On Expressive Description Logics with Composition of Roles in Number Restrictions

Fabio Grandi

CSITE-CNR and DEIS
Alma Mater Studiorum – Università di Bologna
Viale Risorgimento 2, I-40136 Bologna, Italy
Email: fgrandi@deis.unibo.it

Abstract. Description Logics are knowledge representation formalisms which have been used in a wide range of application domains. Owing to their appealing expressiveness, we consider in this paper extensions of the well-known concept language $\mathcal{ALC}$ allowing for *number restrictions* on complex role expressions. These have been first introduced by Baader and Sattler as $\mathcal{ALCN}(M)$ languages, with the adoption of role constructors $M \subseteq \{\circ, ^-, \sqcup, \sqcap\}$.
In particular, as far as languages equipped with role composition are concerned, they showed in 1999 that, although $\mathcal{ALCN}(\circ)$ is decidable, the addition of other operators may easily lead to undecidability: in fact, $\mathcal{ALCN}(\circ, \sqcap)$ and $\mathcal{ALCN}(\circ, ^-, \sqcup)$ were proved undecidable.
In this work, we further investigate the computational properties of the $\mathcal{ALCN}$ family, aiming at narrowing the decidability gap left open by Baader and Sattler's results. In particular, we will show that $\mathcal{ALCN}(\circ)$ extended with inverse roles both in number and in value restrictions becomes undecidable, whereas it can be safely extended with qualified number restrictions without losing decidability of reasoning.

1 Introduction

Description Logics are a family of first-order formalisms that have been found useful for domain knowledge representation in several application fields, from database design —including conceptual, object-oriented, temporal, multimedia and semistructured data modeling— to software engineering and ontology management (e.g. [1, 7, 8, 11, 12, 15, 16, 22] and [2, Part 3]). Different Description Logics provide for *constructors* which can be used to combine atomic *concepts* (unary predicates) and *roles* (binary predicates) to build *complex* concepts and roles. The available constructors characterize the description language as to *expressiveness* and *computational behaviour* (decidability and complexity) of the basic reasoning tasks like concept satisfiability and subsumption.

Well-known Description Logics are $\mathcal{ALC}$ [25], which allows for Boolean propositional constructors on concepts and (universal and existential) value restrictions on atomic roles, and its extension $\mathcal{ALCN}$ [14, 21] introducing (non-qualified) *number restrictions* on atomic roles. Basic inference problems for both these

$$
\begin{array}{rllrl}
C, D \rightarrow & A \mid & & A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} & \text{atomic concept} \\
& \top \mid & & \top^{\mathcal{I}} & = \Delta^{\mathcal{I}} \\
& \bot \mid & & \bot^{\mathcal{I}} & = \emptyset \\
& \neg C \mid & & (\neg C)^{\mathcal{I}} & = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
& C \sqcap D \mid & & (C \sqcap D)^{\mathcal{I}} & = C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
& C \sqcup D \mid & & (C \sqcup D)^{\mathcal{I}} & = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
& \forall R.C \mid & & (\forall R.C)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \forall j.\ R^{\mathcal{I}}(i,j) \Rightarrow C^{\mathcal{I}}(j)\} \\
& \exists R.C \mid & & (\exists R.C)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \exists j.\ R^{\mathcal{I}}(i,j) \wedge C^{\mathcal{I}}(j)\} \\
& \exists^{\geq n} R \mid & & (\exists^{\geq n} R)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j)\} \geq n\} \\
& \exists^{\leq n} R \mid & & (\exists^{\leq n} R)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j)\} \leq n\} \\
* & \exists^{\geq n} R.C \mid & & (\exists^{\geq n} R.C)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j) \wedge C^{\mathcal{I}}(j)\} \geq n\} \\
* & \exists^{\leq n} R.C \mid & & (\exists^{\leq n} R.C)^{\mathcal{I}} & = \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j) \wedge C^{\mathcal{I}}(j)\} \leq n\} \\
R, S \rightarrow & P \mid & & P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} & \text{atomic role} \\
* & R^{-} \mid & & (R^{-})^{\mathcal{I}} & = \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(j,i)\} \\
* & R \circ S \mid & & (R \circ S)^{\mathcal{I}} & = \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists k.\ R^{\mathcal{I}}(i,k) \wedge S^{\mathcal{I}}(k,j)\}
\end{array}
$$

Fig. 1. Syntax and model-theoretic semantics of $\mathcal{ALCN}$ and its extensions (marked with $*$) considered in this paper.

Description Logics are PSPACE-complete [14, 21]. However, in order to better fulfil requirements of real-world application domains, more expressive extensions of the basic concept languages have been investigated. One direction along which useful extensions have been sought is the introduction of *complex* roles under number restrictions. In fact, considering role composition ($\circ$), inversion ($^{-}$), union ($\sqcup$) and intersection ($\sqcap$), expressive extensions of $\mathcal{ALCN}$ can be defined as $\mathcal{ALCN}(M)$ with the adoption of role constructors $M \subseteq \{\circ, ^{-}, \sqcup, \sqcap\}$ [3]. By allowing (different kinds of) complex roles also in value restrictions, other families of logics can also be defined: for example $\mathcal{ALC}_{+}\mathcal{N}$ (or $\mathcal{ALC}_{\mathrm{reg}}\mathcal{N}$) allows the transitive closure of atomic roles (or regular roles) under value restrictions [3, 10]. Also logics $\mathcal{ALC}\bar{\mathcal{N}}(M)$, allowing for the same types of role constructors both in value and in number restrictions, can be considered [18]. Further extensions involve the introduction of *qualified* number restrictions [20] on complex roles, giving rise to $\mathcal{ALCQ}(M)$ logics. Since qualified number restrictions also allow us to express value restrictions, we have the inclusions $\mathcal{ALCN}(M) \subseteq \mathcal{ALC}\bar{\mathcal{N}}(M) \subseteq \mathcal{ALCQ}(M)$ as far as expressiveness (and complexity) are concerned. Therefore, for instance, undecidability of $\mathcal{ALCN}(M)$ directly extends to $\mathcal{ALC}\bar{\mathcal{N}}(M)$ and $\mathcal{ALCQ}(M)$, whereas decidability of $\mathcal{ALCQ}(M)$ implies decidability of $\mathcal{ALCN}(M)$ and $\mathcal{ALC}\bar{\mathcal{N}}(M)$.

Our present investigation is aimed at improving the (un)decidability results presented by Baader and Sattler in [3] for $\mathcal{ALCN}$ extensions including composition of roles ($\circ$). In particular, they proved that concept satisfiability in $\mathcal{ALCN}(\circ, \sqcap)$ and $\mathcal{ALCN}(\circ, ^{-}, \sqcup)$ is undecidable via reduction of a domino problem, and provided a sound and complete Tableau algorithm for deciding satisfiability of $\mathcal{ALCN}(\circ)$-concepts. They also observed that $\mathcal{ALCN}(^{-}, \sqcup, \sqcap)$ is decidable since $\mathcal{ALCN}(^{-}, \sqcup, \sqcap)$-concepts can easily be translated into a formula in $\mathcal{C}^2$ [6], that is the two-variable FOL fragment with counting quantifiers, which has proved to be decidable [17]. In fact, satisfiability of $\mathcal{C}^2$ formulae can be decided in NExpTime [23] if unary coding of numbers is used (which is a common assumption in the field of Description Logics; if binary coding is adopted we have a 2-NExpTime upper bound). We can further observe that a similar translation is still possible when *qualified* number restrictions are considered and, thus, also $\mathcal{ALCQ}(^-, \sqcup, \sqcap)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^-, \sqcup, \sqcap)$ are decidable.

In this paper, we consider extensions of $\mathcal{ALCN}(\circ)$ with role inversion ($\mathcal{I}$) or qualified number restrictions ($\mathcal{Q}$), whose decidability status, to the best of our knowledge, is still unknown. In particular, we will show in Sec. 2 (via reduction of a domino problem) undecidability of $\mathcal{ALCN}(\circ)$ extended with inverse roles both in value and in number restrictions (which we can call $\mathcal{ALCN}(\circ)\mathcal{I}$, but we also show in Sec. 2 that it is a syntactic variant of $\mathcal{ALC}\bar{\mathcal{N}}(\circ, ^-))$ is undecidable. This result implies undecidability of $\mathcal{ALCQ}(\circ, ^-)$, whereas decidability of "pure" $\mathcal{ALCN}(\circ, ^-)$ remains an open question. On the other hand, we will show how the decidability results of [3] lift up to $\mathcal{ALCQ}(\circ)$. In particular, we will show in Sec. 3 that $\mathcal{ALCQ}(\circ)$-concept satisfiability is decidable and provide an effective decision procedure in the form of a tableau-based algorithm, which extends the $\mathcal{ALCN}(\circ)$ Tableau proposed by Baader and Sattler [3].

Due to space limitations, the proofs have not been included but can be found in an extended version of this paper which is available online [19].

Preliminaries on Description Logics

The expressiveness of a Description Logic (DL) is based on the definition of complex concepts and roles, which can be built with the help of available constructors, starting from a set of (atomic) concept names NC and a set of (atomic) role names NR. A DL system, enabling concept descriptions to be interrelated, allows the derivation of implicit knowledge from explicitly represented knowledge by means of inference services. For a full account of Description Logics, the reader is referred, for example, to [2, 10]. An introductory overview of DLs as foundations for class-based knowledge representation is also [9].

In the DL $\mathcal{ALC}$ [25], concept descriptions are formed using the constructors negation, conjunction and disjunction, value (and existential) restrictions. The DL $\mathcal{ALCN}$ [14, 21] additionally allows for unqualified (at-least and at-most) number restrictions on atomic roles. The syntax rules at the left hand side of Fig. 1 inductively define valid concept and role expressions for $\mathcal{ALCN}$ and its extensions considered in this paper. As far as semantics is concerned, concepts are interpreted as sets of individuals and roles as sets of pairs of individuals. Formally, an *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set of individuals (the *domain* of $\mathcal{I}$) and $\mathcal{I}$ is a function (the *interpretation function*) which maps each concept to a subset of $\Delta^{\mathcal{I}}$ and each role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, such that the equations at the right hand side of Fig. 1 are satisfied. One of the most important inference services of DL systems used in knowledge-representation and conceptual modeling applications is computing the subsumption hierarchy of a given finite set of concept descriptions.

Definition 1. *The concept description C is* satisfiable *iff there exist an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$; in this case, we say that $\mathcal{I}$ is a model for C. The concept description D* subsumes *the concept description C (written $C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$; concept descriptions C and D are* equivalent *iff $C \sqsubseteq D$ and $D \sqsubseteq C$.*

Since $\mathcal{ALC}$ is propositionally complete, subsumption can be reduced to concept satisfiability and *vice versa*: $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable and C is satisfiable iff not $C \sqsubseteq A \sqcap \neg A$, where A is an arbitrary concept name.

In $\mathcal{ALCN}$, number restrictions are used to restrict the cardinality of the set of fillers of roles (role successors). For instance, the concept description:

$$\exists^{\leq 3}\texttt{child} \sqcap \forall\texttt{child.Female}$$

defines individuals who have at most three daughters and no sons. Moreover, $\mathcal{ALCN}(\circ)$ [3] allows counting successors of role chains, which can be used to express interesting cardinality constraints on the interrelationships some individuals hold with other objects of the domain. For example, the $\mathcal{ALCN}(\circ)$-concept:

$$\texttt{Man} \sqcap \exists^{\geq 50}(\texttt{friend} \circ \texttt{tel\_number})$$

allows us to define men for which the count of different telephone numbers of their friends amounts at least to fifty. Notice that such description does not impose further constraints (disregarding obvious ones) either on the number of friends one may have, or on the number of telephone numbers each friend may have (e.g. some friends might have no telephone at all), or even on the fact that some numbers may be shared by more than one friends (e.g. if husband and wife). It only gives, for example, a constraint on the minimum size of a phonebook such men need. Number restrictions on composition of roles can be used, for instance, to express cardinality constraints on property paths in conceptual modeling of object-oriented, nested relational or semistructured data.

The additional role constructs we consider in this paper further improve the expressiveness of the resulting DLs and, thus, make them very appealing from an application viewpoint. For instance, we may use the $\mathcal{ALCN}(\circ,^-)$-concept:

$$\texttt{Person} \sqcap \exists\texttt{child}^- \sqcap \exists^{\leq 1}(\texttt{child}^- \circ \texttt{child})$$

to define persons who are a only child, or the $\mathcal{ALCQ}(\circ)$ concept:

$$\texttt{Woman} \sqcap \exists^{\geq 3}(\texttt{husband} \circ \texttt{brother}).\texttt{Lawyer}$$

to describe women having at least three lawyers as brother-in-law.

2 Undecidability of $\mathcal{ALCN}(\circ)$ with Inverse Roles

We consider in this Section the extension of $\mathcal{ALCN}(\circ)$ by inverse roles ($\mathcal{I}$). Notice that allowing the use of role inversion both in number and in value restrictions, we obtain a DL $\mathcal{ALCN}(\circ)\mathcal{I}$ which is a syntactic variant of $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$. Obviously, $\mathcal{ALCN}(\circ)\mathcal{I}$ concept descriptions are also $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$ concept descriptions. Conversely, by recursively applying rules $(R \circ S)^- = S^- \circ R^-$ (pushing inverses inwards and eliminating parentheses) and $(R^-)^- = R$, we can put any $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$ complex role expression in the form $\bar{R}_1 \circ \bar{R}_2 \circ \cdots \circ \bar{R}_n$, where each $\bar{R}_i$ is either an atomic role or the inverse of an atomic role ($\bar{R}_i \in \{R_i, R_i^-\}$). Then we can get rid of role composition in value restrictions thanks to the following equivalences:

$$\exists(\bar{R}_1 \circ \bar{R}_2 \circ \cdots \circ \bar{R}_n).C \equiv \exists \bar{R}_1.\exists \bar{R}_2.\cdots\exists \bar{R}_n.C$$
$$\forall(\bar{R}_1 \circ \bar{R}_2 \circ \cdots \circ \bar{R}_n).C \equiv \forall \bar{R}_1.\forall \bar{R}_2.\cdots\forall \bar{R}_n.C$$

These rules give a translation procedure of concept descriptions from $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$ into $\mathcal{ALCN}(\circ)\mathcal{I}$.

To show undecidability of $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$, borrowing the proof procedure from [3], we will use a reduction of the well-known undecidable domino problem [5].

Definition 2. *A tiling system $\mathcal{D} = (D, H, V)$ is given by a non-empty set $D = \{D_1, \ldots, D_m\}$ of domino types, and by horizontal and vertical matching pairs $H \subseteq D \times D$, $V \subseteq D \times D$. The (unrestricted) domino problem asks for a compatible tiling of the plane, i.e. a mapping $t : \mathbb{Z} \times \mathbb{Z} \to D$ such that, for all $m, n \in \mathbb{Z}$,*

$$\langle\, t(m,n), t(m+1,n) \,\rangle \in H \text{ and } \langle\, t(m,n), t(m,n+1) \,\rangle \in V$$

We will show reducibility of the domino problem to *concept satisfiability* in $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^-)$. In particular, we show how a given tiling system $\mathcal{D}$ can be translated into a concept $E_{\mathcal{D}}$ which is satisfiable iff $\mathcal{D}$ allows for a compatible tiling. Following the same lines of undecidability proofs in [3], such translation can be split into three subtasks which can be described as follows:

Grid Specification It must be possible to represent a "square" of $\mathbb{Z} \times \mathbb{Z}$, which consists of points (m,n), $(m+1,n)$, $(m,n+1)$ and $(m+1,n+1)$, in order to yield a complete covering of the plane via a repeating regular grid structure. The idea is to introduce concepts to represent the grid points and roles to represent the x- and y-successor relations between points.

Local Compatibility It must be possible to express that a tiling is locally compatible, that is that the x-successor and the y-successor of a point have an admissible domino type. The idea is to associate each domino type D_i with an atomic concept D_i, and to express the horizontal and vertical matching conditions via value restrictions.

Total Reachability It must be possible to impose the above local conditions on all points in $\mathbb{Z} \times \mathbb{Z}$. This can be achieved by constructing a "universal" role and a "start" individual such that every grid point can be reached from the start individual. The local compatibility conditions can then be globally imposed via value restrictions.

The grid structure that we will use to tile the plane is shown in Fig. 2. In particular, in addition to grid points, we also consider "centers" of grid squares,

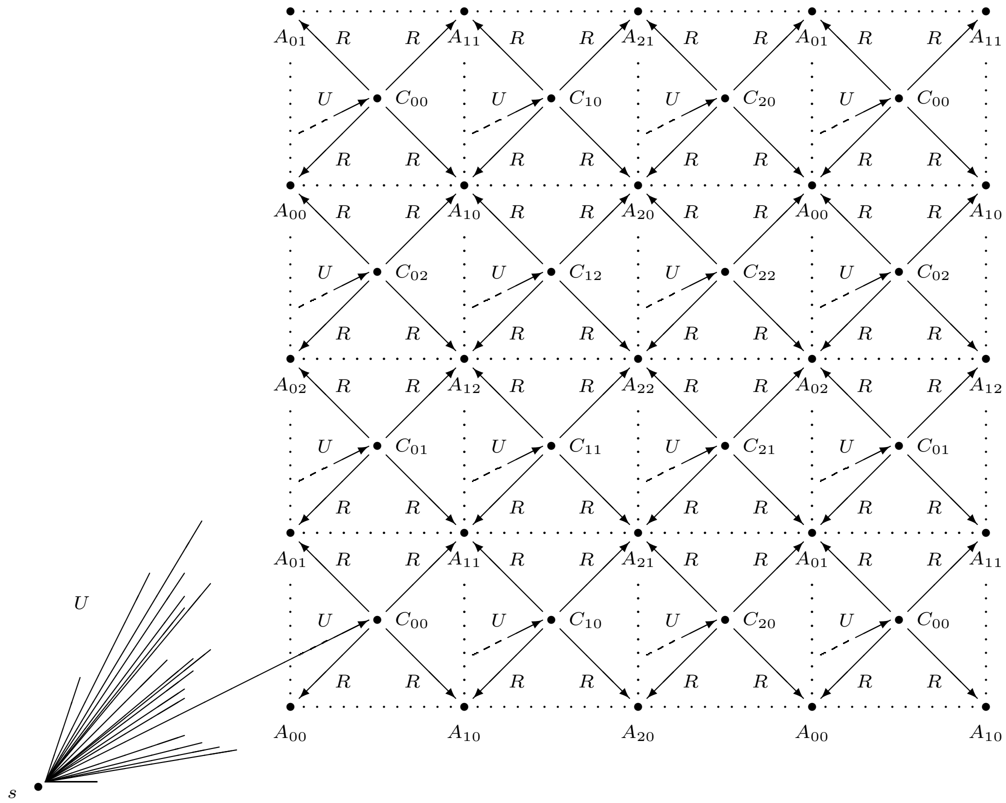

Fig. 2. The grid structure used in the $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^{-})$ undecidability proof.

which are connected to grid square vertices by means of a role named R. All grid cell centers are instances of the C concept, whereas grid points are instances of the A concept. We introduce nine different (disjoint) types of grid centers via the concepts C_{ij} $(0 \leq i,j \leq 2)$ and nine different types of (disjoint) grid points via the concepts A_{ij} $(0 \leq i,j \leq 2)$, as follows:

$$C := \bigsqcup_{0\leq i,j\leq 2} \left(C_{ij} \sqcap (\bigsqcap_{\substack{0\leq k,\ell\leq 2\\(i,j)\neq(k,\ell)}} \neg C_{k\ell}) \right)$$

$$A := \bigsqcup_{0\leq i,j\leq 2} \left(A_{ij} \sqcap (\bigsqcap_{\substack{0\leq k,\ell\leq 2\\(i,j)\neq(k,\ell)}} \neg A_{k\ell}) \right) \sqcap \neg C$$

Grid Specification can then be accomplished by means of the $C_{\boxplus}$ and $A_{\boxplus}$ concepts which follow:

$$
\begin{aligned}
C_{\boxplus} &:= C \sqcap \exists^{\leq 4} R \sqcap \forall R.A_{\boxplus} \sqcap \exists^{\leq 9} R \circ R^{-} \sqcap \\
&\qquad \bigsqcap_{0 \leq i,j \leq 2} \big(C_{ij} \Rightarrow (\exists R.A_{ij} \sqcap \exists R.A_{i\oplus 1,j} \sqcap \exists R.A_{i,j\oplus 1} \sqcap \exists R.A_{i\oplus 1,j\oplus 1}) \big) \\
A_{\boxplus} &:= A \sqcap \bigsqcap_{0 \leq i,j \leq 2} \big(A_{ij} \Rightarrow (\exists R^{-}.C_{ij} \sqcap \exists R^{-}.C_{i\oplus 2,j} \sqcap \exists R^{-}.C_{i,j\oplus 2} \sqcap \exists R^{-}.C_{i\oplus 2,j\oplus 2}) \big)
\end{aligned}
$$

where $a \oplus b = (a+b) \bmod 3$ and $A \Rightarrow B$ is a shorthand for $\neg A \sqcup B$.

Some relevant constraints that are imposed by these concept descriptions on their models are studied in the Lemma which follows.

Lemma 1. *Let c be an instance of $C_{\boxplus}$ and a an instance of $A_{\boxplus}$. Then:*

1. *c has at most one R-successor in each of the nine $A_{k\ell}$ concept extensions.*
2. *c has exactly one $(R \circ R^{-})$-successor in each of the nine $C_{k\ell}$ concept extensions.*
3. *a has only (R^{-})-successors which are instances of $C_{\boxplus}$.*
4. *a has at most one (R^{-})-successor in each of the nine $C_{k\ell}$ concept extensions.*
5. *a has exactly one $(R^{-} \circ R)$-successor in each of the nine $A_{k\ell}$ concept extensions.*

Hence, we will interpret instances of $C_{\boxplus}$ as grid centers and instances of $A_{\boxplus}$ as grid points. In particular, nine different types of grid cells can be defined according to the type of their center: an (i,j)-type grid cell has a C_{ij}-type center, while its lower left, lower right, upper left and upper right vertices can be defined, respectively, as the instances of the A_{ij}, $A_{i\oplus 1,j}$, $A_{i,j\oplus 1}$ and $A_{i\oplus 1,j\oplus 1}$ concepts which are connected to the center via R (according to the $C_{\boxplus}$ definition). Therefore, the x- and y-successor relations on the grid can be defined by means of the $(R^{-} \circ R)$-paths connecting an A_{ij}-type grid point with an $A_{i\oplus 1,j}$-type and an $A_{i,j\oplus 1}$-type grid points, respectively. Such successors always exist and are uniquely defined, owing to Lemma 1. In a similar way, Lemma 1 also allows us to uniquely define the x- and y-predecessors relations on the grid, by means of the $(R^{-} \circ R)$-paths connecting an A_{ij}-type grid point with an $A_{i\oplus 2,j}$-type and an $A_{i,j\oplus 2}$-type grid points, respectively (cf. $(a+2) \mod 3 = (a-1) \mod 3$).

Furthermore, an easy consequence of Lemma 1 is the following:

Proposition 1 (Grid Closure). *For each grid point, the $(x \circ y)$- and $(y \circ x)$-successors are uniquely defined and coincide.*

Local Compatibility is easily achieved by enforcing grid centers to be instances of a $C_{\mathcal{D}}$ concept defined as follows:

$$
\begin{aligned}
C_{\mathcal{D}} := \forall R. \Bigg(\bigsqcup_{1 \leq k \leq m} \big(D_k \sqcap (\bigsqcap_{\substack{1 \leq \ell \leq m \\ k \neq \ell}} \neg D_\ell) \big) \Bigg) \sqcap \bigsqcap_{0 \leq i,j \leq 2} \Bigg(C_{ij} \Rightarrow \bigsqcap_{1 \leq k \leq m} \Bigg(\exists R.(A_{ij} \sqcap D_k) \\
\Rightarrow \big(\exists R.(A_{i\oplus 1,j} \sqcap (\bigsqcup_{(D_k,D_\ell) \in H} D_\ell)) \sqcap \exists R.(A_{i,j\oplus 1} \sqcap (\bigsqcup_{(D_k,D_\ell) \in V} D_\ell)) \big) \Bigg) \Bigg)
\end{aligned}
$$

Each domino type D_k is associated to an atomic concept with the same name. The value restriction in the first conjunct of $C_{\mathcal{D}}$ forces grid points to have a domino type. The second conjunct uses the definition of the x- and y-successors for the bottom left vertex of an (i,j)-type cell to enforce horizontal and vertical matching conditions via value restrictions.

Total Reachability will be achieved by constructing a "start" individual (s) and two "universal" roles: the former (U) which connects s to every grid center and the latter ($U \circ R$) which connects s to every grid point (see Fig. 2). The Lemma which follows justifies the correctness of our construction.

Lemma 2. *Let s be an instance of*

$$D \ := \ \exists U \circ R \sqcap \exists^{\leq 1}(U \circ R) \circ (U \circ R)^{-} \sqcap \forall U.\forall R.\forall R^{-}.\exists U^{-}$$

in a given interpretation $\mathcal{I}$. Then:

1. *Any $(U \circ R)$-successor of s in $\mathcal{I}$ (D ensures that there is at least one) has s as its* unique *$(U \circ R)$-predecessor.*
2. *Any U-successor of s in $\mathcal{I}$ has s as its* unique *U-predecessor.*
3. *Any $(U \circ R \circ R^{-})$-successor of s in $\mathcal{I}$ (D ensures that there is at least one) is a U-successor of s in $\mathcal{I}$ and has s as its* unique *U-predecessor.*

As a consequence, we have the following result:

Proposition 2 (Plane Covering and Compatible Tiling). *Let s be an instance of*

$$E_{\mathcal{D}} := \exists U \circ R \sqcap \exists^{\leq 1}(U \circ R) \circ (U \circ R)^{-} \sqcap \forall U.\forall R.\forall R^{-}.\exists U^{-} \sqcap \forall U.(C_{\boxplus} \sqcap C_{\mathcal{D}})$$

in a given interpretation $\mathcal{I}$. Then, for the grid[1] *that tiles the plane $\mathbb{Z} \times \mathbb{Z}$, any grid center can be reached from s via U, any grid point can be reached from s via $U \circ R$ and local tiling conditions are imposed on all grid points (yielding a compatible tiling of the plane).*

Thanks to Proposition 2, it is easy to see that a tiling system $\mathcal{D}$ has a compatible tiling iff concept $E_{\mathcal{D}}$ is satisfiable (i.e. there is an interpretation $\mathcal{I}$ such that $(E_{\mathcal{D}})^{\mathcal{I}} \neq \emptyset$).

Theorem 1. *Satisfiability (and, thus, subsumption) of concepts is undecidable for $\mathcal{ALC}\bar{\mathcal{N}}(\circ,^{-})$ (and $\mathcal{ALCQ}(\circ,^{-})$).*

3 Decidability of $\mathcal{ALCQ}(\circ)$

We will show in this Section how an effective decision procedure for $\mathcal{ALCQ}(\circ)$-concept satisfiability can be provided as a tableau-based algorithm [4]. To this

[1] In order to prevent s, as in Fig. 2, from being a grid center or grid point, further conjuncts can be added to $E_{\mathcal{D}}$ (e.g. $\neg(\exists R^{-} \sqcup \exists U^{-})$ or $\neg(A \sqcup C)$).

end, we consider $\mathcal{ALCQ}(\circ)$-concept descriptions in Negation Normal Form (NNF [25]), where the negation sign is allowed to appear before atomic concepts only. In fact, $\mathcal{ALCQ}(\circ)$-concept descriptions can be transformed into NNF in linear time via application of the same rules which can be used for $\mathcal{ALCQ}$ (pushing negations inwards):

$$\begin{array}{rclcrcl} \neg\exists^{\leq n}R.C & = & \exists^{\geq n+1}R.C & \qquad & \neg\exists^{\geq n}R.C & = & \exists^{\leq n-1}R.C \quad (\bot \text{ if } n=0) \\ \neg\exists R.C & = & \forall R.\neg C & & \neg\forall R.C & = & \exists R.\neg C \end{array}$$

in addition to the absorption rule for double negations and De Morgan's laws for $\sqcap$ and $\sqcup$. Obviously, unqualified number restrictions are treated as particular cases of qualified restrictions (with $C = \top$). We can further make use of the rules:

$$\exists R.C \;=\; \exists^{\geq 1}R.C \qquad \forall R.C \;=\; \exists^{\leq 0}R.\neg C$$

to get rid of (existential and) value restrictions. We define the concept descriptions obtained in this way as in $\mathrm{NNF}^{\bowtie}$ and denote the $\mathrm{NNF}^{\bowtie}$ of the $\mathcal{ALCQ}(\circ)$-concept description $\neg C$ as $\sim C$. We will use the symbol $\bowtie$ in number restrictions $\exists^{\bowtie n}R.C$ as a placeholder for either $\geq$ or $\leq$.

The Tableau algorithm we are going to introduce manipulates, as basic data structures, ABox assertions involving domain individuals. In fact, our algorithm is a simple extension of the tableau-based algorithm to decide $\mathcal{ALCN}(\circ)$-concept satisfiability presented by Baader and Sattler in [3] (also the proofs given in [19] are very similar to the proofs provided in [3] for the $\mathcal{ALCN}(\circ)$ Tableau). The extension is based on the modification of the transformation rules for number restrictions ($\geq$- and $\leq$-rules) to take into account the "qualifying" conditions and on the introduction of a so-called *choose* rule (called ©-rule here), which ensures that all "relevant" concepts that are implicitly satisfied by an individual are made explicit in the ABox. Basically, the proposed extension is similar to the one which extends the tableau-based $\mathcal{ALCN}$ satisfiability algorithm [14, 21] to an $\mathcal{ALCQ}$ satisfiability algorithm [4, 20].

Definition 3. *Let* NI *be a set of individual names. An* ABox $\mathcal{A}$ *is a finite set of assertions of the form* $C(a)$ –concept assertion– *or* $R(a, b)$ –role assertion– *where* C *is a concept description,* R *a role name, and* a, b *are individual names. An interpretation* $\mathcal{I}$*, which additionally assigns elements* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ *to individual names* a*, is a* model *of an ABox* $\mathcal{A}$ *iff* $a^{\mathcal{I}} \in C^{\mathcal{I}}$ *(resp.* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$*) for all assertions* $C(a)$ *(resp.* $R(a, b)$*) in* $\mathcal{A}$*. The ABox* $\mathcal{A}$ *is consistent iff it has a model. The individual* a *is an instance of the description* C *w.r.t.* $\mathcal{A}$ *iff* $a^{\mathcal{I}} \in C^{\mathcal{I}}$ *holds for all models* $\mathcal{I}$ *of* $\mathcal{A}$*. We also consider in a ABox* inequality assertions *of the form* $a \neq b$*, with the obvious semantics that an interpretation* $\mathcal{I}$ *satisfies* $a \neq b$*, iff* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$*. Inequality assertions are assumed to be symmetric, that is saying that* $a \neq b \in \mathcal{A}$ *is the same as saying* $b \neq a \in \mathcal{A}$*.*

Sometimes in the DL field, a *unique name assumption* is made in works concerning reasoning with individuals, that is the mapping $\pi : \mathsf{NI} \rightarrow \Delta^{\mathcal{I}}$ from individual names to domain elements is required to be injective. We dispense from this requirement as it has no effect for the $\mathcal{ALC}$ extensions studied here and the explicitly introduced inequality assertions can be used anyway to enforce the uniqueness of names if necessary.

Definition 4. *The individual y is a $(R_1 \circ R_2 \circ \cdots \circ R_m)$*-successor *of x in $\mathcal{A}$ iff $\exists y_2 y_3 \ldots y_m$ variables in $\mathcal{A}$ such that $\{R_k(y_k, y_{k+1}) \mid 2 \leq k \leq m-1\} \cup \{R_1(x, y_2), R_m(y_m, y)\} \subseteq \mathcal{A}$.*

Definition 5. *An ABox $\mathcal{A}$ contains a* clash *iff, for an individual name $x \in \mathsf{NI}$, one of the two situations below occurs:*

- *$\{A(x), \neg A(x)\} \subseteq \mathcal{A}$, for a concept name $A \in \mathsf{NC}$;*
- *$(\exists^{\leq n} R_1 \circ \cdots \circ R_m.C)(x) \in \mathcal{A}$ and x has p $(R_1 \circ \cdots \circ R_m)$-successors $y_1, \ldots, y_p$ with $p > n$ such that $\{C(y_i) \mid 1 \leq i \leq p\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq p\} \subseteq \mathcal{A}$, for role names $\{R_1, \ldots, R_m\} \subseteq \mathsf{NR}$, a concept description C and an integer $n \geq 0$.*

To test the satisfiability of an $\mathcal{ALCQ}(\circ)$ concept C_0 in NNF$^{\bowtie}$, the proposed $\mathcal{ALCQ}(\circ)$-algorithm works as follows. Starting from the initial ABox $\{C_0(x_0)\}$, it applies the *completion rules* in Fig. 3, which modify the ABox. It stops when no rule is applicable (when a clash is generated, the algorithm does not immediately stops but it always generate a complete ABox). An ABox $\mathcal{A}$ is called *complete* iff none of the completion rules is any longer applicable. The algorithm answers "C is satisfiable" iff a complete and clash-free ABox has been generated. The $\mathcal{ALCQ}(\circ)$-algorithm is non-deterministic, due to the $\sqcup$-, $\leq$- and ©-rules (for instance, the $\sqcup$-rule non-deterministically chooses which disjunct to add for a disjunctive concept).

Lemma 3. *Let C_0 be an $\mathcal{ALCQ}(\circ)$-concept in NNF$^{\bowtie}$, and let $\mathcal{A}$ be an ABox obtained by applying the completion rules to $\{C_0(x_0)\}$. Then:*

1. *For each completion rule $\mathcal{R}$ that can be applied to $\mathcal{A}$ and for each interpretation $\mathcal{I}$, the following equivalence holds: $\mathcal{I}$ is a model of $\mathcal{A}$ iff $\mathcal{I}$ is a model of the ABox $\mathcal{A}'$ obtained by applying $\mathcal{R}$.*
2. *If $\mathcal{A}$ is a complete and clash-free ABox, then $\mathcal{A}$ has a model.*
3. *If $\mathcal{A}$ is complete but contains a clash, then $\mathcal{A}$ does not have a model.*
4. *The completion algorithm terminates when applied to $\{C_0(x_0)\}$.*

As a matter of fact, termination (4) yields that after finitely many steps we obtain a complete ABox. If C_0 is satisfiable, then $\{C_0(x_0)\}$ is also satisfiable and, thus, at least one of the complete ABoxes that the algorithm can generate is satisfiable by (1). Hence, such an ABox must be clash-free by (3). Conversely, if the application of the algorithm produces a complete and clash-free ABox $\mathcal{A}$, then it is satisfiable by (2) and, owing to (1), this implies that $\{C_0(x_0)\}$ is satisfiable. Consequently, the algorithm is a decision procedure for satisfiability of $\mathcal{ALCQ}(\circ)$-concepts.

Theorem 2. *Concept satisfiability (and subsumption) for $\mathcal{ALCQ}(\circ)$ is decidable, and the Tableau algorithm based on the completion rules in Fig. 3 is an effective decision procedure.*

$\sqcap$-rule: **if** 1. $(C_1 \sqcap C_2)(x) \in \mathcal{A}$ and
2. $\{C_1(x), C_2(x)\} \not\subseteq \mathcal{A}$
then $\mathcal{A}' := \mathcal{A} \cup \{C_1(x), C_2(x)\}$

$\sqcup$-rule: **if** 1. $(C_1 \sqcup C_2)(x) \in \mathcal{A}$ and
2. $\{C_1(x), C_2(x)\} \cap \mathcal{A} = \emptyset$
then $\mathcal{A}' := \mathcal{A} \cup \{D(x)\}$ for some $D \in \{C_1, C_2\}$

$\geq$ -rule: **if** 1. $(\exists^{\geq n} R_1 \circ \cdots \circ R_m.C)(x) \in \mathcal{A}$ and
2. x has exactly p $(R_1 \circ \cdots \circ R_m)$-successors $y_1, \ldots, y_p$ with $p < n$ such that $\{C(y_i) \mid 1 \leq i \leq p\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq p\} \subseteq \mathcal{A}$
then $\mathcal{A}' := \mathcal{A} \cup \{R_1(x, z_{i2}), R_2(z_{i2}, z_{i3}), \ldots, R_m(z_{im}, z_i), C(z_i) \mid 1 \leq i \leq n-p\}$
$\cup\{z_i \neq z_j \mid 1 \leq i < j \leq n-p\} \cup \{y_i \neq z_j \mid 1 \leq i \leq p, 1 \leq j \leq n-p\}$
where z_{ik}, z_i (for $1 \leq i \leq n-p, 2 \leq k \leq m$) are $m(n-p)$ fresh variables

$\leq$ -rule: **if** 1. $(\exists^{\leq n} R_1 \circ \cdots \circ R_m.C)(x) \in \mathcal{A}$ and
2. x has more than n $(R_1 \circ \cdots \circ R_m)$-successors $y_1, \ldots, y_p$ such that $\{C(y_i) \mid 1 \leq i \leq p\} \subseteq \mathcal{A}$ and $\{y_i \neq y_j\} \cap \mathcal{A} = \emptyset$ for some i, j $(1 \leq i < j \leq p)$,
then for some pair y_i, y_j $(1 \leq i < j \leq p)$ such that $\{y_i \neq y_j\} \cap \mathcal{A} = \emptyset$
$\mathcal{A}' := [y_i/y_j]\mathcal{A}$ (i.e. $\mathcal{A}'$ is obtained by replacing each occurrence of y_i by y_j)

©-rule: **if** 1. $(\exists^{\bowtie n} R_1 \circ \cdots \circ R_m.C)(x) \in \mathcal{A}$ and
2. y is an $(R_1 \circ \cdots \circ R_m)$-successor of x such that $\{C(y), \sim C(y)\} \cap \mathcal{A} = \emptyset$
then $\mathcal{A}' := \mathcal{A} \cup \{D(y)\}$ for some $D \in \{C, \sim C\}$

Fig. 3. The Completion Rules for $\mathcal{ALCQ}(\circ)$

3.1 Complexity issues

The tableau-based satisfiability algorithm proposed above for $\mathcal{ALCQ}(\circ)$ may require exponential time and space. The strategies leading to optimized algorithms for $\mathcal{ALCN}$ and $\mathcal{ALCQ}$ [4, 26] do not seem to be applicable to $\mathcal{ALCN}(\circ)$ and $\mathcal{ALCQ}(\circ)$. As a matter of fact, such strategies rely on the fact that the underlying logics have the *tree model* property, and, for the sake of satisfiability testing, the individuality of different role-successors of a given domain object is not relevant. Only the number of such successors counts (for $\geq$- and $\leq$-rule applicability and clash testing) and, thus, a single successor at a time can be used as "representative" also for its siblings, when continuing the algorithm for its further role-successors. In such a way, only one branch of the tree model at a time can be generated and investigated by the algorithm, giving rise to a non-deterministic procedure consuming only polynomial space and, thus, to PSpace complexity (since NPSpace =PSpace, owing to Savitch's Theorem [24]). In our case, such an optimization does not seem to be possible, since $\mathcal{ALCN}(\circ)$ and $\mathcal{ALCQ}(\circ)$ do not have the tree model property, as number restrictions $\exists^{\geq p} R_1 \circ \cdots \circ R_{m-1} \sqcap \exists^{\leq q} R_1 \circ \cdots \circ R_{m-1} \circ R_m$ (with $p > q$) make some separate $(R_1 \circ \cdots \circ R_{m-1})$ role chains merge into confluent $(R_1 \circ \cdots \circ R_{m-1} \circ R_m)$ chains to respect both kinds of number restrictions. In fact, if the *level* of x is the unique length of the role chains that connect x_0 with x, the identifications of successors effected by the $\leq$-rule (say at level ℓ) may involve individuals generated by previous executions of the $\geq$-rule for different $(\exists^{\geq n} R_1 \circ \cdots \circ R_m.C)(x)$ constraints, with possibly different values of $\mathit{level}(x)$ and role chain lengths (with the proviso that $\mathit{level}(x) + 1 \leq \ell \leq \mathit{level}(x) + m$). The enforcement of mutual constraints between possibly "intersecting" role chains strictly relies on the *individuation* of single successors, and cannot be surrogated, in general, via representatives. As a result, the algorithm in Fig. 3 is a non-deterministic procedure possibly producing complete ABoxes of *exponential size* in the length of the input concept description (also if binary coding of numbers is assumed), as stated by the following Lemma.

Lemma 4. *Given a complete ABox $\mathcal{A}$ generated by the algorithm in Fig. 3, the size of $\mathcal{A}$ is exponential in the input size s, thanks to the following facts:*

1. *The number a of individuals in $\mathcal{A}$ is $O(2^{p(s)})$, where p is a polynomial function.*
2. *The number of constraints in $\mathcal{A}$ is a polynomial function of a.*

Hence, by the given algorithm, deciding satisfiability (subsumption) of $\mathcal{ALCQ}(\circ)$ concepts is in the NExpTime *(*co-NExpTime*) complexity class.*

4 Conclusions

In this paper we studied expressive Description Logics allowing for number restrictions on complex roles built with the composition operator ($\circ$), extended with inverse roles and qualified number restrictions.

In this framework, we improved the (un)decidability results by Baader and Sattler on logics of the $\mathcal{ALCN}$ family [3] by showing that $\mathcal{ALC\bar{N}}(\circ,^{-})$ is undecidable via reduction of a domino problem, whereas the introduction of qualified number restrictions in $\mathcal{ALCQ}(\circ)$ does not hinder decidability of reasoning. For $\mathcal{ALCQ}(\circ)$, a tableau-based satisfiability algorithm with a NExpTime upper bound has been proposed.

As we observed in the Introduction that known decidability results also lift up to $\mathcal{ALCQ}(^{-}, \sqcup, \sqcap)$, we shed some new light on the whole decidability scenario ranging from $\mathcal{ALCN}$ to $\mathcal{ALCQ}(\circ,^{-}, \sqcup, \sqcap)$. In this picture, a big unanswered question concerns decidability of $\mathcal{ALCN}(\circ, \sqcup)$, whereas a small gap left open concerns decidability of "pure" $\mathcal{ALCN}(\circ,^{-})$ (around the narrow borders of this gap, we proved in this work that the language with inverses in value restrictions and inverses and composition of roles under unqualified number restrictions is undecidable, whereas the language with inverses and role composition under value restrictions and inverses under qualified number restrictions is decidable, as it is a sublanguage of $\mathcal{CIQ}$ [13]). Another open question is the exact characterization of $\mathcal{ALCQ}(\circ)$ (and $\mathcal{ALCN}(\circ)$) complexity, as the NExpTime bound we derived may be far from being tight. Future work will also consider such issues.

References

1. A. Artale and E. Franconi. Temporal ER Modeling with Description Logics. In *Proc. Intl' Conf. on Conceptual Modeling (ER'99)*, pages 81–95, 1999.
2. F. Baader, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Decsription Logic Handbook: Theory, implementation and applications.* Cambridge University Press, 2002 *(to appear)*.
3. F. Baader and U. Sattler. Expressive Number Restrictions in Description Logics. *J. of Logic and Computation*, 9(3):319–350, 1999.
4. F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. *Studia Logica*, 69:5–40, 2001.
5. R. Berger. The Undecidability of the Dominoe Problems. *Mem. Amer. Mathematical Society*, 66:1–72, 1966.
6. A. Borgida. On the Relative Expressiveness of Description Logics and First Order Logics. *Artificial Intelligence*, 82:353–367, 1996.
7. A. Borgida and M. Jarke. Knowledge Representation and Reasoning in Software Engineering. *IEEE Transactions on Software Engineering*, 18(6):449–450, 1992.
8. D. Calvanese, G. De Giacomo, and M. Lenzerini. Representing and Reasoning on XML Documents: A Description Logic Approach. *J. of Logic and Computation*, 9(3):295–318, 1999.
9. D. Calvanese, G. De Giacomo, and M. Lenzerini. Description Logics: Foundations for Class-based Knowledge Representation. In *Proc. of IEEE Symposium on Logic in Computer Science (LICS'99)*, 2002.
10. D. Calvanese, G. De Giacomo, M. Lenzerini, and D. Nardi. Reasoning in Expressive Description Logics. In *Handbook of Automated Reasoning*, pages 1581–1634. Elsevier, 2001.
11. D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description Logic Framework for Information Integration. In *Proc. of Intl' Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, 1998.
12. D. Calvanese, M. Lenzerini, and D. Nardi. Description Logics for Conceptual Data Modeling. In *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publishers, 1998.
13. G. De Giacomo and M. Lenzerini. TBox and ABox Reasoning in Expressive Description Logics. In *Proc. of Intl' Conf. on the Principles of Knowledge Representation and Reasoning (KR'96)*, pages 348–353, 1996.
14. F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The Complexity of Concept Languages. *Information and Computation*, 134:1–58, 1997.
15. E. Franconi, F. Grandi, and F. Mandreoli. A Semantic Approach for Schema Evolution and Versioning in Object-Oriented Databases. In *Proc. Intl' Conf. on Deductive and Object-Oriented Databases (DOOD 2000)*, pages 1048–1062, 2000.
16. C.A. Goble and C. Haul and S. Bechhofer. Describing and Classifying Multimedia Using the Description Logic GRAIL. In *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)*, 1996.
17. E. Grädel, M. Otto, and E. Rosen. Two-variable Logic with Counting is Decidable. In *Proc. Annual IEEE Symp. on Logic in Computer Science (LICS'97)*, pages 306–317, 1997.
18. F. Grandi. On Expressive Number Restrictions in Description Logics. In *Proc. of Intl' Workshop on Description Logics (DL'01)*, 2001.
19. F. Grandi. On Expressive Description Logics with Composition of Roles in Number Restrictions. Technical Report CSITE-01-02, CSITE-CNR Bologna, 2002. (URL *ftp://ftp-db.deis.unibo.it/pub/fabio/TR/CSITE-01-02.pdf*).

20. B. Hollunder and F. Baader. Qualifying Number Restrictions in Concept Languages. In *Proc. of 2nd International Conference on Principles of Knowledge Representation and Reasoning, KR-91*, pages 335–346, 1991.
21. B. Hollunder, W. Nutt, and M. Schmidt-Schauß. Subsumption Algorithms for Concept Description Languages. In *Proc. of Europ. Conf. on Artificial Intelligence (ECAI'90)*, pages 335–346, 1990.
22. I. Horrocks and U. Sattler. Ontology Reasoning in the $\mathcal{SHOQ}(D)$ Description Logic. In *Proc. of Intl' Joint Conf. on Artificial Intelligence (IJCAI'01)*, 2001.
23. L. Pacholski, W. Szwast, and L. Tendera. Complexity of Two-variable Logic with Counting. In *Proc. Annual IEEE Symp. on Logic in Computer Science (LICS'97)*, pages 318–327, 1997.
24. W. J. Savitch. Relationship between Nondeterministic and Deterministic Tape Complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970.
25. M. Schmidt-Schauß and G. Smolka. Attributive Concept Descriptions with Complements. *Artificial Intelligence*, 48(1):1–26, 1991.
26. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.