all functionalities for working with labeled direct graphs, an auxiliary function CLASH($\mathcal{S}$) which detects clashes, and auxiliary procedures for selecting objects: CHOOSENODE selects a node to be reduced; CHOOSERULE selects an applicable rule for a node.

Algorithm CWORLDS;
input node $\underline{x}_0$; set of concepts $\mathcal{C}$;
output set of concepts $\mathcal{D}\supset\mathcal{C}$ if $\mathcal{C}$ is satisfiable, *unsat* otherwise;
variables node x; $\mathcal{S}$ labeled graph; $\mathcal{D}$ set of concepts;
global variables set of inclusions $\mathcal{T}$;
begin
$\mathcal{N}$= $\{\underline{x}_0\}$; $\mathcal{E}$= $\emptyset$; $\mathcal{C}(\cdot)$= $\{\langle\underline{x}_0,\mathcal{C}\rangle\}$; $\mathcal{R}(\cdot,\cdot)$ = $\emptyset$;
if CLASH($\mathcal{S}$) **then return** *unsat*;
start: **while** x=CHOOSENODE($\mathcal{S}$,$\mathcal{T}$)=/=*none* **do**
R=CHOOSERULE(x,$\mathcal{S}$,$\mathcal{T}$)
"Apply R to x, $\mathcal{S}$ and $\mathcal{T}$ updating $\mathcal{S}$ and $\mathcal{T}$"
if CLASH($\mathcal{S}$) **then return** *unsat*;
forall $x \in \mathcal{N} \setminus \{\underline{x}_0\}$ **do**
$\mathcal{D}$=CWORLDS(x,$\mathcal{C}(x)$);
if $\mathcal{D}$==*unsat* **then return** *unsat*;
else if $\mathcal{C}(x)$=/= $\mathcal{D}$;
then $\mathcal{C}(x)$= $\mathcal{D}$;
goto start;
return($\mathcal{C}(\underline{x}_0)$)
end;

Figure 4: The CWORLDS algorithm

These procedures must work in time polynomial in the size of the input and respect the following constraints:

Search Criteria 1 *A rule is not selected if the structure is already reduced for it. A node is not selected if the structure is already reduced for all rules applicable to the node.*

Search Criteria 2 *The* Existential restriction *rule is selected only for reduction of concepts labeling the initial node* $\underline{x}_0$.

If the procedures cannot select anything respecting the above constraints they return the value *none.*

The tricky bit is showing that the algorithm is in PSPACE (Sec. 5). We just highlight the differences from trace-based methods used for $\mathcal{ALC}$ by Ladner [1977], Schmidt-Schauss and Smolka [1991] or Tobies [2001].

The classical tableau-based algorithm for $\mathcal{ALC}$ would have been identical to ours except for the last **for** cycle:

for all $\exists R.C \in \mathcal{C}(\underline{x}_0)$ **do**
if WORLD($\{C\} \cup \{D \mid \forall R.D \in \mathcal{C}(\underline{x}_0)\}$)==*unsat*
then return(*unsat*)

In words, for $\mathcal{ALC}$ we explore the one-step R-successors of $\underline{x}_0$ by recursively calling the procedure. So, we only examine one node ($\underline{x}_0$) at every call.

The WORLDS algorithm examines the initial node $\underline{x}_0$ and *reduces all existential concepts labeling* $\underline{x}_0$, i.e. all one step R-successors of $\underline{x}_0$. Thus, it builds a fragment of the Kripke structure. However, the *reduction of existential-concepts labeling successors nodes is deferred* to recursive calls. The rules for intersection and composition may transform a "one-step" R-successors into "many-steps" atomic P-successors. Still, the number of intermediate P-steps is linearly bounded by the size of the input concept.

The full algorithm is called CWORLDS because it returns a set of concepts. It is shown on Figure 4. The rules that can be applied to the algorithm are those from Fig. 1,2. It must respect an additional search constraint:

Search Criteria 3 *If* $x_1 \approx x_2,\ x_2 \approx x_3,\ldots x_{n-1} \approx x_n$ *are present in the structure then* only one node $x_i \in \{x_1,\ldots x_n\}$ *can be selected for the recursive call of* CWORLDS *in the* ***forall*** *cycle. If* $\underline{x}_0$ *is among them, none can be selected.*

We must return a set of concepts rather than just *sat* to cope with converse and role identity. The idea is borrowed from modal logics with symmetric relations [Massacci, 2000] and DLs with converse [Tobies, 2001].

For example, the concept $\exists R\circ id(\exists id(\forall R.C).\top)\circ R.\neg C$ is unsatisfiable. If we used algorithm WORLDS with the full set of rules it would return *sat.* The explanation is that we would introduce a node y labeled by $\exists id(\forall R.C).\top$ and would evaluate the existential concept in the next recursive call. In the recursive call we would add $\forall R.C$ to the initial node (which would correspond to y) but we could not use this information to derive a clash in the initial call.

The use of skolemization, rather than truly fresh names, is necessary to guarantee that CWORLDS is both correct and terminating. For instance, if fresh names are used in the recursive calls following a restart, the concept $\exists R.\exists S.\forall S^- \sqcap T.C$ would make CWORLDS non-terminating.

5 Complexity Analysis

We denote by n the *size of the input concept* C which must be proved (un)satisfiable, measured as the number of symbols.

We first prove that WORLDS requires only polynomial space in n. The idea is that $\mathcal{ALC}(\sqcap,\circ,\sqcup)$ *lost the tree-model property but kept the cactus-model property.*

Loosely speaking, our models can be arranged into the shape of a cactus, i.e. a tree in which the edge connecting two nodes of the tree is not a slim branch but a "fat", cactus-like stem. This fat stem is made by other nodes and edges, but its size is polynomially bounded by n. Many stems sprout from each stem, as in a real cactus, but their number is still polynomially bounded. Since the height of the cactus is also polynomially bounded we are done.

Concepts labeling nodes can be seen as spikes. We must also prove that they aren't exponentially many. For most DLs this is obvious: we only introduce subconcepts of the input concept. Here, we introduce new atomic concepts.

The next tree lemmata make these intuitions precise. At first, we say that the cactus height is polynomially bounded:

Lemma 1 *The call stack of* WORLDS *is at most* $O(n)$ *deep.*

Then the size of each cactus stem is bounded:

Lemma 2 *Each invocation of* WORLDS *generate a structure with at most* $O(n)$ *nodes.*

Finally, we say that the the number of spikes is bounded.

Lemma 3 *The number of new atoms occurring as (sub)concepts in the the labels of the nodes of an invocation of* WORLDS *is bounded by* $O(n^3)$.

To prove Lemma 1 we cannot use the standard proof that each interaction reduces the modal depth (nesting of universal and existential roles) of the set of concepts labeling a node because of the Axiom rule. We need a new notion of modal depth: we associate different depths to atomic concepts from the input concept and to "invented" atomic concepts.

- if A is from input concept then $\mathrm{d}(A) = 0$
- if $A_{x,R}$ is an atomic concept introduced by the reduction of one or more $x : \forall R \sqcap S.C$ then $\mathrm{d}(A_{x,R})$ is equal to $\max_{S,C}\{\mathrm{d}(C) | x{:}\forall R \sqcap S.C\}$
- $\mathrm{d}(\forall R.C) = |R| + \mathrm{d}(C)$ where $|R|$ is the size of R, measured as the number of symbols.

Now by induction on the number of applied rules we can show that for all nodes $x \neq \underline{x}_0$ and all concepts $C \in \mathcal{C}(x)$, there is a concept $D \in \mathcal{C}(\underline{x}_0)$ such $\mathrm{d}(D) > \mathrm{d}(C)$. Since the depth of every concept, and hence the maximal depth of a set of concepts, is bounded by n we are done.

For the proof of Lemma 2, observe that new nodes in the structure $\mathcal{S}$ can only be created by two rules: the rules reducing existential and the rules reducing composition of roles. The first rule can only be applied to $\underline{x}_0$, thus bounding the nodes so introduced by $O(n)$. For the second rule, the number of nodes thus introduced is bounded by $O(n)$.

For Lemma 3, observe that two "new" concepts are introduced only when we reduce an universally quantified role intersection. The newly generated concepts depend on the same node and on the roles which are immediate subroles of an intersection of roles occurring in the input concept. Hence, by Lemma 2, for each invocation we can create at most $O(n \cdot n)$ different new concepts. By Lemma 1 the number of nested recursive calls of WORLDS is at most $O(n)$. When a leaf call is reached we generated at most $O(n^2 \cdot n)$ different concepts.

Theorem 3 WORLDS *can be implemented using only polynomial space in* n.

For the proof observe that the stack is at most $O(n)$ deep by Lemma 1 and for each call of WORLDS at depth i we have

- $O(n) + O(n)$ new nodes and at most $O(n^2)$ edges,
- $O(n^4)$ concepts for each node and each concepts taking at most $O(n)$ space,
- $O(n)$ roles labeling each edge and each role taking at most $O(n)$ space
- $O(i \cdot n^5)$ inclusions, which can be added by reducing a concept $x : \forall R \sqcap S.C$ where the number of different xs is bounded by Lemma 2, $R \sqcap S$ occurs in the input concept and C either occurs in the input concept or is one of $O(n^3)$ new concepts invented at any $j \leq i$ steps.

PSPACE-hardness follows from $\mathcal{ALC}$.

Corollary 4 *The satisfiability of* $\mathcal{ALC}(\sqcap, \circ, \sqcup)$ *concepts is* PSPACE*-complete.*

The NEXPTIME-upper bound of $\mathcal{ALC}(\sqcap, \circ, \sqcup, \cdot^{-}, id(\cdot))$ satisfiability, has a more involved proof. The analogous of Lemma 1 and Lemma 2 can be proved with a similar argument. The only twist is observing that, by Criteria 3, we only reduce the existential concepts in one of the node linked by the equality predicate. Thus, we never recursively call CWORLDS on the nodes "equal" to the root node $\underline{x}_0$ (as for them the induction would fail). The equivalent of Lemma 3 fails and we only have a global bound:

Lemma 4 *The total number of new atoms occurring as (sub)concepts in the the labels of the nodes throughout the execution of* WORLDS *is bounded by* $O(n^n) = 2^{O(n \log n)}$.

Using Lemma 4, we bound the number of concepts labeling a formula to $O(n) \cdot 2^{O(n \log n)} = 2^{O(n \log n)}$ and the number of inclusions by doubling the multiplicative constant in the $O(n \log n)$ expression at the exponent.

Theorem 5 CWORLDS *terminates after* $2^{O(n^2 \log n)}$ *time.*

For the proof observe that CWORLDS can be "restarted" at most finitely many times. More precisely, (i) the set of concepts returned by CWORLDS is larger or equal to the initial set upon which CWORLDS is called (in symbols $\mathcal{C}(\underline{x}_0) \supseteq \mathcal{C}$), and (ii) the total number of concepts is bounded by $2^{O(n \log n)}$. Then the total number of restarts for a structure in a single invocation of CWORLDS is bounded by $O(n) \cdot 2^{O(2n \log n)}$. The bound on the stack does the rest.

Corollary 6 *The satisfiability of* $\mathcal{ALC}(\sqcap, \circ, \sqcup, \cdot^{-}, id(\cdot))$ *concepts is in* NEXPTIME.

An *intriguing question* is why the standard technique for giving PSPACE-bounds for DLs with converse [Tobies, 2001] or symmetric modal logics [Massacci, 2000] fails here. According this technique, we take the "converse-free" algorithm and add some "restarts" (this is what CWORLDS does). For the bound, we observe that (i) restarts add a bounded number of new concepts (indeed subconcepts of the input concept) and (ii) the time wasted by restarts doesn't matter.

The argument fails here because converse or role identity *and* intersection plus composition can force CWORLDS to label a node with exponentially many new atomic concepts.

To force this exponential generation of concepts, we start by constructing a concept G_n whose model is a cyclic graph with edges labeled by two roles R and S. Looking only at role R, the graph is a binary tree with height n, with G_n labeling its root. Each S-edge connects the child node of an R-edge to its parent. The key feature of G_n is that we can *start at the root, traverse forward a path of* R*-edges*, reach a leaf and then continue to *traverse* S*-edges forward to the root*. Once there, we can *take another* R*-path*, and so on. The model of G_n has a path whose length is exponential in n. Pictorially, G_n can be seen as a daisy with exponentially many petals.

This construction is impossible in $\mathcal{ALC}$, $\mathcal{ALC}$ with converse, or $\mathcal{ALC}(\sqcap, \circ, \sqcup)$ because there are always acyclic models. In $\mathcal{ALC}$ we can keep on traversing edges (roles) on *one path* until we arrive at dead-end. Since the depth of the longest path is linear in the size of the concept to be verified we are done. In $\mathcal{ALC}$ with converse, edges can be oriented forward or backward: we get a two-ways path but still it terminates into a dead-end. With intersection we must check that *two paths* meet at certain points but the idea is the same.

Still, CWORLDS only uses polynomial space for visiting G_n. The problems start when we add to G_n one or more concepts that ask to verify the intersection of two paths. Loosely speaking, one path reaches the top of a petal, the other goes back to center, then round another petal and back to the top of the first petal. The problem is that we don't know a priori which petal we must visit among the exponentially many that are available. We can put one of these concepts in each leaf. Then, the number of pebbles that accumulate in the root, restart after restart, will be exponential in n.

6 Related Methods

A number of decision procedures and complexity results for logics extending $\mathcal{ALC}$ (or multimodal K) can be found in the literature. Yet, none of them fully tackle our results.

With respect to complexity results based on encodings, De Giacomo and Lenzerini [1995] proved the EXPTIME-completeness of a DL with composition, converse and intersection restricted to atomic roles with additional limitations. Calvanese et al. [1998] proved the EXPTIME-completeness of $D\mathcal{LR}$, where intersection, union and difference of roles are allowed but composition is not permitted. Baader and Sattler [1999] proved the undecidability of a number of combination of expressive DLs with number restrictions and intersection.

With respect to decision procedures, Horrocks et al. [1999; 2000] tamed expressive DLs with converse and role-hierarchies where intersection between atomic roles can be simulated. They do not allow for composition of roles. Baader and Sattler [1999] have proposed an EXPTIME-calculi for $\mathcal{ALC}$ with number restrictions with composition, intersection and union of role chains. However, concepts are restricted to plain $\mathcal{ALC}$. Lutz and Sattler [2000] give a decision procedure for $\mathcal{ALB}$ ($\mathcal{ALC}$ plus intersection, union, and negation of roles but without composition). Tobies [2001] gives a PSPACE-algorithm for DLs with number restrictions, converse and intersection of atomic relations.

In the realm of modal and dynamic logic, Danecki [1984] has shown, using automata-based techniques, that IPDL (PDL with Intersection) can be decided in 2-EXPTIME. IPDL strictly extends the logic $\mathcal{ALC}(\sqcap,\circ,\sqcup)$ allowing for the transitive and reflexive closure of roles and for role identity. However, Danecki claims that it is possible to use pebbles which depends only on the occurrence of the subformula. For our calculus, this is unsound and Danecki's involved construction may need to be checked against our counterexample. For Converse-PDL automata theoretic techniques have been proposed by Vardi and Wolper [1986] and a tableau calculus has been given by De Giacomo and Massacci [1997].

Hustadt and Schmidt [2001] have shown a decision procedures for boolean modal logic with converse. Their procedure is based on a clever translation into first-order logic of modal formulae followed by a decision procedure for the guarded fragment. They only prove decidability results. The possibility of a PSPACE-algorithm for the extension of $\mathcal{ALC}$ with intersection, converse and union is just claimed.

Decision procedures for first order logic with two variables (see the survey by Grädel and Otto [1999]) can be used for DLs with intersection but without composition, via first-order translations. One could also use the decision procedures by Ganzinger and De Nivelle [1999] for the guarded fragment. Current methods based on translations into decidable fragments of first order logic cannot treat intersection and composition at the same time, as this requires to have at least three variables and does not allows for guards in predicate clauses.

References

[Baader and Sattler, 1999] F. Baader and U. Sattler. Expressive number restrictions in description logics. *JLC*, 9:319–350, 1999.

[Baader *et al.*, 2001] F. Baader, D. McGuinness, D. Nardi, and P. Patel Schneider, editors. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge Univ. Press. 2001. To appear.

[Balbiani and Vakarelov, 2001] P. Balbiani and D. Vakarelov. Iteration-free PDL with intersection: a complete axiomatization. *Fundamenta Informaticae*, 2001. To appear.

[Borgida, 1996] A. Borgida. On the relative expressiveness of description logics and predicate logics. *AIJ*, 82:353–367, 1996.

[Buchheit *et al.*, 1993] M. Buchheit, F. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *JAIR*, 1:109–138, 1993.

[Calvanese *et al.*, 1998] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, pp.149–158, 1998.

[Castilho *et al.*, 1997] M. Castilho, L. Fariñas del Cerro, O. Gasquet, and A. Herzig. Modal tableaux with propagation rules and structural rules. *Fund. Inf.*, 32:281–297, 1997.

[Danecki, 1984] R. Danecki. Nondeterministic Propositional Dynamic Logic with Intersection is decidable. In *Proc. of the 5th Sym. on Comp. Theory*, LNCS 208, pp.34–53. Springer, 1984.

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

[De Giacomo and Massacci, 1997] G. De Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for Converse-PDL. *Information and Computation*, 162:117-137, 2001. Short version in *Proc. of CADE-96*.

[Ganzinger and de Nivelle, 1999] H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. of LICS'99*, pp.295–304, 1999.

[Grädel and Otto, 1999] E. Grädel and M. Otto. On logics with two variables. *TCS*, 224:73–113, 1999.

[Harel, 1984] D. Harel. Dynamic logic. In *Handbook of Philosophical Logic*, vol, II, pp.497–640. Reidel, 1984.

[Horrocks and Sattler, 1999] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *JLC*, 9:385–410, 1999.

[Horrocks and Tobies, 2000] I. Horrocks and S. Tobies. Reasoning with axioms: Theory and practice. In *Proc. of KR 2000*, pp.285–296. 2000.

[Horrocks *et al.*, 2000] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. *Logic J. of the IGPL*, 8:239–263, 2000.

[Hustadt and Schmidt, 2001] U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. *JAR*, 2001. To appear.

[Ladner, 1977] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM JoC*, 6:467–480, 1977.

[Lutz and Sattler, 2000] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logic. In *Proc. of AiML 2000*.

[Massacci, 1998] F. Massacci. Tableaux methods for formal verification in multi-agent distributed systems. *JLC*, 8:373–400, 1998.

[Massacci, 2000] F. Massacci. Single step tableaux for modal logics: methodology, computations, algorithms. *JAR*, 24:319–364, 2000.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics. In *Proc. of IJCAI'91*, pp.466–471. 1991.

[Schmidt-Schauß and Smolka, 1991] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *AIJ*, 48:1–26, 1991.

[Tobies, 2001] S. Tobies. PSPACE reasoning for graded modal logic. *JLC*, 2001, To appear.

[Vardi and Wolper, 1986] M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *JCSS*, 32:183–221, 1986.