On Expressive Number Restrictions in Description Logics

Fabio Grandi
CSITE-CNR and DEIS, University of Bologna, Italy
Email: `fgrandi@deis.unibo.it`

Abstract

We consider expressive Description Logics ($\mathcal{ALCN}$) allowing for number restrictions on complex roles built with combinations of role constructors. In particular, we are mainly interested in Logics (called $\mathcal{ALC}\bar{\mathcal{N}}$) allowing for the same kind of complex roles *both* in number and in value restrictions, which represent very expressive description languages and can be shown very useful for applications.

We investigate the computational properties of various $\mathcal{ALCN}$ extensions and slightly improve the (un)decidability results following from the study published by Baader and Sattler in 1999. In particular, we will show by reduction of a domino problem that $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$ are undecidable.

1 Introduction

The starting point of our study is the Description Logic $\mathcal{ALCN}$ [1], that is the well-known $\mathcal{ALC}$ language [13] equipped with (non-qualified) *number restrictions* on possibly complex roles ($\mathcal{N}$). Expressive extensions of $\mathcal{ALCN}$ can be defined as $\mathcal{ALCN}(M)$ with the adoption of role constructors $M \subseteq \{^+, \circ, ^-, \sqcup, \sqcap\}$ [1]. By allowing different kinds of complex roles also in value (existential) restrictions, different families of Logics can also be defined: for example $\mathcal{ALC}_+\mathcal{N}$ (or $\mathcal{ALC}_{\text{reg}}\mathcal{N}$) allows the transitive closure of atomic roles (or regular roles) under value restrictions [1, 5].

In particular, since their rich expressiveness and versatility is highly appealing from an application perspective, we are mainly interested in $\mathcal{ALCN}$ extensions, that we will denote by $\mathcal{ALC}\bar{\mathcal{N}}$, equipped with the same combination of role constructors either in value *and* in number restrictions. As a matter of fact, expressive Description Logics [5] possibly providing for (qualified) number restrictions have been used for solving interesting reasoning tasks in the fields

$$
\begin{array}{rlrcl}
C, D \to & A \mid & A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} & & \text{atomic concept} \\
& \top \mid & \top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
& \bot \mid & \bot^{\mathcal{I}} & = & \emptyset \\
& \neg C \mid & (\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
& C \sqcap D \mid & (C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
& C \sqcup D \mid & (C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
& \forall R.C \mid & (\forall R.C)^{\mathcal{I}} & = & \{i \in \Delta^{\mathcal{I}} \mid \forall j.\ R^{\mathcal{I}}(i,j) \Rightarrow C^{\mathcal{I}}(j)\} \\
& \exists R.C \mid & (\exists R.C)^{\mathcal{I}} & = & \{i \in \Delta^{\mathcal{I}} \mid \exists j.\ R^{\mathcal{I}}(i,j) \wedge C^{\mathcal{I}}(j)\} \\
& \exists^{\geq n} R \mid & (\exists^{\geq n} R)^{\mathcal{I}} & = & \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j)\} \geq n\} \\
& \exists^{\leq n} R \mid & (\exists^{\leq n} R)^{\mathcal{I}} & = & \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(i,j)\} \leq n\} \\
R, S \to & P \mid & P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} & & \text{atomic role} \\
& R^{+} \mid & (R^{+})^{\mathcal{I}} & = & \cup_{i \geq 1} (R^{\mathcal{I}})^{i} \\
& R \circ S \mid & (R \circ S)^{\mathcal{I}} & = & \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists k.\ R^{\mathcal{I}}(i,k) \wedge S^{\mathcal{I}}(k,j)\} \\
& R^{-} \mid & (R^{-})^{\mathcal{I}} & = & \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(j,i)\} \\
& R \sqcup S \mid & (R \sqcup S)^{\mathcal{I}} & = & R^{\mathcal{I}} \cup S^{\mathcal{I}} \\
& R \sqcap S & (R \sqcap S)^{\mathcal{I}} & = & R^{\mathcal{I}} \cap S^{\mathcal{I}}
\end{array}
$$

Figure 1: Syntax and model-theoretic semantics of $\mathcal{ALCN}$ extensions.

of knowledge representation and conceptual modeling (e.g. for reasoning on database schemata [7, 6, 8]). In this respect, the characterization of the computational behaviour of such description languages is the foundation to ground their effective usability in applications.

Our investigation is aimed at improving the (un)decidability results presented by Baader and Sattler in [1]. In particular, they proved that concept satisfiability in $\mathcal{ALCN}(\circ, \sqcap)$ and $\mathcal{ALCN}(\circ, ^{-}, \sqcup)$ (and, thus, in $\mathcal{ALC}\bar{\mathcal{N}}(\circ, \sqcap)$ and $\mathcal{ALC}\bar{\mathcal{N}}(\circ, ^{-}, \sqcup)$) is undecidable, and undecidability of $\mathcal{ALC}\bar{\mathcal{N}}(^{+}, \circ)$ is a straightforward consequence of their results for $\mathcal{ALC}_{+}\mathcal{N}(\circ)$. Furthermore, $\mathcal{ALCN}(^{-}, \sqcup, \sqcap)$ [1], $\mathcal{ALC}\bar{\mathcal{N}}(^{-}, \sqcup, \sqcap)$ and all their sublanguages are known to be decidable since $\mathcal{ALC}\bar{\mathcal{N}}(^{-}, \sqcup, \sqcap)$-concepts can easily be translated into a formula in $\mathcal{C}^2$ [3], that is the two-variable FOL fragment with counting quantifiers, which has proved to be decidable [10]. In fact, satisfiability of $\mathcal{C}^2$ formulae can be decided in NExpTime [14] 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). Baader and Sattler also presented in [1] a sound and complete Tableau algorithm for deciding satisfiability of $\mathcal{ALCN}(\circ)$-concepts; anyway, the decidability of other $\mathcal{ALCN}(\circ)$ extensions beyond the undecidable $\mathcal{ALC}\bar{\mathcal{N}}(^{+}, \circ)$ is still unknown. For example, to the best of our knowledge, decidability of $\mathcal{ALCN}(\circ, ^{-})$ (and $\mathcal{ALC}\bar{\mathcal{N}}(\circ, ^{-})$) is still an open problem [1, 5]. Moreover, whereas $\mathcal{ALC}_{+}\mathcal{N}(^{-})$ is known to be decidable (due to the results of De Giacomo and Lenzerini in [9]), it is not known what it happens to its extension with a *unrestricted* use of transitive closure $\mathcal{ALC}_{+}\mathcal{N}(^{+}, ^{-})$, and further to $\mathcal{ALC}\bar{\mathcal{N}}(^{+}, ^{-})$.

Other extensions with potential interest for applications and which will be investigated in this paper are also $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$. For example, by means of the following $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ concept:

$$\texttt{Patriarch} \sqsubseteq \texttt{LivingMan} \sqcap \exists^{\geq 10}(\texttt{daughter} \sqcup \texttt{son})^+$$

we could easily define a "patriarch" as a living man having at least ten descendants, whereas the $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$ concept:

$$\texttt{PhDStudent} \sqcap \exists^{\leq 0}(\texttt{parent}^+ \sqcap \texttt{advisor})$$

denotes the PhD students whose advisor is not an ancestor of them.

In this paper, we will slightly extend the (un)decidability results on $\mathcal{ALC}\bar{\mathcal{N}}$ languages which directly follow from the results by Baader and Sattler in [1], by showing undecidability of $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$ via reduction of a domino problem.

Since we may assume the reader be familiar with Description Logics, we shall not indulge in preliminaries. The syntax and the semantics of $\mathcal{ALCN}$ with the extensions considered in this paper are sketched in Fig. 1.

2 Undecidable $\mathcal{ALC}\bar{\mathcal{N}}(^+)$ Extensions

In this Section we will widen the undecidability results known for $\mathcal{ALCN}$ extensions including the transitive closure operator, by showing that $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$ Logics are undecidable[1]. To prove that, we borrow the proof procedure from [1] and use a reduction of the well-known undecidable domino problem [2] adapted from [12]:

Definition 1 *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 domino problem asks for a compatible tiling of the first quadrant $\mathbb{N} \times \mathbb{N}$ of the plane, i.e. a mapping $t : \mathbb{N} \times \mathbb{N} \to D$ such that, for all $m, n \in \mathbb{N}$, $\langle\, t(m, n), t(m + 1, n)\,\rangle \in H$ and $\langle\, t(m, n), t(m, n + 1)\,\rangle \in V$.*

We will show now reducibility of the domino problem to *concept satisfiability* in the desired Description Logics. 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. Since $\mathcal{ALC}$ is propositionally complete, also *subsumption* can be reduced to concept satisfiability ($C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable). Following the same lines of undecidability proofs in [1], such translation can be split into three subtasks which can be described as follows:

[1] Actually, we will prove undecidability of $\mathcal{ALC}_+\mathcal{N}(^+, \sqcup)$ and $\mathcal{ALC}_+\mathcal{N}(^+, \sqcap)$.

Grid specification . It must be possible to represent a "square" of $\mathbb{N} \times \mathbb{N}$, 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 first quadrant 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 relationships 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 condition via value restrictions on the stepping roles.

Total reachability . It must be possible to impose the above local conditions on all points in $\mathbb{N}\times\mathbb{N}$. 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.

In particular, to solve our subtasks, we will use a proof framework quite similar to the one adopted by Horrocks, Sattler and Tobies in the context of a strictly related undecidability proof which is described, for instance, in [11]. They used a grid similar to the one in Fig. 2, embedding an alternating pattern of "horizontal" and "vertical" roles, with four disjoint primitive concepts representing grid points with different combinations of successors. Horrocks, Sattler and Tobies used that framework to prove that allowing transitive roles (or roles having a transitive subrole) into number restrictions in an $\mathcal{ALC}$ extension —called $\mathcal{SHIN}$— supporting an explicit role hierarchy and transitive (transitively closed primitive) roles, leads to undecidability. Their framework is not directly applicable to our cases, since we cannot *explicitly* represent a role hierarchy in $\mathcal{ALCN}$ languages. However, we will exploit the "natural" hierarchy *implicitly* induced by the inclusion between a role and a Boolean expression containing it. As a matter of fact, if R and S are arbitrary roles, due to the semantics of the $\sqcup$ and $\sqcap$ constructors, we have the trivial inclusions $(R \sqcap S) \sqsubseteq R$ (which was also exploited in [4], where a language that we would call $\mathcal{ALCN}(\sqcap)$ was proved decidable) and $R \sqsubseteq (R \sqcup S)$.

2.1 Undecidability of $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$

In this case, the concepts to encode the domino problem must be built using number restrictions on complex roles involving role union and transitive closure. The first quadrant will be covered by means of the grid structure represented in Fig. 2. Five roles will be used to represent the connections between grid points,

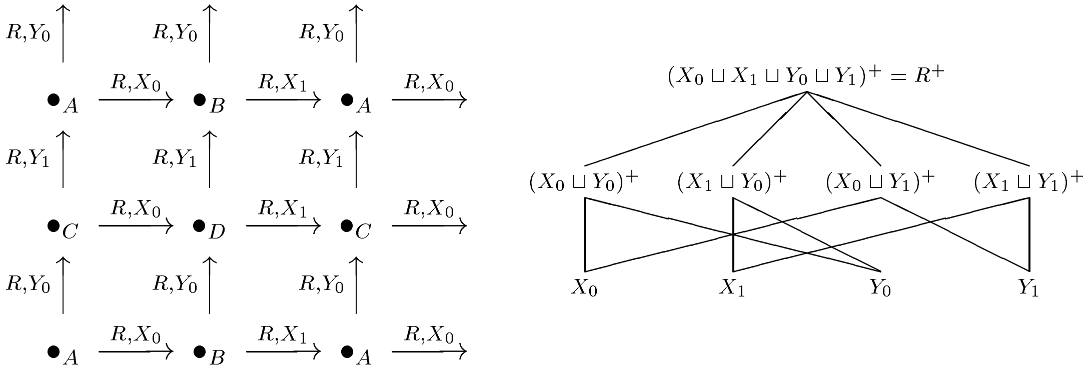

Figure 2: The grid structure and role hierarchy used for Theorem 1.

two (namely X_0 and X_1) representing a single step into the horizontal direction and two (namely Y_0 and Y_1) representing a single step into the vertical direction; the fifth one (R) is used to represent a generic forward step in the grid (so that $R = X_0 \sqcup X_1 \sqcup Y_0 \sqcup Y_1$). The implicit role hierarchy underlying the used roles is also drawn in the Figure.

Grid specification can be accomplished by means of the following concept:

$$\begin{aligned}
C_{\boxplus} \; := \; & \big(A \Rightarrow (\exists X_0.B \sqcap \exists^{=1}X_0 \sqcap \exists Y_0.C \sqcap \exists^{=1}Y_0 \sqcap \\
& \qquad \exists^{=2}R \sqcap \exists^{=2}(X_0 \sqcup Y_0 \sqcup R) \sqcap \exists^{=3}(X_0 \sqcup Y_0)^+)\big) \sqcap \\
& \big(B \Rightarrow (\exists X_1.A \sqcap \exists^{=1}X_1 \sqcap \exists Y_0.D \sqcap \exists^{=1}Y_0 \sqcap \\
& \qquad \exists^{=2}R \sqcap \exists^{=2}(X_1 \sqcup Y_0 \sqcup R) \sqcap \exists^{=3}(X_1 \sqcup Y_0)^+)\big) \sqcap \\
& \big(C \Rightarrow (\exists X_0.D \sqcap \exists^{=1}X_0 \sqcap \exists Y_1.A \sqcap \exists^{=1}Y_1 \sqcap \\
& \qquad \exists^{=2}R \sqcap \exists^{=2}(X_0 \sqcup Y_1 \sqcup R) \sqcap \exists^{=3}(X_0 \sqcup Y_1)^+)\big) \sqcap \\
& \big(D \Rightarrow (\exists X_1.C \sqcap \exists^{=1}X_1 \sqcap \exists Y_1.B \sqcap \exists^{=1}Y_1 \sqcap \\
& \qquad \exists^{=2}R \sqcap \exists^{=2}(X_1 \sqcup Y_1 \sqcup R) \sqcap \exists^{=3}(X_1 \sqcup Y_1)^+)\big)
\end{aligned}$$

where, as usual, $\exists^{=n}R$ and $P \Rightarrow Q$ are shorthands for the concepts $(\exists^{\leq n}R) \sqcap (\exists^{\geq n}R)$ and $\neg P \sqcup Q$, respectively. In this way, every point in the grid is described as having exactly one x-successor and one y-successor (e.g. an A-type point is connected through X_0 to a B-type and through Y_0 to a C-type point, and so on). Furthermore, every A-type point is connected, through any occurrence of X_0 or X_1 to exactly three other points. W.l.o.g., let $p_{(m,n)}$ be one chosen A-type point. It has exactly one X_0 direct successor, which we can draw to its right and call $p_{(m+1,n)}$, one Y_0 direct successor, which we can draw above it and call $p_{(m,n+1)}$, one common $X_0 \circ Y_0$ and $Y_0 \circ X_0$ successor, which we can draw as the D-type point closing the square on the top-right corner and that we can call

$p_{(m+1,n+1)}$. Moreover, $p_{(m,n)}$ is connected to exactly two R-successors, which are also $(X_0 \sqcup Y_0)$-successors. Since the two $(X_0 \sqcup Y_0)$-successors of $p_{(m,n)}$ are exactly $p_{(m+1,n)}$ and $p_{(m,n+1)}$, they can be reached from $p_{(m,n)}$ also via R. Similarly, we can consider the construction of squares starting from a B-, C- or D-type point. With this intuition (see also Fig. 2), it is easy to understand how the complete grid can be drawn and how each point is connected to its direct x- and y-successors via R.

Local compatibility is easy to achieve by means of value restrictions. The concept $C_{\mathcal{D}}$ which serves the purpose (which is also very similar to the $C_{\text{prim}} \sqcap C_{\mathcal{D}}$ concept used in [1, Th. 6]) is the following:

$$\begin{aligned}
C_{\mathcal{D}} \;:=\; & \bigsqcup_{E \in \{A,B,C,D\}} (E \sqcap (\bigsqcap_{F \in \{A,B,C,D\} \setminus \{E\}} \neg F)) \sqcap \bigsqcup_{1 \leq i \leq m} \Big(D_i \sqcap (\bigsqcap_{1 \leq i \leq m, i \neq j} \neg D_j)\Big) \sqcap \\
& \bigsqcap_{1 \leq i \leq m} \Big(\big((A \sqcap D_i) \Rightarrow (\exists X_0.(B \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists Y_0.(C \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))\big) \sqcap \\
& \qquad \big((B \sqcap D_i) \Rightarrow (\exists X_1.(A \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists Y_0.(D \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))\big) \sqcap \\
& \qquad \big((C \sqcap D_i) \Rightarrow (\exists X_0.(D \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists Y_1.(A \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))\big) \sqcap \\
& \qquad \big((D \sqcap D_i) \Rightarrow (\exists X_1.(C \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists Y_1.(B \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))\big) \Big)
\end{aligned}$$

Total reachability is also easy to ensure, due to the availability of the transitive closure of roles:

$$E_{\mathcal{D}} \;:=\; \exists R.A \sqcap \exists^{=1} R \sqcap \forall R^{+}.(C_{\boxplus} \sqcap C_{\mathcal{D}})$$

If s is an instance of $E_{\mathcal{D}}$, then s has exactly one R-successor, which is an instance of A (we could have chosen a B, C or D instance as well), say $p_{(0,0)}$. Each point in the grid can be reached from s via the universal role R^{+} (e.g. $p_{(1,0)}$ and $p_{(0,1)}$ are direct R-successors of $p_{(0,0)}$, so they can also be reached from s via R^{+} and so on) and, thus, the local conditions are imposed on all points in the grid by $\forall R^{+}.(C_{\boxplus} \sqcap C_{\mathcal{D}})$.

With the intuition given above, 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 $\mathcal{ALC}_{+}\mathcal{N}(^{+}, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^{+}, \sqcup)$.*

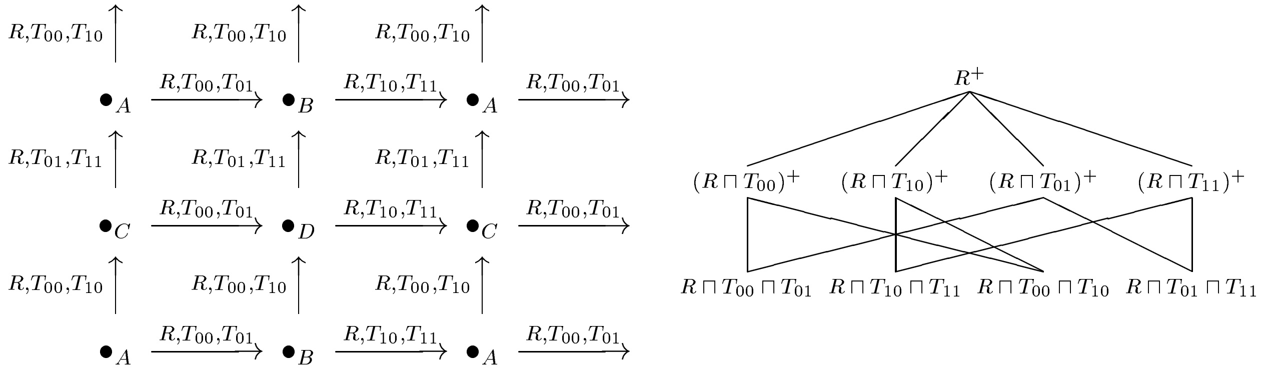

Figure 3: The grid structure and role hierarchy used for Theorem 2.

2.2 Undecidability of $\mathcal{ALC}\bar{\mathcal{N}}(^+,\sqcap)$

In this case, the concepts to encode the domino problem must be built using number restrictions on complex roles involving role intersection and transitive closure. The first quadrant will be covered by means of the grid structure represented in Fig. 3. Five roles will be used to represent the connections between grid points: one role R (whose transitive closure is the universal role) connecting each point with all its direct successors, and four roles T_{00}, T_{10}, T_{01} and T_{11}, connecting (to their x- and y- successors) the points in a square having an A-type, B-type, C-type and D-type bottom-left vertex, respectively (see Fig. 3). The new roles are related to the roles used in Theorem 1 as follows:

$$\begin{array}{c} X_0 := T_{00} \sqcap T_{01},\ X_1 := T_{10} \sqcap T_{11},\ Y_0 := T_{00} \sqcap T_{10},\ Y_1 := T_{01} \sqcap T_{11} \\ T_{00} := X_0 \sqcup Y_0,\ T_{01} := X_0 \sqcup Y_1,\ T_{10} := X_1 \sqcup Y_0,\ T_{11} := X_1 \sqcup Y_1 \end{array}$$

The implicit role hierarchy is also depicted in Fig.3.

Grid specification can be accomplished by means of the $C_{\boxplus}$ concept which follows:

$$\begin{array}{rcl} C_{\boxplus} & := & \big(A \Rightarrow (\exists T_{00}.B \sqcap \exists T_{01}.B \sqcap \exists R.B \sqcap \exists T_{00}.C \sqcap \exists T_{10}.C \sqcap \exists R.C \\ & & \qquad \sqcap \exists^{=1}(T_{00} \sqcap T_{01} \sqcap R) \sqcap \exists^{=1}(T_{00} \sqcap T_{10} \sqcap R) \sqcap \exists^{=3}T_{00}^{+})\big) \\ & \sqcap & \big(B \Rightarrow (\exists T_{10}.A \sqcap \exists T_{11}.A \sqcap \exists R.A \sqcap \exists T_{00}.D \sqcap \exists T_{10}.D \sqcap \exists R.D \\ & & \qquad \sqcap \exists^{=1}(T_{10} \sqcap T_{11} \sqcap R) \sqcap \exists^{=1}(T_{00} \sqcap T_{10} \sqcap R) \sqcap \exists^{=3}T_{10}^{+})\big) \\ & \sqcap & \big(C \Rightarrow (\exists T_{00}.D \sqcap \exists T_{01}.D \sqcap \exists R.D \sqcap \exists T_{01}.A \sqcap \exists T_{11}.A \sqcap \exists R.A \\ & & \qquad \sqcap \exists^{=1}(T_{00} \sqcap T_{01} \sqcap R) \sqcap \exists^{=1}(T_{01} \sqcap T_{11} \sqcap R) \sqcap \exists^{=3}T_{01}^{+})\big) \\ & \sqcap & \big(D \Rightarrow (\exists T_{10}.C \sqcap \exists T_{11}.C \sqcap \exists R.C \sqcap \exists T_{01}.B \sqcap \exists T_{11}.B \sqcap \exists R.B \\ & & \qquad \sqcap \exists^{=1}(T_{10} \sqcap T_{11} \sqcap R) \sqcap \exists^{=1}(T_{01} \sqcap T_{11} \sqcap R) \sqcap \exists^{=3}T_{11}^{+})\big) \end{array}$$

In this way, every point in the grid is described as having exactly one x-successor and one y-successor (e.g. an A-type point is connected through T_{00} and T_{01} (and R) to a B-type and through T_{00} and T_{10} (and R) to a C-type point, and so on). Furthermore, every A-type point is connected, through any occurrence of T_{00} to exactly three other points. W.l.o.g., let $p_{(m,n)}$ be one chosen A-type point. It has exactly one x-successor, which we can draw to its right and call $p_{(m+1,n)}$, one y-successor, which we can draw above it and call $p_{(m,n+1)}$, and one more common T_{00}^{+}-successor (which is also an x-successor of a C-type point and an y-successor of a B-type point) which we can draw as the D-type point closing the square on the top-right corner and that we can call $p_{(m+1,n+1)}$. Obviously, $p_{(m+1,n)}$ and $p_{(m,n+1)}$ can be reached from $p_{(m,n)}$ also via R. Similarly, we can consider the construction of squares starting from a B-, C- or D-type point. With this intuition (see also Fig. 3), it is easy to understand how the complete grid can be drawn and how each point is connected to its direct x- and y-successors via R.

Local compatibility is easy to achieve by means of a $C_{\mathcal{D}}$ concept very similar to the one used in Theorem 1, for instance:

$$\begin{aligned}
C_{\mathcal{D}} \ :=\ & \bigsqcup_{E\in\{A,B,C,D\}}\big(E \sqcap (\bigsqcap_{F\in\{A,B,C,D\}\setminus\{E\}} \neg F)\big) \sqcap \bigsqcup_{1\le i\le m}\Big(D_i \sqcap (\bigsqcap_{1\le i\le m, i\ne j} \neg D_j)\Big) \sqcap \\
\bigsqcap_{1\le i\le m}\Bigg(& \big((A \sqcap D_i) \Rightarrow (\exists T_{01}.(B \sqcap (\bigsqcup_{(D_i,D_j)\in H} D_j)) \sqcap \exists T_{10}.(C \sqcap (\bigsqcup_{(D_i,D_j)\in V} D_j)))\big) \sqcap \\
& \big((B \sqcap D_i) \Rightarrow (\exists T_{11}.(A \sqcap (\bigsqcup_{(D_i,D_j)\in H} D_j)) \sqcap \exists T_{00}.(D \sqcap (\bigsqcup_{(D_i,D_j)\in V} D_j)))\big) \sqcap \\
& \big((C \sqcap D_i) \Rightarrow (\exists T_{00}.(D \sqcap (\bigsqcup_{(D_i,D_j)\in H} D_j)) \sqcap \exists T_{11}.(A \sqcap (\bigsqcup_{(D_i,D_j)\in V} D_j)))\big) \sqcap \\
& \big((D \sqcap D_i) \Rightarrow (\exists T_{10}.(C \sqcap (\bigsqcup_{(D_i,D_j)\in H} D_j)) \sqcap \exists T_{01}.(B \sqcap (\bigsqcup_{(D_i,D_j)\in V} D_j)))\big)\Bigg)
\end{aligned}$$

Total reachability is straightforward to ensure, as the same $E_{\mathcal{D}}$ as in Theorem 1 can be used:

$$E_{\mathcal{D}} \ :=\ \exists R.A \sqcap \exists^{=1} R \sqcap \forall R^{+}.(C_{\boxplus} \sqcap C_{\mathcal{D}})$$

With the intuition given above, 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 2 *Satisfiability (and, thus, subsumption) of concepts is undecidable for $\mathcal{ALC}_{+}\mathcal{N}(^{+},\sqcap)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^{+},\sqcap)$.*

3 Conclusions

In this paper we studied expressive Description Logics $\mathcal{ALC}\bar{\mathcal{N}}$, allowing both for value and number restrictions on complex roles built with *combinations* of constructors.

In particular, we slightly improved the (un)decidability results by Baader and Sattler on logics of the $\mathcal{ALCN}$ family [1] by showing by reduction of a domino problem that $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcup)$ and $\mathcal{ALC}\bar{\mathcal{N}}(^+, \sqcap)$ are undecidable.

Acknowledgment

The author wish to thank Enrico Franconi for his always valuable bibliographic suggestions.

References

[1] F. Baader and U. Sattler. Expressive Number Restrictions in Description Logics. *Journal of Logic and Computation*, 9(3):319–350, 1999.

[2] R. Berger. The Undecidability of the Dominoe Problems. *Mem. Amer. Math. Soc.*, 66, 1966.

[3] A. Borgida. On the Relative Expressiveness of Description Logics and First Order Logics. *Artificial Intelligence*, 82:353–367, 1996.

[4] M. Buchheit, F. Donini, and A. Schaerf. Decidable Reasoning in Terminological Knowledge Representation Systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1999.

[5] D. Calvanese, G. De Giacomo, M. Lenzerini, and D. Nardi. Reasoning in Expressive Description Logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier, 2000.

[6] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Information Integration: Conceptual Modeling and Reasoning Support. In *Proc. of Intl' Conf. on Cooperative Information Systems (CoopIS'98)*, 1998.

[7] D. Calvanese, M. Lenzerini, and D. Nardi. Description Logics for Conceptual Data Modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publishers, 1998.

[8] 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)*, 2000.

[9] 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)*, 1996.

[10] E. Grädel, M. Otto, and E. Rosen. Two-variable Logic with Counting is Decidable. In *Proc. 12th Annual IEEE Symposium on Logic in Computer Science*, 1997.

[11] I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logics Journal of the IGPL*, 3(3):239–263, 2000.

[12] D. Knuth. *The Art of Computer Programming Vol. I – Fundamental Algorithms.* Addison Wesley, 1968.

[13] M. Schmidt-Schauß and G. Smolka. Attributive Concept Descriptions with Complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[14] L. Pacholski, W. Szwast, and L. Tendera. Complexity of Two-variable Logic with Counting. In *Proc. 12th Annual IEEE Symposium on Logic in Computer Science*, 1997.