Expressive number restrictions in Description Logics*

Franz Baader and Ulrike Sattler

August 12, 1999

Abstract

Number restrictions are concept constructors that are available in almost all implemented Description Logic systems. However, they are mostly available only in a rather weak form, which considerably restricts their expressive power.

On the one hand, the roles that may occur in number restrictions are usually of a very restricted type, namely atomic roles or complex roles built using either intersection or inversion. In the present paper, we increase the expressive power of Description Logics by allowing for more complex roles in number restrictions. As role constructors, we consider composition of roles (which will be present in all our logics) and intersection, union, and inversion of roles in different combinations. We will present two decidability results (for the basic logic that extends $\mathcal{ALC}$ by number restrictions on roles with composition, and for one extension of this logic), and three undecidability results for three other extensions of the basic logic.

On the other hand, with the rather weak form of number restrictions available in implemented systems, the number of role successors of an individual can only be restricted by a fixed non-negative integer. To overcome this lack of expressiveness, we allow for variables ranging over the non-negative integers in place of the fixed numbers in number restrictions. The expressive power of this constructor is increased even further by introducing explicit quantifiers for the numerical variables. The Description Logic obtained this way turns out to have an undecidable satisfiability problem. For a restricted logic we show that concept satisfiability is decidable.

1 Introduction

Description Logics provide *constructors* that can be used to build complex concepts and roles from atomic concepts (unary predicates) and roles (binary predicates). The well-known Description Logic $\mathcal{ALC}$ [24] allows for *propositional* constructors $\sqcap, \sqcup, \neg$ on concepts as well as for *universal* and *existential value restrictions*. For example,[1] the following concept describes happy parents as humans having a nice child and whose children are happy and have some nice friends:

$$\texttt{Human} \sqcap (\exists \texttt{child}.\texttt{Nice}) \sqcap (\forall \texttt{child}.(\texttt{Happy} \sqcap (\exists \texttt{friend}.\texttt{Nice}))).$$

* To appear in: *Journal of Logic and Computation*, Vol. 9(3), 1999

[1] This investigation was motivated by a process engineering application. However, to present our results in a way that is more intuitive for readers not familiar with process engineering, we give examples concerning families.

The general idea underlying knowledge representation systems based on Description Logics (DL-systems) is the following. First, the *terminology* of an application domain is fixed. In the terminology below, *number restrictions* are used to describe parents as those humans having at least one child, parents of many children as those having at least four children, etc. Number restrictions allow one to restrict the number of *role-successors*, that is, the number of those objects an object is related to via a role. In this example, $(\geq 4\ \mathtt{child})$ restricts the number of $\mathtt{child}$-successors to at least 4, whereas $(\leq 2\ \mathtt{child})$ restricts this number to at most 2.

$$
\begin{array}{rcl}
\mathtt{Parent} & := & \mathtt{Human} \sqcap (\geq 1\ \mathtt{child}) \\
\mathtt{Parent\_of\_many} & := & \mathtt{Parent} \sqcap (\geq 4\ \mathtt{child}) \\
\mathtt{Parent\_of\_few} & := & \mathtt{Parent} \sqcap (\leq 2\ \mathtt{child}) \\
\mathtt{HappyParent} & := & \mathtt{Human} \sqcap (\exists \mathtt{child}.\mathtt{Nice}) \\
 & & \sqcap (\forall \mathtt{child}.(\mathtt{Happy} \sqcap (\exists \mathtt{friend}.\mathtt{Nice})))
\end{array}
$$

In the next modelling step, this terminology can be used to describe a concrete "world." DL-systems are designed to *reason* about both the terminology and the description of concrete worlds. For example, they should be able to infer that $\mathtt{HappyParent}$, $\mathtt{Parent\_of\_many}$, and $\mathtt{Parent\_of\_few}$ are subsumed by $\mathtt{Parent}$. Another relevant inference problem is to decide whether a given concept is satisfiable, that is, whether its description is non-contradictory.

To be useful in an application, the *expressive power* of a given Description Logic must be adequate for the application (see [2, 15] for a formal definition of expressive power). Intuitively, the Description Logic should allow one to describe the relevant properties of objects of the application.

Number restrictions appear to provide expressive power required by many applications. Moreover, humans also tend to describe objects by restricting the number of objects they are related to. As a consequence, number restrictions are present in most implemented DL systems [16, 20, 21, 3]. Unfortunately, they are usually found in their weakest form:

1. They are *not qualifying*, that is, we may not restrict the number of role-successors of a certain kind, but only the total number of role successors. For example, we cannot restrict the number of children *that are girls*, but we can only restrict the total number of children.

2. Inside number restrictions, only atomic roles are allowed, that is, *complex* roles built using some role-forming constructors are disallowed. Thus, one cannot restrict the number of grandchildren using only the role $\mathtt{child}$.

3. Finally, it is only possible to restrict the number of role-successors to at least or at most n, for a *fixed* non-negative integer n. For example, it is not possible to describe persons having *more* children than they have friends or persons having *the same number of* children as their spouse or husband—without fixing a bound for this number.

The first shortcoming has been overcome in [12], where so-called *qualifying* number restrictions were introduced. For example, $(\geq 4\ \mathtt{child}\ \mathtt{Girl})$ is a qualifying number restriction describing parents having at least four children that are girls. To overcome the second and third shortcoming, we will introduce *complex roles in number restrictions* and *symbolic number restrictions*.

Complex roles in number restrictions

Complex roles are built using role constructors such as composition, union, intersection, inversion (or converse), or the transitive closure of roles. It was shown that Description Logics can be extended with complex roles in value restrictions without losing decidability of the relevant inference problems [1, 22, 23, 7, 6, 8]. However, investigations of the computational complexity of complex roles in number restrictions were restricted to intersection [9] and inversion [5]. If both complex roles and number restrictions are present in a Description Logic, one thus must distinguish between the roles allowed in value restrictions and those allowed in number restrictions.

By restricting the use of complex roles to value restrictions, one loses expressiveness, as illustrated by the following examples. For example, by using *composition* of roles in number restrictions one can describe persons having at least four grandchildren:

$$\texttt{Human} \sqcap (\geq 4\ \texttt{child} \circ \texttt{child}).$$

To describe those persons whose children still live at home, additionally, the *union* of roles inside number restrictions is needed:

$$\texttt{Human} \sqcap (= 1\ \texttt{has-address} \sqcup (\texttt{child} \circ \texttt{has-address})).$$

To describe persons having at least five siblings, *inversion* comes into play:

$$\texttt{Human} \sqcap (\geq 6\ \texttt{child} \circ \texttt{child}^{-1}).$$

Finally, using *intersection* of roles, we can describe persons having at least five friends in common with their husband or spouse:

$$\texttt{Human} \sqcap (\geq 5\ \texttt{friend} \sqcap (\texttt{married-to} \circ \texttt{friend})).$$

Symbolic number restrictions

In traditional number restrictions, we always have to fix a non-negative integer by which the number of role successors is restricted. Thus, we cannot describe, for example, parents whose children like at least as many things as they dislike—without giving an upper bound on the number of things their children may dislike. Symmetry-conditions like the one above (i.e., conditions of the form "having the same number of xs as of ys) often occur in practice, but they cannot be expressed using traditional number restrictions.

To overcome this lack of expressiveness, we introduce numerical variables $\alpha, \beta, \ldots$ to be used in number restrictions. Thus, the above example can be described by

$$\texttt{Parent} \sqcap \forall \texttt{child}.((= \alpha\ \texttt{dislikes}) \sqcap (> \alpha\ \texttt{likes})),$$

where α is supposed to be interpreted by a non-negative integer. This example reveals a certain ambiguity: the exact meaning of the concept expression depends on whether the variable α must be interpreted by the same non-negative integer for all children, or whether it can have different values for different children. To avoid this ambiguity, we will introduce explicit existential quantification of numerical variables (denoted by $\downarrow\alpha$) to distinguish between (1) parents all of whose children like more things than they dislike

$$\texttt{Parent} \sqcap (\forall \texttt{child}.(\downarrow\alpha.(= \alpha\ \texttt{dislikes}) \sqcap (> \alpha\ \texttt{likes}))), \tag{1}$$

and (2) parents where all children dislike the same number of things, and like more things than they dislike:

$$\texttt{Parent} \sqcap (\downarrow\alpha.(\forall \texttt{child}\,.\,(= \alpha\ \texttt{dislikes}) \sqcap (> \alpha\ \texttt{likes}))). \tag{2}$$

Outline of this paper

In the following, these two ways of augmenting the expressive power of number restrictions are investigated in detail. It turns out that these extensions are of such a high expressive power that they lead, in many cases, to undecidability. To keep things as simple as possible, we will restrict our attention to the basic inference problems subsumption and satisfiability of concepts, and not mix both extensions. In Section 2, the basic Description Logics and the relevant inference problems are introduced.

In Section 3, the extensions by complex roles in number restrictions are introduced, and their computational properties are investigated. Extensions of $\mathcal{ALC}$ by different kinds of complex roles in number restrictions are almost completely classified with respect to the decidability of the satisfiability and subsumption problem. These results are obtained either as a consequence of a general decidability result in [11], or they are explicitly proved in this paper. The latter ones include the

- decidability of $\mathcal{ALC}$ with composition in number restrictions,
- undecidability of $\mathcal{ALC}$ with composition and intersection in number restrictions,
- undecidability of $\mathcal{ALC}$ with composition, union, and inversion in number restrictions.

In addition, we also consider $\mathcal{ALC}_+$(i.e., the extension of $\mathcal{ALC}$ by transitive closure of roles in value restrictions), and show that its extension by number restrictions on roles with composition is undecidable.

In Section 4, symbolic number restrictions are introduced. It turns out that, for "full" symbolic number restrictions, satisfiability and subsumption are undecidable, whereas a restriction to the kind of symbolic number restrictions used in all of the above examples leads to decidability of satisfiability. Unfortunately, this restriction leads to a logic that is no longer closed under negation, and it turns out that, for this logic, the subsumption problem is still undecidable.

Finally, in Section 5, we mention related decidability and undecidability results from Description Logics, Modal Logics, and Predicate Logic.

2 Preliminaries

All investigations in this work concern extensions of the Description Logic $\mathcal{ALCN}$ [13, 9], which is the extension of $\mathcal{ALC}$ [24] with (non-qualifying) number restrictions on atomic roles. For these two Description Logics, both satisfiability and subsumption are decidable. More precisely, these inference problems were shown to be PSpace-complete [13, 9].

Definition 1 Let N_C be a set of *concept names*, and N_R a set of *role names*. The set of $\mathcal{ALC}$*-concepts* is the smallest set such that

- every concept name is a concept.

- if C and D are concepts and R is a role name, then
  - $(C \sqcap D),\ (C \sqcup D),\ (\neg C),$ (Boolean operators)
  - $(\forall R.C),\ (\exists R.C)$ (value restrictions)

  are concepts.

Starting with role names in N_R, *regular roles* are built using the role constructors composition $(R \circ S)$, union $(R \sqcup S)$, and transitive closure (R^+).

- $\mathcal{ALC}_{\mathsf{reg}}$ is obtained from $\mathcal{ALC}$ by allowing, additionally, for regular roles in value restrictions.
- $\mathcal{ALC}_+$ is obtained from $\mathcal{ALC}$ by allowing, additionally, for the transitive closure of roles in value restrictions.
- $\mathcal{ALCN}$ (resp. $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}$ and $\mathcal{ALC}_+\mathcal{N}$) is obtained from $\mathcal{ALC}$ (resp. $\mathcal{ALC}_{\mathsf{reg}}$ and $\mathcal{ALC}_+$) by allowing, additionally, for concepts of the form $(\geq n\ R)$ and $(\leq n\ R)$ (number restrictions), for all role names R and non-negative integers n.

In the next section, we will also consider the additional role constructors intersection $(R \sqcap S)$ and inversion (R^{-1}).

The meaning of these constructors, and thus also of the Description Logics we have just introduced, is defined using a Tarski-style model-theoretic semantics.

Definition 2 An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and an extension function $\cdot^{\mathcal{I}}$ that maps every concept to a subset of $\Delta^{\mathcal{I}}$, and every (complex) role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that the following equalities are satisfied:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
\neg C^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\
(\exists R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d,e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}, \\
(\forall R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \forall e \in \Delta^{\mathcal{I}} : (d,e) \in R^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}}\}, \\
(\geq n\ R)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid (d,e) \in R^{\mathcal{I}}\} \geq n\}, \\
(\leq n\ R)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid (d,e) \in R^{\mathcal{I}}\} \leq n\}, \\
(R_1 \sqcup R_2)^{\mathcal{I}} &= {R_1}^{\mathcal{I}} \cup {R_2}^{\mathcal{I}}, \\
(R_1 \sqcap R_2)^{\mathcal{I}} &= {R_1}^{\mathcal{I}} \cap {R_2}^{\mathcal{I}}, \\
(R^{-1})^{\mathcal{I}} &= \{(d,e) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (e,d) \in R^{\mathcal{I}}\}, \\
(R_1 \circ R_2)^{\mathcal{I}} &= \{(d,f) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d,e) \in R_1^{\mathcal{I}} \wedge (e,f) \in R_2^{\mathcal{I}}\}, \\
(R^+)^{\mathcal{I}} &= \cup_{i \geq 1} (R^{\mathcal{I}})^i.
\end{aligned}
$$

where $\#X$ denotes the cardinality of a set X and $(R^{\mathcal{I}})^i$ the i-times composition of $R^{\mathcal{I}}$ with itself. If $d \in C^{\mathcal{I}}$, we say that d is an *instance of* C in $\mathcal{I}$. If $(d,e) \in R^{\mathcal{I}}$, we say that d is an R-predecessor of e, and e is an R-successor of d in $\mathcal{I}$.

A concept C is called *satisfiable* iff there is some interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$. We call such an interpretation a *model of* C. A concept D *subsumes* a concept C (written $C \sqsubseteq D$) iff for all interpretations $\mathcal{I}$ we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Additional Boolean operators, such as implication, will be used as abbreviations: for example, $A \Rightarrow B$ stands for $\neg A \sqcup B$. Furthermore, we can express all relations in $\{=,<,>\}$ inside number restrictions, for example $(> n\ R) \equiv \neg(\leq n\ R)$ and $(= n\ R) \equiv ((\leq n\ R) \sqcap (\geq n\ R))$.

If a Description Logic allows for negation and conjunction of concepts, subsumption and (un)satisfiability can be reduced to each other:

- $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable,
- C is unsatisfiable iff $C \sqsubseteq A \sqcap \neg A$ (for a concept name A).

Since all but one Description Logic considered here are in fact propositionally closed, this connection between satisfiability and subsumption will be heavily exploited: we restrict our attention to one of the two inference problems, namely satisfiability, both in the decidability and in the undecidability proofs.

3 Number Restrictions on Complex Roles

In this section, we introduce extensions of $\mathcal{ALCN}$, $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}$, and $\mathcal{ALC}_{+}\mathcal{N}$ with number restrictions on complex roles and investigate the complexity of the corresponding inference problems. This investigation yields an almost complete classification of the extensions of $\mathcal{ALCN}$ by different kinds of complex roles in number restrictions. Furthermore, it turns out that it suffices to extend $\mathcal{ALC}_{+}\mathcal{N}$ with number restrictions on role chains (that is to allow for number restrictions with composition) to obtain undecidability.

To simplify the presentation of our results, we start by giving a scheme of how to build extensions of $\mathcal{ALCN}$, $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}$, and $\mathcal{ALC}_{+}\mathcal{N}$ with number restrictions on complex roles. The name of such an extension consists of the name of the base logic followed by the set of role constructors that are allowed inside number restrictions.

Definition 3 For a set $M \subseteq \{\sqcup, \sqcap, \circ, \ ^{-1}\}$ of role constructors and a complex role R, we call a number restriction of the form $(\geq\ n\ \ R)$ or $(\leq\ n\ \ R)$ an *M-number restriction* iff R is built using only constructors from M. The set of *$\mathcal{ALCN}(M)$-concepts* (resp. *$\mathcal{ALC}_{+}\mathcal{N}(M)$-concepts* and *$\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(M)$-concepts*) is obtained from $\mathcal{ALC}$-concepts (resp. $\mathcal{ALC}_{+}$- and $\mathcal{ALC}_{\mathsf{reg}}$-concepts) by additionally allowing for M-number restrictions.

Composition is present in all extensions investigated in this paper for the following reasons. On the one hand, composition in number restrictions strongly increases the expressive power: it allows one to restrict the number of role-*chain*-successors. The expressiveness of this extension even leads to the loss of the tree-model property, a property satisfied by most of the Description Logics considered in the literature. For example, the concept $(\geq\ 2\ R) \sqcap (\forall R.\exists S.A) \sqcap (\leq\ 1\ R \circ S)$ is obviously satisfiable, but each of its instances has two R-successors having a common S-successor. Thus, models of this concept cannot be tree-models. On the other hand, decidability of satisfiability and subsumption for $\mathcal{ALCN}(M)$ for sets $M \subseteq \{\sqcup, \sqcap, \ ^{-1}\}$ follows immediately from a result in [11]; this result is discussed in more detail in Section 5.

The examples introduced in Section 1 should provide an intuition of what can be expressed using complex roles inside number restrictions. To obtain a deeper insight into the expressive power of Description Logics with complex number restriction, we first show the undecidability results.

3.1 Undecidable Extensions

We will use a reduction of the domino problem—a well-known undecidable problem [14, 4] often used in undecidability proofs in logic—to show that concept satisfiability is undecidable for the three extensions $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$, $\mathcal{ALCN}(\circ, \sqcap)$, and $\mathcal{ALC}_+\mathcal{N}(\circ)$ of the decidable logic $\mathcal{ALCN}(\circ)$ considered in the next subsection. For didactic reasons, we will also consider the logics $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(\circ, \sqcup)$ and $\mathcal{ALC}_+\mathcal{N}(\circ, \sqcup)$, although their undecidability follows from the other results.

Definition 4 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 $m, n \in \mathbb{N}$,

$$\begin{array}{rcl} (t(m,n), t(m+1,n)) & \in & H \text{ and} \\ (t(m,n), t(m,n+1)) & \in & V. \end{array}$$

The standard domino problem asks for a compatible tiling of the whole plane. However, a compatible tiling of the first quadrant yields compatible tilings of arbitrarily large finite rectangles, which in turn yield a compatible tiling of the plane [14]. Thus, the undecidability result for the standard problem [4] carries over to this variant.

In order to reduce the domino problem to satisfiability of concepts, we must show how a given tiling system $\mathcal{D}$ can be translated into a concept $E_\mathcal{D}$ (of the logic under consideration) such that $E_\mathcal{D}$ is satisfiable iff $\mathcal{D}$ allows for a compatible tiling. This task can be split into three subtasks, which we will first explain on an intuitive level, before showing how they can be achieved for the five Description Logics under consideration.

Task 1: It must be possible to represent a single "square" of $\mathbb{N} \times \mathbb{N}$, which consists of points $(n, m), (n, m+1), (n+1, m)$, and $(n+1, m+1)$. The idea is to introduce roles X, Y, where X goes one step into the horizontal (i.e. x-) direction, and Y goes one step into the vertical (i.e. y-) direction. The Description Logic must be expressive enough to describe that an individual (a point (n, m)) has exactly one X-successor (the point $(n+1, m)$), exactly one Y-successor (the point $(n, m+1)$), and that the $X \circ Y$-successor coincides with the $Y \circ X$-successor (the point $(n+1, m+1)$).

Task 2: It must be possible to express that a tiling is locally compatible, i.e., that the X- and Y-successors 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 on the roles X, Y.

Task 3: 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 U and a "start" individual such that every point is a U-successor of this start individual. The local conditions can then be imposed on all points via value restrictions on U for the start individual.

Task 2 is rather easy, and can be realized using the $\mathcal{ALC}$-concept $C_\mathcal{D}$ given in Figure 1. The first conjunct expresses that every point has exactly one domino type, and the value restrictions in the second conjunct express the horizontal and vertical matching conditions.

Task 1 can be achieved in any extension of $\mathcal{ALCN}(\circ)$ with either union or intersection of roles in number restrictions: see the concepts $C_\sqcap$ and $C_\sqcup$ in Figure 1.

$$
\begin{array}{ll}
C_{\mathcal{D}} := & \bigsqcup_{1\leq i\leq m}(D_i \sqcap (\bigsqcap_{\substack{1\leq j\leq m\\ i\neq j}} \neg D_j)) \sqcap \\
& \bigsqcap_{1\leq i\leq m}(D_i \Rightarrow ((\forall X.(\bigsqcup_{(D_i,D_j)\in H} D_j)) \sqcap (\forall Y.(\bigsqcup_{(D_i,D_j)\in V} D_j)))) \\
C_{\sqcup} := & (=1\ X) \sqcap (=1\ Y) \sqcap (=1\ X\circ Y) \sqcap (=1\ Y\circ X) \sqcap (=1\ Y\circ X \sqcup X\circ Y) \\
C_{\sqcap} := & (=1\ X) \sqcap (=1\ Y) \sqcap (=1\ X\circ Y) \sqcap (=1\ Y\circ X) \sqcap (=1\ Y\circ X \sqcap X\circ Y)
\end{array}
$$

$$
E_{\mathcal{D}}^{(1')} := (=1\ R) \sqcap (\forall R^+.(C_{\sqcup} \sqcap C_{\mathcal{D}} \sqcap (\geq 2\ R) \sqcap (\leq 2\ R \sqcup X \sqcup Y)))
$$

$$
\begin{array}{ll}
E_{\mathcal{D}}^{(2)} := & (\geq 1\ U) \sqcap (\forall U.\ (C_{\sqcup} \sqcap C_{\mathcal{D}} \sqcap (=1\ X\circ U^{-1}) \sqcap (=1\ Y\circ U^{-1}) \sqcap \\
& \qquad (\leq 1\ U^{-1} \sqcup Y\circ U^{-1} \sqcup X\circ U^{-1}))) \\
E_{\mathcal{D}}^{(3)} := & (=1\ R) \sqcap (=1\ R \sqcap R\circ T\circ R) \sqcap \\
& (\forall R.\forall T.\forall R.\ (C_{\sqcap} \sqcap C_{\mathcal{D}} \sqcap (\leq 1\ T) \sqcap (\forall Y.(\leq 1\ T)) \sqcap (\forall X.(\leq 1\ T)) \sqcap \\
& \qquad (=1\ T \sqcap X\circ T \sqcap Y\circ T) \sqcap \\
& \qquad (=1\ X \sqcap X\circ T\circ R) \sqcap (=1\ Y \sqcap Y\circ T\circ R)))
\end{array}
$$

where $A \Rightarrow B$ is an abbreviation for $\neg A \sqcup B$ and
$(= n\ R)$ is an abbreviation for $(\geq n\ R) \sqcap (\leq n\ R)$.

Figure 1: Concepts used in the proof of Theorem 5

Task 3 is easy for logics that extend $\mathcal{ALC}_+$, and more difficult for logics without transitive closure. The general idea is that the start individual s is an instance of the concept $E_{\mathcal{D}}$ to be constructed. From this individual, one can reach via U the origin $(0,0)$ of $\mathbb{N} \times \mathbb{N}$ and all points that are connected with the origin via arbitrary X- and Y-paths.

With this intuition in mind, the reduction concepts that achieve Task 3 are now explained in detail for each undecidable extension of $\mathcal{ALC}$, $\mathcal{ALC}_+$, and $\mathcal{ALC}_{\mathsf{reg}}$ by complex number restrictions.

$\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(\circ,\sqcup)$: We start with an extension of $\mathcal{ALC}_{\mathsf{reg}}$ since here it is rather easy to reach, from the start individual, all individuals representing points in the plane. In fact, in extensions of $\mathcal{ALC}_{\mathsf{reg}}$, we can use the complex role $(X \sqcup Y)^+$ to reach every point accessible from the origin $(0,0)$ via arbitrary X- and Y-paths. Thus, for each tiling system $\mathcal{D}$, the $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(\circ,\sqcup)$-concept

$$
E_{\mathcal{D}}^{(1)} := (=1\ R) \sqcap (\forall (R \sqcup (R\circ (X \sqcup Y)^+)).(C_{\sqcup} \sqcap C_{\mathcal{D}})).
$$

can be constructed, which is obviously satisfiable if, and only if, $\mathcal{D}$ admits a compatible tiling.

$\mathcal{ALC}_+\mathcal{N}(\circ,\sqcup)$: The complex role in the value restriction can even be restricted to a simple transitive closure of an atomic role. Intuitively, a starting point outside the plane is used which is connected to each point in the plane via some R-path. To achieve this, the concept $E_{\mathcal{D}}^{(1')}$ in Figure 1 makes sure that the X- and the Y-successors of each point in the plane are also R-successors of this point. Hence R^+ can be used in place of $(X \sqcup Y)^+$ as "universal" role, and thus the concept $E_{\mathcal{D}}^{(1')}$ is in $\mathcal{ALC}_+\mathcal{N}(\circ,\sqcup)$.

$\mathcal{ALCN}(\circ, \sqcup, ^{-1})$: In $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$, a role name U for the "universal" role is explicitly introduced, and number restrictions involving composition, union, and inversion of roles are used to make sure that the start individual is directly connected via U with every point: see the concept $E_{\mathcal{D}}^{(2)}$ in Figure 1 and the left diagram in Figure 2. The number restrictions inside the value restriction make sure that every point p that is reached via U from the start individual satisfies the following: Its X-successor and its Y-successor each have exactly one U-predecessor, which coincides with the (unique) U-predecessor of p, i.e., the start individual. Thus, the X-successor and the Y-successor of p are also U-successors of the start individual.

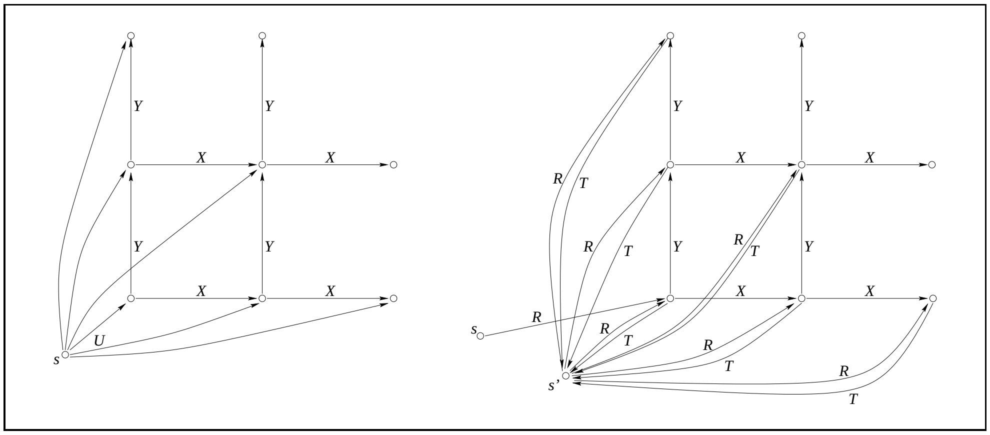

Figure 2: The universal role for $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$ and $\mathcal{ALCN}(\circ, \sqcap)$

$\mathcal{ALCN}(\circ, \sqcap)$: For $\mathcal{ALCN}(\circ, \sqcap)$, a similar construction is possible. Since inversion of roles is not allowed in $\mathcal{ALCN}(\circ, \sqcap)$, two role names R and T are needed for the construction of the universal role. The intuition is that T plays the rôle of the inverse of R (except for one individual), and the "universal" role corresponds to the composition $R \circ T \circ R$; see the right diagram in Figure 2. The start individual s (which is an instance of $E_{\mathcal{D}}^{(3)}$), has exactly one R-successor $p_{(0,0)}$, which coincides with its $R \circ T \circ R$-successor. The individual $p_{(0,0)}$ corresponds to the origin of $\mathbb{N} \times \mathbb{N}$. The number restrictions of $E_{\mathcal{D}}^{(3)}$ make sure that $p_{(0,0)}$ satisfies the following: It has exactly one T-successor, call it s', which coincides with the $R \circ T$-successor of s, and with the (unique) T-successors of the X- and Y-successors of $p_{(0,0)}$. In addition, the (unique) X-successor of $p_{(0,0)}$ is also an $X \circ T \circ R$-successor of $p_{(0,0)}$, which makes sure that the X-successor of $p_{(0,0)}$ is an R-successor of s', and thus an $R \circ T \circ R$-successor of s. The same holds for the Y-successor. One can now continue the argument with the X-successor (resp. Y-successor) of $p_{(0,0)}$ in place of $p_{(0,0)}$.

With the intuitions given above, it is not hard to show for all $i, 1 \leq i \leq 3$, that a tiling system $\mathcal{D}$ has a compatible tiling iff $E_{\mathcal{D}}^{(i)}$ is satisfiable, and that the same is true for $E_{\mathcal{D}}^{(1')}$.

Theorem 5 Satisfiability (and thus also subsumption) of concepts is undecidable for $\mathcal{ALC}_{+}\mathcal{N}(\circ, \sqcup)$, $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$, and $\mathcal{ALCN}(\circ, \sqcap)$.

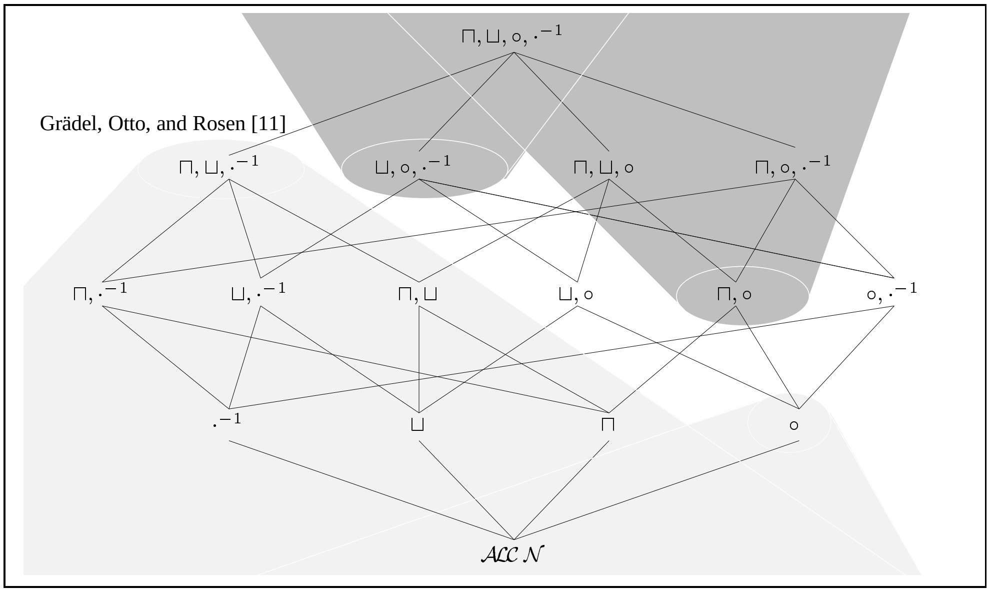

Figure 3: (Un)decidability results for extensions of $\mathcal{ALCN}$.

This theorem does not explicitly mention the undecidability result for $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(\circ, \sqcup)$, since it is an immediate consequence of the result for $\mathcal{ALC}_{+}\mathcal{N}(\circ, \sqcup)$.

Figure 3 gives an overview of the (un)decidability results for extensions of $\mathcal{ALCN}$ by complex roles in number restrictions. Decidable extensions are light grey, whereas undecidable ones are dark grey. The overview shows the results from Theorem 5 together with the decidability results that follow from [11] and the decidability result that will be shown in the next section. The only problems that remain open for the extensions of $\mathcal{ALCN}$ concern $\mathcal{ALCN}(\circ, ^{-1})$ and $\mathcal{ALCN}(\circ, \sqcup)$. Until now, neither a decision procedure for one of these extensions nor a proof of their undecidability could be found.

To make the picture more complete, we will now focus on extensions of $\mathcal{ALC}_{+}\mathcal{N}$. So far, only $\mathcal{ALC}_{+}\mathcal{N}(\circ, \sqcup)$ was shown to be undecidable. It will now be shown that, in extensions of $\mathcal{ALC}_{+}\mathcal{N}$, it suffices to allow for composition in number restriction in order to lose decidability (see Figure 6 for an overview of the (un)decidability results for extensions of $\mathcal{ALC}_{+}\mathcal{N}$ by number restrictions on complex roles). Again, a reduction of the domino problem to concept satisfiability is used to show undecidability of $\mathcal{ALC}_{+}\mathcal{N}(\circ)$. Since this reduction is rather different from the ones above and more complicated, it is treated separately. The (redundant) reduction for $\mathcal{ALC}_{+}\mathcal{N}(\circ, \sqcup)$ was given since it served to give the intuition for $\mathcal{ALCN}(\circ, \sqcup, ^{-1})$ and $\mathcal{ALCN}(\circ, \sqcap)$. The concepts used for the reduction of the domino problem to $\mathcal{ALC}_{+}\mathcal{N}(\circ)$-concept satisfiability are given in Figure 4.

The concept C_{prim} makes sure that each point will be an instance of either A or B or C (which are disjoint), and that with each point exactly one domino type D_i will be associated.

Task 1 is achieved via the concept $C_{\boxplus}$, which describes a square by using a single role X. Each instance of $C_{\boxplus}$ has two X-successors that in turn each have two X-successors. The conjunct $(= \ 3 \ \ X \circ X)$ makes sure that the X-successors of an instance of $C_{\boxplus}$ have one

$$
\begin{aligned}
C_{\boxplus} &:= (=2\ X) \sqcap (\forall X.(=2\ X)) \sqcap (=3\ X \circ X) \\
C_{\text{prim}} &:= (A \sqcup B \sqcup C) \sqcap \bigsqcup_{1 \le i \le m} (D_i \sqcap (\bigsqcap_{\substack{1 \le j \le m \\ i \neq j}} \neg D_j)) \\
C_{\text{diag}} &:= (A \Rightarrow ((\exists X.B) \sqcap (\exists X.C))) \sqcap \\
&\quad\ (B \Rightarrow ((\exists X.A) \sqcap (\exists X.C))) \sqcap \\
&\quad\ (C \Rightarrow ((\exists X.A) \sqcap (\exists X.B))) \\
C_{\mathcal{D}} &:= \\
&\bigsqcap_{1 \le i \le m} \big(((A \sqcap D_i) \Rightarrow (\exists X.(C \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists X.(B \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))) \sqcap \\
&\qquad ((B \sqcap D_i) \Rightarrow (\exists X.(A \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists X.(C \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j)))) \sqcap \\
&\qquad ((C \sqcap D_i) \Rightarrow (\exists X.(B \sqcap (\bigsqcup_{(D_i,D_j) \in H} D_j)) \sqcap \exists X.(A \sqcap (\bigsqcup_{(D_i,D_j) \in V} D_j))))\big) \\
E_{\mathcal{D}}^{(4)} &:= (=1\ X) \sqcap (\exists X.A) \sqcap (\forall X^+.(C_{\boxplus} \sqcap C_{\text{prim}} \sqcap C_{\text{diag}} \sqcap C_{\mathcal{D}}))
\end{aligned}
$$

where A, B and C are disjoint concepts since they are abbreviations for
$A := A_1, \qquad B := \neg A_1 \sqcap A_2 \qquad C := \neg A_1 \sqcap \neg A_2$

Figure 4: Concepts used in the proof of Theorem 6

common X-successor.

Task 3 is easy because $\mathcal{ALC}_+\mathcal{N}(\circ)$ allows for the transitive closure of roles. If s is an instance of $E_{\mathcal{D}}^{(4)\mathcal{I}}$, then s has exactly one X-successor, say $p_{(0,0)}$, which is an instance of A. Each point in the grid is an X^n-successor of s. Thus, the local conditions on all points in the grid are imposed by $\forall X^+(C_{\boxplus} \sqcap C_{\text{prim}} \sqcap C_{\text{diag}} \sqcap C_{\mathcal{D}})$.

Task 2 is difficult because we must distinguish between the "horizontal" and the "vertical" X-successor of a point. For this purpose, the concepts A, B, and C are used in the following way (see Figure 5 for a better intuition).

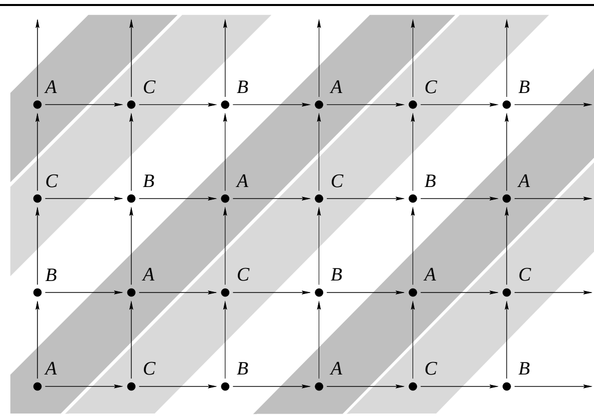

Figure 5: Visualisation of the grid as enforced by the $\mathcal{ALC}_+\mathcal{N}(\circ)$ reduction concept.

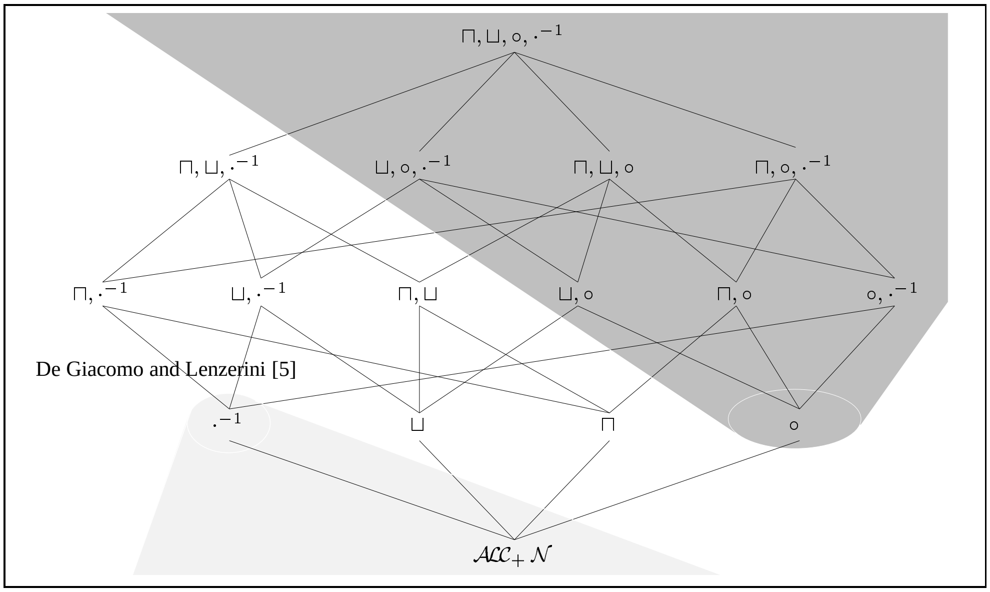

Figure 6: (Un)decidability results for extensions of $\mathcal{ALC}_+$.

The concept C_{diag} makes sure that each instance of A has one X-successor in B and one in C, and similar for instances of B and C. Without loss of generality, we draw the X-successor of $p_{0,0}$ that is in C to its right and call it $p_{1,0}$. The other X-successor of $p_{0,0}$, which is in B, is called $p_{0,1}$ and is drawn above it. Now, it is easy to see that the remaining parts of the grid are determined in the sense that

- for each diagonal in the grid there is an $E \in \{A, B, C\}$ such that all points on this diagonal are instances of E,
- horizontal successors of points in A are always in C, of points in C are always in B, and of points in B are always in A,
- vertical successors of points in A are always in B, of points in B are always in C, and of points in C are always in A.

With the intuitions given above, it is not hard to show that a tiling system $\mathcal{D}$ has a compatible tiling iff $E_{\mathcal{D}}^{(4)}$ is satisfiable,[2]

Theorem 6 Satisfiability (and thus also subsumption) of concepts is undecidable for $\mathcal{ALC}_+\mathcal{N}(\circ)$.

[2]To make the reduction more obvious, the concept $E_{\mathcal{D}}^{(4)}$ is longer than necessary. In fact, the subconcept C_{diag} could have been left out.

3.2 $\mathcal{ALCN}(\circ)$ is decidable

We present a tableau-like algorithm for deciding satisfiability of $\mathcal{ALCN}(\circ)$-concepts. The algorithm and the proof of its correctness are very similar to existing algorithms and proofs for $\mathcal{ALC}$ with number restrictions on atomic roles [13, 12]. These proofs heavily employ the fact that each satisfiable $\mathcal{ALCN}$-concept has a tree-model.[3] It can easily be seen that, in contrast to $\mathcal{ALCN}$, the logic $\mathcal{ALCN}(\circ)$ does not have the tree-model property. For example, the concept

$$(\geq 2\ R) \sqcap (\forall R.\exists S.A) \sqcap (\leq 1\ R{\circ}S)$$

is obviously satisfiable, but each of its instances has an $R{\circ}S$ successor d that is reachable via two different paths. In particular, d has two different role predecessors.

Nevertheless, the models that will be generated by our algorithm are very similar to tree-models in that every element of the model can be reached from an initial (root) element via role chains, the root does not have a role predecessor, and every role chain from the root to an element has the same length (even though there may exist more than one such chain). In the proof of the termination of the algorithm, this fact will be used in the place of the tree-model property.

As usual [24], we assume without loss of generality that all concepts are in negation normal form (NNF), i.e., negation occurs only immediately in front of atomic concepts. The basic data structure our algorithm works on are constraints:

Definition 7 Let $\tau = \{x, y, z, \ldots\}$ be a countably infinite set of individual variables. A *constraint* is either of the form

xRy, where R is a role name in N_R and $x, y \in \tau$,

$x : D$ for some $\mathcal{ALCN}(\circ)$-concept D in NNF and some $x \in \tau$, or

$x \neq y$ for $x, y \in \tau$.[4]

A *constraint system* is a set of constraints. For a constraint system S, let $\tau_S \subseteq \tau$ denote the individual variables occurring in S.

An interpretation $\mathcal{I}$ is a *model of a constraint system* S iff there is a mapping $\pi : \tau_S \to \Delta^{\mathcal{I}}$ such that $\mathcal{I}, \pi$ satisfy each constraint in S, i.e.,

$$\begin{aligned} (\pi(x), \pi(y)) &\in R^{\mathcal{I}} && \text{for all } xRy \in S, \\ \pi(x) &\neq \pi(y) && \text{for all } (x \neq y) \in S, \\ \pi(x) &\in D^{\mathcal{I}} && \text{for all } x : D \in S. \end{aligned}$$

For a constraint system S, individual variables x, y, and role names R_i, we say that y is an $R_1 \circ \ldots \circ R_m$*-successor* of x in S iff there are $y_0, \ldots, y_m \in \tau$ such that $x = y_0, y = y_m$, and $\{y_i R_{i+1} y_{i+1} \mid 0 \leq i \leq m-1\} \subseteq S$. The system S contains a *clash* iff $\{x : A, x : \neg A\} \subseteq S$ for some concept name A and some variable $x \in \tau_S$, or $x : (\leq n\ R) \in S$ and x has $\ell > n$ R-successors $y_1, \ldots, y_\ell$ in S such that for all $i \neq j$ we have $y_i \neq y_j \in S$. A constraint system S is called *complete* iff none of the completion rules given in Figure 7 can be applied to S.

[3]A tree-model is a model having the shape of a tree, i.e., it has a root, which does not have role predecessors, and every other element of the model has exactly one role predecessor. In particular, there are no cyclic role chains in the model.

[4]We consider such inequalities as being symmetric, i.e., if $x \neq y$ belongs to a constraint system, then $y \neq x$ (implicitly) belongs to it as well.

1. Conjunction: If $x:(C_1 \sqcap C_2) \in S$ and $x:C_1 \notin S$ or $x:C_2 \notin S$, then
$S \rightarrow \quad S \cup \{x:C_1, x:C_2\}$

2. Disjunction: If $x:(C_1 \sqcup C_2) \in S$ and $x:C_1 \notin S$ and $x:C_2 \notin S$, then
$S \rightarrow \quad S_1 = S \cup \{x:C_1\}$
$S \rightarrow \quad S_2 = S \cup \{x:C_2\}$

3. Value restriction: If $x:(\forall R.C) \in S$ for a role name R, y is an R-successor of x in S and $y:C \notin S$, then
$S \rightarrow \quad S \cup \{y:C\}$

4. Existential restriction: If $x:(\exists R.C) \in S$ for a role name R and there is no R-successor y of x in S with $y : C \in S$, then
$S \rightarrow \quad S \cup \{xRz, z : C\}$ for a new variable $z \in \tau \setminus \tau_S$.

5. Number restriction: If $x:(\geq n\ R_1 \circ \ldots \circ R_m) \in S$ for role names $R_1, \ldots, R_m$ and x has less than $n\ R_1 \circ \ldots \circ R_m$-successors in S, then
$S \rightarrow \quad S \cup \{xR_1y_2, y_mR_mz\} \cup \{y_iR_iy_{i+1} \mid 2 \leq i \leq m-1\} \cup$
$\{z \neq w \mid w$ is an $R_1 \circ \ldots \circ R_m$-successor of x in $S\}$
where z, y_i are new variables in $\tau \setminus \tau_S$.

6. Number restriction: If $x:(\leq n\ R_1 \circ \ldots \circ R_m) \in S$, x has more than n $R_1 \circ \ldots \circ R_m$-successors in S, and there are $R_1 \circ \ldots \circ R_m$-successors y_1, y_2 of x in S with $(y_1 \neq y_2) \notin S$, then
$S \rightarrow S_{y_1,y_2} = S[y_2/y_1]$
for all pairs y_1, y_2 of $R_1 \circ \ldots \circ R_m$-successors of x with $(y_1 \neq y_2) \notin S$.

Figure 7: The completion rules for $\mathcal{ALCN}(\circ)$

Figure 7 introduces the *completion rules* that are used to test $\mathcal{ALCN}(\circ)$-concepts for satisfiability. In these rules, the constraint system $S[y_2/y_1]$ is obtained from S by substituting each occurrence of y_2 in S by y_1.

The *completion algorithm* works on a tree where each node is labelled with a constraint system. It starts with the tree consisting of a root labelled with $S = \{x_0 : C_0\}$, where C_0 is the $\mathcal{ALCN}(\circ)$-concept in NNF to be tested for satisfiability. A rule can only be applied to a leaf labelled with a clash-free constraint system. Applying a rule $S \rightarrow S_i$, for $1 \leq i \leq n$, to such a leaf leads to the creation of n new successors of this node, each labelled with one of the constraint systems S_i. The algorithm terminates if none of the rules can be applied to any of the leaves. In this situation, it answers with "C_0 is satisfiable" iff one of the leaves is labelled with a clash-free constraint system.

Soundness and completeness of this algorithm is an immediate consequence of the following facts:

Lemma 8 Let C_0 be an $\mathcal{ALCN}(\circ)$-concept in NNF, and let S be a constraint system obtained by applying the completion rules to $\{x_0 : C_0\}$. Then

1. For each completion rule $\mathcal{R}$ that can be applied to S and for each interpretation $\mathcal{I}$, the following equivalence holds: $\mathcal{I}$ is a model of S iff $\mathcal{I}$ is a model of one of the systems S_i obtained by applying $\mathcal{R}$.

2. If S is a complete and clash-free constraint system, then S has a model.

3. If S contains a clash, then S does not have a model.

4. The completion algorithm terminates when applied to $\{x_0 : C_0\}$.

Indeed, termination shows that after finitely many steps we obtain a tree such that all its leaf nodes are labelled with complete constraint systems. If C_0 is satisfiable, then $\{x_0 : C_0\}$ is also satisfiable, and thus one of the complete constraint systems is satisfiable by (1). By (3), this system must be clash-free. Conversely, if one of the complete constraint systems is clash-free, then it is satisfiable by (2), and because of (1) this implies that $\{x_0 : C_0\}$ is satisfiable. Consequently, the algorithm is a decision procedure for satisfiability of $\mathcal{ALCN}(\circ)$-concepts:

Theorem 9 Subsumption and satisfiability of $\mathcal{ALCN}(\circ)$-concepts is decidable.

Proof of Part 1 of Lemma 8: We consider only the rules concerned with number restrictions, since the proof for Rules 1–4 is just as for $\mathcal{ALC}$.

5. Number restriction: Assume that the rule is applied to the constraint $x : (\geq n\ R_1 \circ \ldots \circ R_m)$, and that its application yields

$$S' = S \cup \{xR_1y_2, y_mR_mz\} \cup \{y_iR_iy_{i+1} \mid 2 \leq i \leq m-1\} \\ \cup \{z \neq w \mid w \text{ is an } R_1 \circ \ldots \circ R_m\text{-successor of } x \text{ in } S\}.$$

Since S is a subset of S', any model of S' is also a model of S.
Conversely, assume that $\mathcal{I}$ is a model of S, and let $\pi : \tau_S \to \Delta^{\mathcal{I}}$ be the corresponding mapping of individual variables to elements of $\Delta^{\mathcal{I}}$. On the one hand, since $\mathcal{I}$ satisfies $x : (\geq n\ R_1 \circ \ldots \circ R_m)$, $\pi(x)$ has at least n $R_1 \circ \ldots \circ R_m$-successors in $\mathcal{I}$. On the other hand, since Rule 5 is applicable to $x : (\geq n\ R_1 \circ \ldots \circ R_m)$, x has less than n $R_1 \circ \ldots \circ R_m$-successors in S. Thus, there exists an $R_1 \circ \ldots \circ R_m$-successor b of $\pi(x)$ in $\mathcal{I}$ such that $b \neq \pi(w)$ for all $R_1 \circ \ldots \circ R_m$-successors w of x in S. Let $b_2, \ldots, b_m \in \Delta^{\mathcal{I}}$ be such that $(\pi(x), b_2) \in R_1^{\mathcal{I}}, (b_2, b_3) \in R_2^{\mathcal{I}}, \ldots, (b_m, b) \in R_m^{\mathcal{I}}$. We define $\pi' : \tau_{S'} \to \Delta^{\mathcal{I}}$ by $\pi'(y) := \pi(y)$ for all $y \in \tau_S$, $\pi'(y_i) := b_i$ for all $i, 2 \leq i \leq m$, and $\pi'(z) := b$. Obviously, $\mathcal{I}, \pi'$ satisfy S'.

6. Number restriction: Assume that the rule can be applied to $x : (\leq n\ R_1 \circ \ldots \circ R_m) \in S$, and let $\mathcal{I}$ together with the valuation $\pi : \tau_S \to \Delta^{\mathcal{I}}$ be a model of S. On the one hand, since the rule is applicable, x has more than n $R_1 \circ \ldots \circ R_m$-successors in S. On the other hand, $\mathcal{I}, \pi$ satisfy $x : (\leq m\ R_1 \circ \ldots \circ R_m) \in S$, and thus there are two different $R_1 \circ \ldots \circ R_m$-successors y_1, y_2 of x in S such that $\pi(y_1) = \pi(y_2)$. Obviously, this implies that $(y_1 \neq y_2) \notin S$, which shows that $S_{y_1,y_2} = S[y_2/y_1]$ is one of the constraint systems obtained by applying Rule 6 to $x : (\leq n\ R_1 \circ \ldots \circ R_m)$. In addition, since $\pi(y_1) = \pi(y_2)$, $\mathcal{I}, \pi$ satisfy S_{y_1,y_2}.
Conversely, assume that $S_{y_1,y_2} = S[y_2/y_1]$ is obtained from S by applying Rule 6, and let $\mathcal{I}$ together with the valuation π be a model of S_{y_1,y_2}. If we take a valuation π' that coincides with π on the variables in $\tau_{S_{y_1,y_2}}$ and satisfies $\pi'(y_2) = \pi(y_1)$, then $\mathcal{I}, \pi'$ obviously satisfy S.

Proof of Part 2 of Lemma 8: Let S be a complete and clash-free constraint system that is obtained by applying the completion rules to $\{x_0 : C_0\}$. We define a canonical model $\mathcal{I}$ of S as follows:

$$\begin{array}{rrl} & \Delta^{\mathcal{I}} := \tau_S & \text{and} \\ \text{for all } A \in N_C: & x \in A^{\mathcal{I}} & \text{iff } x : A \in S, \\ \text{for all } R \in N_R: & (x,y) \in R^{\mathcal{I}} & \text{iff } xRy \in S. \end{array}$$

In addition, let $\pi : \tau_S \to \Delta^{\mathcal{I}}$ be the identity on τ_S. We show that $\mathcal{I}, \pi$ satisfy every constraint in S.

By definition of $\mathcal{I}$, a role constraint of the form xRy is satisfied by $\mathcal{I}, \pi$ iff $xRy \in S$. More generally, y is an $R_1 \circ \ldots \circ R_m$-successor of x in S iff y is an $R_1 \circ \ldots \circ R_m$-successor of x in $\mathcal{I}$. We show by induction on the structure of the concept C that every concept constraint $x : C \in S$ is satisfied by $\mathcal{I}, \pi$. Again, we restrict our attention to number restrictions since the induction base and the treatment of the other constructors is just as for $\mathcal{ALC}$.

- Consider $x : (\geq n\ R_1 \circ \ldots \circ R_m) \in S$. Since S is complete, Rule 5 cannot be applied to $x : (\geq n\ R_1 \circ \ldots \circ R_m)$, and thus x has at least n $R_1 \circ \ldots \circ R_m$-successors in S, which are also $R_1 \circ \ldots \circ R_m$-successors of x in $\mathcal{I}$. This shows that $\mathcal{I}, \pi$ satisfy $x : (\geq n\ R_1 \circ \ldots \circ R_m)$.

- Constraints of the form $x : (\leq n\ R_1 \circ \ldots \circ R_m) \in S$ are satisfied because S is clash-free and complete. In fact, assume that x has more than n $R_1 \circ \ldots \circ R_m$-successors in $\mathcal{I}$. Then x also has more than n $R_1 \circ \ldots \circ R_m$-successors in S. If S contained inequality constraints $y_i \neq y_j$ for all these successors, then we would have a clash. Otherwise, Rule 6 could be applied.

Proof of Part 3 of Lemma 8: Assume that S contains a clash. If $\{x : A, x : \neg A\} \subseteq S$, then it is clear that no interpretation can satisfy both constraints. Thus assume that $x : (\leq n\ R) \in S$ and x has $\ell > n$ R-successors $y_1, \ldots, y_\ell$ in S with $(y_i \neq y_j) \in S$ for all $i \neq j$. Obviously, this implies that in any model $\mathcal{I}, \pi$ of S, $\pi(x)$ has $\ell > n$ distinct R-successors $\pi(y_1), \ldots, \pi(y_\ell)$ in $\mathcal{I}$, which shows that $\mathcal{I}, \pi$ cannot satisfy $x : (\leq n\ R)$.

Proof of Part 4 of Lemma 8: The detailed proof can be found in the appendix. For this proof, the following observations, which are an easy consequence of the definition of the completion rules, are important:

Lemma 10 Let C_0 be an $\mathcal{ALCN}(\circ)$-concept in NNF, and let S be a constraint system obtained by applying the completion rules to $\{x_0 : C_0\}$.

1. Every variable $x \neq x_0$ that occurs in S is an $R_1 \circ \ldots \circ R_m$-successor of x_0 for some role chain of length $m \geq 1$. In addition, every other role chain that connects x_0 with x has the same length.

2. If x can be reached in S by a role chain of length m from x_0, then for each constraint $x : C$ in S, the maximal role depth[5] of C is bounded by the maximal role depth of C_0 minus m. Consequently, m is bounded by the maximal role depth of C_0.

[5]The role depth is formally defined in the appendix. Intuitively, it is the depth of nested role "expressions" in value restrictions and number restrictions.

Intuitively, these two facts are used as follows. Let m_0 be the maximal role depth of C_0. Because of the first fact, every individual x in a constraint system S (reached from $\{x_0 : C_0\}$ by applying completion rules) has a unique role level *level*(x), which is its distance from the root node x_0, i.e., the unique length of the role chains that connect x_0 with x. Because of the second fact, the level of each individual is an integer between 0 and m_0. Both facts together imply that the length of role chains is bounded by m_0. Since the number of direct role successors of a given individual can also be bounded by the size of C_0, this implies that the size of the constraint systems that can be built by the completion algorithm is bounded. A formal proof of termination based on an explicit termination ordering is given in the appendix.

Discussion of the result: For logics where number restrictions may contain—in addition to composition—union or intersection of roles, an important property used in the above termination proof is no longer satisfied. It is not possible to associate each individual generated by a tableau-like procedure with a unique role level, which is its distance from the "root" individual x_0 (i.e., the instance x_0 of C_0 to be generated by the tableau algorithm). Indeed, in the concept

$$C_0 := (\exists R.\exists R.A) \sqcap (\leq 1\ R \sqcup R \circ R),$$

the number restriction enforces that an R-successor of an instance of C_0 is also an $R \circ R$-successor of this instance. For this reason, an R-successor of the root individual must be both on level 1 and on level 2, and thus the relatively simple termination argument that was used above is not available for these larger logics. However, as we will show below, this termination argument can still be used if union and intersection are restricted to role chains of the same length. Without this restriction, satisfiability may become undecidable: in Section 3.1 we have shown that satisfiability is in fact undecidable for $\mathcal{ALCN}(\circ, \sqcap)$. For $\mathcal{ALCN}(\circ, \sqcup)$, decidability of satisfiability is still an open problem.

3.3 An extension of the decidability result

The algorithm given in Section 3.2 will be extended such that it can also treat union and intersection of role chains that have the same length. The proof of soundness, completeness and termination of this extended algorithm is very similar to the one for the basic algorithm, and will thus only be sketched.

In the remainder of this section, a *complex role* is

- a role chain $\mathcal{R} = R_1 \circ \ldots \circ R_n$, or
- an intersection $\mathcal{R} = R_1 \circ \ldots \circ R_n \sqcap S_1 \circ \ldots \circ S_n$ of two role chains of the same length, or
- a union $\mathcal{R} = R_1 \circ \ldots \circ R_n \sqcup S_1 \circ \ldots \circ S_n$ of two role chains of the same length.

The satisfiability algorithm is extended by adding two new rules to handle number restrictions $(\geq n\ \mathcal{R})$ for complex roles with union or intersection and by modifying the rule for number restrictions such that it can handle the new types of complex roles. To formulate the new rules, we must extend the notion of a role successor in a constraint system appropriately. Building up on the notion of a role successor for a role chain, we define:

- y is an $(R_1 \circ \ldots \circ R_n \sqcup S_1 \circ \ldots \circ S_n)$-successor of x in S iff y is an $R_1 \circ \ldots \circ R_n$-successor or an $S_1 \circ \ldots \circ S_n$-successor of x in S, and

5a. Number restriction: If $x:(\geq n\ R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m) \in S$ and x has less than n $(R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m)$-successors in S, then

$$S \rightarrow S_1 = S \cup \{xR_1y_2, y_mR_mz\} \cup \{y_iR_iy_{i+1} \mid 2 \leq i \leq m-1\} \cup \{z \neq w \mid w \text{ is an } (R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m)\text{-successor of } x \text{ in } S\}$$

$$S \rightarrow S_2 = S \cup \{xS_1y_2, y_mS_mz\} \cup \{y_iS_iy_{i+1} \mid 2 \leq i \leq m-1\} \cup \{z \neq w \mid w \text{ is an } (R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m)\text{-successor of } x \text{ in } S\}$$

where z, y_i are new variables in $\tau \setminus \tau_S$.

5b. Number restriction: If $x:(\geq n\ R_1\circ\ldots\circ R_m \sqcap S_1\circ\ldots\circ S_m) \in S$ and x has less than n $(R_1\circ\ldots\circ R_m \sqcap S_1\circ\ldots\circ S_m)$-successors in S, then

$$S \rightarrow S \cup \{xR_1y_2, xS_1y_2', y_mR_mz, y_m'S_mz\} \cup \{y_iR_iy_{i+1}, y_i'S_iy_{i+1}' \mid 2 \leq i \leq m-1\} \cup \{z \neq w \mid w \text{ is an } (R_1\circ\ldots\circ R_m \sqcap S_1\circ\ldots\circ S_m)\text{-successor of } x \text{ in } S\}$$

where z, y_i', y_i are new variables in $\tau \setminus \tau_S$.

6'. Number restriction: If $x:(\leq n\ \mathcal{R}) \in S$ for some complex role $\mathcal{R}$, x has more than n $\mathcal{R}$-successors in S, and there are $\mathcal{R}$-successors y_1, y_2 of x in S with $(y_1 \neq y_2) \notin S$, then

$$S \rightarrow S_{y_1,y_2} = S[y_2/y_1]$$

for all pairs y_1, y_2 of $\mathcal{R}$-successors of x with $(y_1 \neq y_2) \notin S$.

Figure 8: The additional completion rules.

- y is an $(R_1\circ\ldots\circ R_n \sqcap S_1\circ\ldots\circ S_n)$-successor of x in S iff y is an $R_1\circ\ldots\circ R_n$-successor and an $S_1\circ\ldots\circ S_n$-successor of x in S.

Obviously, this definition is such that role successors in S are also role successors in every model of S: if $\mathcal{I}, \pi$ satisfy S, and y is an $\mathcal{R}$-successor of x in S for a complex role $\mathcal{R}$, then $\pi(y)$ is an $\mathcal{R}$-successor of $\pi(x)$ in $\mathcal{I}$.

The new rules are described in Figure 8. The rules $5a$, $5b$ are added to the completion rules, whereas rule $6'$ substitutes rule 6 in Figure 7. To show that the new algorithm obtained this way decides satisfiability of concepts for the extended logic, we must prove that all four parts of Lemma 8 still hold.

1. *Local correctness* of the rules $5a, 5b$ and $6'$ can be shown as in the proof of Part 1 of Lemma 8 above.

2. The *canonical model* induced by a complete and clash-free constraint system is defined as in the proof of Part 2 of Lemma 8. The proof that this canonical model really satisfies the constraint system is also similar to the one given there. Note that our notion of an $\mathcal{R}$-successor of a complex role $\mathcal{R}$ in a constraint system was defined such that it coincides with the notion of an $\mathcal{R}$-successor in the canonical model $\mathcal{I}$ induced by the constraint system.

3. The proof that a constraint system containing a clash is unsatisfiable is the same as the

one given above. Note that this depends on the fact that role successors in a constraint system are also role successors in every model of the constraint system.

4. The proof of *termination* is also very similar to the one given above. The definition of the depth of a concept (see the appendix) is extended in the obvious way to concepts with number restrictions on complex roles:

$$\begin{aligned} \mathit{depth}(\geq n\ R_1\circ\ldots\circ R_m \sqcap S_1\circ\ldots\circ S_m) &:= m, \\ \mathit{depth}(\geq n\ R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m) &:= m, \\ \mathit{depth}(\leq n\ R_1\circ\ldots\circ R_m \sqcap S_1\circ\ldots\circ S_m) &:= m, \\ \mathit{depth}(\leq n\ R_1\circ\ldots\circ R_m \sqcup S_1\circ\ldots\circ S_m) &:= m. \end{aligned}$$

   Because the role chains in complex roles are of the same length, it is easy to see that Lemma 10 still holds. Thus, we can define the same measure $\kappa(S)$ as in the appendix for all constraint systems obtained by applying the extended completion rules to $\{x_0 : C_0\}$. It is easy to see that the proof that $S \rightarrow S'$ implies $\kappa(S) \succ \kappa(S')$ can be extended to the new rules. It should be noted that the proof given in the appendix was already formulated in a more general way than necessary for the logic considered there. Actually, we have only used the fact that all role chains connecting two individuals have the same length (which is still satisfied for the extended logic), and not that these role chains also have the same name (which is only satisfied for $\mathcal{ALCN}(\circ)$).

The following theorem is an immediate consequence of these observations:

Theorem 11 Subsumption and satisfiability is decidable for the logic that extends $\mathcal{ALCN}(\circ)$ by number restrictions on union and intersection of role chains of the same length.

4 Symbolic Number Restrictions

In this section, we introduce the extension of $\mathcal{ALCN}$ by symbolic number restrictions and investigate the complexity of satisfiability and subsumption of this extension. As motivated by the examples in the introduction, we need a formalism that allows us to introduce explicitly existentially quantified numerical variables in number restrictions. If we want to extend $\mathcal{ALCN}$ such that it is still closed under negation, universal quantification of numerical variables comes in as the dual of existential quantification. We will show that this propositionally closed extension is undecidable. However, if we restrict the use of negation such that universally quantified numerical variables do not occur, satisfiability becomes decidable. Unfortunately, subsumption of this restricted logic is still undecidable.

4.1 Syntax and Semantics

In order to introduce symbolic number restrictions, we must extend our vocabulary by variables that stand for non-negative integers.

Definition 12 Let N_V be a set of numerical variables. Then $\mathcal{ALCN}^S$ is obtained from $\mathcal{ALCN}$ by additionally allowing for

- *symbolic* number restrictions $(\leq \alpha\ R)$ and $(\geq \alpha\ R)$ for a role name R and a numerical variable α, and

- the existential quantification $(\downarrow\alpha.C)$ of numerical variables α where C is an $\mathcal{ALCN}^S$-concept.

As in the case of traditional number restrictions, we use additional relations $=, <, >$ as abbreviations. For example, $(= \alpha\ R)$ is an abbreviation for $(\leq \alpha\ R) \sqcap (\geq \alpha\ R)$. To give an intuitive understanding of the meaning of symbolic number restrictions, we first present two examples: the concept

$$\texttt{Human} \sqcap (\forall \texttt{child}.\downarrow\alpha.(= \alpha\ \texttt{vice}) \sqcap (> \alpha\ \texttt{virtue}))$$

describes persons whose children all have less vices than virtues, whereas the concept

$$\texttt{Human} \sqcap (\downarrow\alpha.\forall \texttt{child}.(= \alpha\ \texttt{vice}) \sqcap (> \alpha\ \texttt{virtue}))$$

describes persons whose children all have the same number of vices, which is smaller than the number of their virtues.

Since $\mathcal{ALCN}^S$ allows for full negation of concepts, universal quantification of numerical variables can be expressed: in the following, we us $(\uparrow\alpha.C)$ as shorthand for $\neg(\downarrow\alpha.\neg C)$. Before giving the semantics of $\mathcal{ALCN}^S$-concepts, we define what it means for a numerical variable to occur free in a concept.

Definition 13 The occurrence of a variable $\alpha \in N_V$ is said to be *bound in* C iff α occurs in the scope C' of a quantified subterm $(\downarrow\alpha.C')$ of C. Otherwise, the occurrence is said to be *free*. The set free$(C) \subseteq N_V$ denotes the set of variables that occur free in C. The concept C is *closed* iff free$(C) = \emptyset$. For a non-negative integer n, the concept $C[\frac{n}{\alpha}]$ is obtained from the concept C by substituting all free occurrences of α by n.

Note that, as usual, a variable can occur both free and bound in a concept. For example, α occurs both free and bound in $((= \alpha\ R) \sqcap (\downarrow\alpha.(\exists R.(> \alpha\ R))))$.

Using this notation, we can define the semantics of $\mathcal{ALCN}^S$-concepts.

Definition 14 An $\mathcal{ALCN}^S$-interpretation is an $\mathcal{ALCN}$-interpretation that, additionally, satisfies the equation

$$(\downarrow\alpha.C)^{\mathcal{I}} = \bigcup_{n \in \mathbf{N}} (C[\frac{n}{\alpha}])^{\mathcal{I}}$$

for all *closed* $\mathcal{ALCN}^S$-concepts $(\downarrow\alpha.C)$. If C is not closed and free$(C) = \{\alpha_1, \ldots, \alpha_n\}$ for $n \geq 1$ then

$$C^{\mathcal{I}} := (\downarrow\alpha_1. \ldots \downarrow\alpha_n.C)^{\mathcal{I}}.$$

This definition reduces the semantics of symbolic number restrictions to the semantics of traditional ones. Since $(\uparrow\alpha.C)$ is an abbreviation for $\neg(\downarrow\alpha.\neg C)$, we can give its semantics directly by

$$(\uparrow\alpha.C)^{\mathcal{I}} = \bigcap_{n \in \mathbf{N}} (C[\frac{n}{\alpha}])^{\mathcal{I}}.$$

Similar to $\mathcal{ALCN}$, it can be shown that $\mathcal{ALCN}^S$ still has the tree-model property, but in contrast to $\mathcal{ALCN}$, the logic $\mathcal{ALCN}^S$ does not have the finite-model property. For example, the concept

$$(\uparrow\alpha.(\geq \alpha\ R)) \tag{3}$$

is satisfiable, but each instance of (3) has infinitely many R-successors. On the one hand, the interpretation $\mathcal{I}$ where

$$\begin{array}{rcl} \Delta^{\mathcal{I}} & := & \{x, y_0, y_1, y_2, \ldots\} \\ R^{\mathcal{I}} & := & \{(x, y_i) \mid i \in \mathbb{N}\} \end{array}$$

is clearly a model of (3). On the other hand, each model $\mathcal{I}$ of (3) satisfies $\cap_{n \in \mathbb{N}}(\geq n\ R)^{\mathcal{I}} \neq \emptyset$, hence each instance of (3) has infinitely many R-successors.

To give a better insight into the expressive power of symbolic number restrictions we first give the undecidability result.

4.2 $\mathcal{ALCN}^S$ is undecidable

Similar to the undecidability proofs in Section 3.1, undecidability of satisfiability for $\mathcal{ALCN}^S$ is shown by a reduction of the domino problem to concept satisfiability. For $\mathcal{ALCN}^S$, however, the proof is easier if we take another variant of the domino problem: instead of asking for a compatible tiling of the first quadrant of the plane, we now ask for a compatible tiling of the "second eighth" $(\mathbb{N} \times \mathbb{N})_{\leq} := \{(a, b) \mid a, b \in \mathbb{N} \text{ and } a \leq b\}$ of the plane. Since such a tiling yields compatible tilings of arbitrarily large finite rectangles, it also yields a compatible tiling of the plane [14].

In contrast to the reduction given in Section 3.1, in this reduction, the individuals representing points in the grid are not related to each other by roles—there is no equivalent to the "horizontal" and "vertical" roles X and Y. Instead, the reduction works as follows: First, we define an $\mathcal{ALCN}^S$-concept $C_{\mathbb{N}}$ such that, for each model of $C_{\mathbb{N}}$ with $o \in C_{\mathbb{N}}^{\mathcal{I}}$, there is a natural relationship between tuples $(a, b) \in (\mathbb{N} \times \mathbb{N})_{\leq}$ and S-successors $y_{a,b}$ of o. The point (a, b) is represented by an S-successor of o having a L-successors and b R-successors. Second, for a given tiling system $\mathcal{D}$, we construct a concept $C_{\mathcal{D}}$ that (1) is subsumed by $C_{\mathbb{N}}$, (2) ensures that every $y_{a,b}$ has exactly one domino type, and (3) encodes the compatibility conditions of the matching pairs.

The formal definition of $C_{\mathbb{N}}$ is given in Figure 9. Assume that $\mathcal{I}$ is a model of $C_{\mathbb{N}}$ with $o \in C_{\mathbb{N}}^{\mathcal{I}}$. Now, C_1 makes sure that, for every non-negative integer a, o has an S-successor having exactly a L-successors. The precondition of C_2 makes sure that a is smaller than b, and thus the whole implication says that, for each pair $a \leq b$ of non-negative integers, o has an S-successor having exactly a L-successors and b R-successors (there can be more than one such S-successor). Finally, C_3 says that, whenever an S-successor of o has a L-successors and b R-successors, we have $a \leq b$. Thus, there is an obvious correspondence between S-successors of o and points in the second eighth of the plane: every S-successor corresponds to a point in $(\mathbb{N} \times \mathbb{N})_{\leq}$ and vice versa. More formally, we will prove the following observations concerning $C_{\mathbb{N}}$ where, for a role name R and some $x \in \Delta^{\mathcal{I}}$, $x_R{}^{\mathcal{I}}$ denotes the number of R-fillers of x in $\mathcal{I}$, that is

$$x_R{}^{\mathcal{I}} := \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\}.$$

Lemma 15 Let $C_{\mathbb{N}}$ be the concept introduced in Figure 9.

1. $C_{\mathbb{N}}$ is satisfiable.

2. Let $\mathcal{I}$ be a model of $C_{\mathbb{N}}$ with $o \in C_{\mathbb{N}}^{\mathcal{I}}$ and let $Y = \{y \in \Delta^{\mathcal{I}} \mid (o, y) \in S^{\mathcal{I}}\}$.

$$C_{\mathbb{N}} := (\uparrow\alpha.\uparrow\beta.(C_1 \sqcap C_2 \sqcap C_3)) \text{ where}$$

$$\begin{aligned}
C_1 &:= (\exists S.(=\alpha\ L))\\
C_2 &:= ((\exists S.(=\alpha\ L) \sqcap (\leq \beta\ L)) \Rightarrow (\exists S.(=\alpha\ L) \sqcap (=\beta\ R)))\\
C_3 &:= (\forall S.((=\alpha\ L) \sqcap (=\beta\ R)) \Rightarrow (\leq \beta\ L))
\end{aligned}$$

Given a tiling system $\mathcal{D} = (\{D_1, \ldots, D_m\}, H, V)$ and the subconcepts C_1, C_2, C_3 of $C_{\mathbb{N}}$ as defined above, let

$$\begin{aligned}
C_{\mathcal{D}} := C_{\mathbb{N}} \sqcap (\forall S.(\bigsqcup_{1\leq i\leq m}(D_i \sqcap (\bigsqcap_{\substack{1\leq j\leq m\\ i\neq j}} \neg D_j)))) \sqcap\\
(\uparrow\alpha.\uparrow\beta.\bigsqcap_{1\leq i\leq m}(\exists S\ .((=\alpha\ L) \sqcap (=\beta\ R) \sqcap D_i)) \Rightarrow\\
((\forall S.((\neq \alpha\ L) \sqcup (\neq \beta\ R) \sqcup D_i)) \sqcap \quad (1)\\
(\uparrow\gamma.(<(\alpha,\beta) \sqcap =(\alpha+1,\gamma)) \Rightarrow\\
(\forall S.(((=\gamma\ L) \sqcap (=\beta\ R)) \Rightarrow \bigsqcup_{(D_i,D_j)\in H} D_j))) \sqcap \quad (2)\\
(\uparrow\gamma.(=(\beta+1,\gamma) \Rightarrow\\
(\forall S.(((=\alpha\ L) \sqcap (=\gamma\ R)) \Rightarrow \bigsqcup_{(D_i,D_j)\in V} D_j)))))) \quad (3)
\end{aligned}$$

Figure 9: Definition of the concepts $C_{\mathbb{N}}$ and $C_{\mathcal{D}}$ used for the reduction of the domino problem to the $\mathcal{ALCN}^S$ satisfiability problem

(i) For each $(a, b) \in (\mathbb{N} \times \mathbb{N})_{\leq}$ there exists $y_{a,b} \in Y$ with ${(y_{a,b})_L}^{\mathcal{I}} = a$ and ${(y_{a,b})_R}^{\mathcal{I}} = b$.

(ii) If $y \in Y$ and ${y_L}^{\mathcal{I}} = a$ and ${y_R}^{\mathcal{I}} = b$, then $a \leq b$.

3. If $o \in {C_{\mathbb{N}}}^{\mathcal{I}}$, then there is an injective mapping $\phi\colon (\mathbb{N} \times \mathbb{N})_{\leq} \to Y$ from the second eighth of the plane to the set of S-successors of o.

PROOF. 1. Define $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and o as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{o\} \uplus \{y_{a,b} \mid (a,b) \in (\mathbb{N} \times \mathbb{N})_{\leq}\} \uplus \{\ell_a, r_b \mid a, b \in \mathbb{N}\},\\
S^{\mathcal{I}} &= \{(o, y_{a,b}) \mid (a,b) \in (\mathbb{N} \times \mathbb{N})_{\leq}\},\\
L^{\mathcal{I}} &= \{(y_{a,b}, \ell_{a'}) \mid (a,b) \in (\mathbb{N} \times \mathbb{N})_{\leq} \text{ and } a' < a\},\\
R^{\mathcal{I}} &= \{(y_{a,b}, r_{b'}) \mid (a,b) \in (\mathbb{N} \times \mathbb{N})_{\leq} \text{ and } b' < b\}.
\end{aligned}$$

$\mathcal{I}$ is a well-defined $\mathcal{ALCN}^S$-interpretation and it is clear that, for all $(a, b) \in (\mathbb{N} \times \mathbb{N})_{\leq}$, we have ${(y_{a,b})_L}^{\mathcal{I}} = a$ and ${(y_{a,b})_R}^{\mathcal{I}} = b$. It remains to be shown that $o \in {C_{\mathbb{N}}}^{\mathcal{I}}$:

We know that $o \in {C_{\mathbb{N}}}^{\mathcal{I}}$ iff for all $a, b \in \mathbb{N}$: $o \in (C_1[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$, $o \in (C_2[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$, and $o \in (C_3[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$. Thus, let $a, b \in \mathbb{N}$.

- $o \in (C_1[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ since $(o, y_{a,b'}) \in S^{\mathcal{I}}$ for some $b' \geq a$.

- $o \in (C_2[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$: If $o \in (\exists S.(= a\ L) \sqcap (\leq b\ L))^{\mathcal{I}}$, then $a \leq b$ and $(o, y_{a,b}) \in S^{\mathcal{I}}$, which implies $o \in (\exists S.(= a\ L) \sqcap (= b\ R))^{\mathcal{I}}$.

- $o \in (C_3[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$: Let $(o, y) \in S^{\mathcal{I}}$. If $y \in ((= a\ L) \sqcap (= b\ R))^{\mathcal{I}}$, then $y = y_{a,b}$ with $a \leq b$, which implies $y \in (\leq b\ L)^{\mathcal{I}}$.

2(i). The subconcept C_1 ensures that, for each $a \in \mathbb{N}$, there exists some $y_a \in Y$ with ${(y_a)_L}^{\mathcal{I}} = a$. If $a, b \in \mathbb{N}$ satisfy $a \leq b$, then y_a obviously belongs to $((= a\ L) \sqcap (\leq b\ L))^{\mathcal{I}}$. Thus, the subconcept C_2 ensures that there also exists an S-successor $y_{a,b}$ of o that has a L-successors and b R-successors.

2(ii). The subconcept C_3 ensures that, for all $y \in Y$, ${y_L}^{\mathcal{I}} = a$ and ${y_R}^{\mathcal{I}} = b$ implies ${y_L}^{\mathcal{I}} \leq b$, and thus $a \leq b$.

3. This is a direct consequence of 2(i): we define $\phi(a, b) := y$ where $y \in Y$ is such that ${y_L}^{\mathcal{I}} = a$ and ${y_R}^{\mathcal{I}} = b$. ■

Please note that, for $a, b \in \mathbb{N}$, there might be more than one $y \in Y$ with ${y_L}^{\mathcal{I}} = a$ and ${y_R}^{\mathcal{I}} = b$.

The definition of the concept $C_{\mathcal{D}}$ associated with a tiling system $\mathcal{D}$ is also given in Figure 9, where the following abbreviations are employed:

$$\begin{aligned} <(\alpha, \beta) &:= (\exists S.((= \alpha\ L) \sqcap (= \beta\ R) \sqcap \neg(= \beta\ L))), \\ =(\alpha + 1, \beta) &:= <(\alpha, \beta) \sqcap (\forall S.((\leq \alpha\ L) \sqcup (\geq \beta\ L))). \end{aligned}$$

In the context of the concept $C_{\mathbb{N}}$, these abbreviations really express the relation $<$ and the successor relation on natural numbers: for $o \in {C_{\mathbb{N}}}^{\mathcal{I}}$, we have $o \in (<(\alpha, \beta)[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ iff $a < b$ as an immediate consequence of Lemma 15.2. Furthermore, $o \in (=(\alpha + 1, \beta)[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ iff $a + 1 = b$ since o has some S-successor having a L-successors for each $a \in \mathbb{N}$.

The first line in the definition of $C_{\mathcal{D}}$ makes sure that $C_{\mathbb{N}}$ subsumes $C_{\mathcal{D}}$, and that every S-successor of an instance o of $C_{\mathcal{D}}$ has exactly one domino type. In the remainder of the definition, we consider an S-successor $y_{a,b}$ with domino type D_i and a L- and b R-successors. Now, (1) ensures that every S-successor with the same number of L- and R-successors as $y_{a,b}$ has the same domino type D_i, (2) takes care of the horizontal matching condition, and (3) of the vertical matching condition. Given this intuition, it is easy to show that the following lemma holds.

Lemma 16 $C_{\mathcal{D}}$ is satisfiable iff there exists a compatible tiling of the second eighth of the plane using $\mathcal{D}$.

The proof of this lemma can be found in the appendix.

Now, undecidability of the domino problem yields undecidability of the satisfiability problem for $\mathcal{ALCN}^S$-concepts. Since C is unsatisfiable iff $C \sqsubseteq (A \sqcap \neg A)$, this implies undecidability of subsumption.

Theorem 17 Satisfiability and subsumption of $\mathcal{ALCN}^S$-concepts are undecidable.

4.3 A decidable restriction of $\mathcal{ALCN}^S$

We have seen in the last section that, by using universally quantified numerical variables in $\mathcal{ALCN}^S$, we can enforce infinite models. The undecidability proof also makes strong use of universal quantification. In order to obtain a decidable extension of $\mathcal{ALCN}$ with symbolic number restrictions, which also has the finite-model property, we introduce $\mathcal{ALUEN}^S$, a fragment of $\mathcal{ALCN}^S$ that is obtained by allowing only for existential quantification of numerical variables. This is achieved by restricting the use of negation.

Definition 18 $\mathcal{ALUEN}^S$-concepts are those $\mathcal{ALCN}^S$-concepts where negation occurs only in front of concept names or number restrictions.

In the following, we will refer to concept names and number restrictions as *atomic* concepts. Since in $\mathcal{ALCN}^S$ universal quantification of numerical variables came in only as an abbreviation of negated existential quantification, all numerical variables in $\mathcal{ALUEN}^S$ are therefore existentially quantified. Nevertheless, the logic is still an extension of $\mathcal{ALCN}$ since $\mathcal{ALCN}$-concepts in NNF satisfy the above restriction. Furthermore, all examples given in Section 1 to motivate the introduction of symbolic number restrictions are $\mathcal{ALUEN}^S$-concepts.

In this section, it will be shown that satisfiability of $\mathcal{ALUEN}^S$-concepts is decidable. In order to simplify our investigation of the satisfiability problem for $\mathcal{ALUEN}^S$-concepts, we will restrict our attention to concepts where each numerical variable occurs either bound or free, and where each variable is bound at most once by $\downarrow$. It is easy to see that each $\mathcal{ALUEN}^S$-concept can be transformed into an equivalent concept of this form by existentially quantifying all free variables and by appropriately renaming bound variables.

Decidability of satisfiability of $\mathcal{ALUEN}^S$-concepts will be shown by presenting a tableau-based algorithm and showing that, for each $\mathcal{ALUEN}^S$-concept C, this algorithm is sound, complete, and terminating. Similarly to the algorithm presented in Section 3.2, the algorithm works on constraints, but for $\mathcal{ALUEN}^S$-concepts we need additional variables α_x: Suppose we have the constraint $y : (\forall R.(\downarrow\alpha.C))$. Then, for each R-successor x of y, we need a variable α_x that stand for α "in the context of x". Since there are further subtle differences between the algorithm in Section 3.2 and the one for $\mathcal{ALUEN}^S$, we provide a complete description of the latter.

Definition 19 We assume that we have a countably infinite set $\tau = \{x, y, z, \ldots\}$ of individual variables, and for each pair $(\alpha, x) \in N_V \times \tau$ a new numerical variable α_x which may occur free in concepts. A *constraint* is either of the form

xRy, where R is a role name in N_R and $x, y \in \tau$, or

$x : D$ for some $\mathcal{ALUEN}^S$-concept D and some $x \in \tau$.

A *constraint system* is a set of constraints.

An interpretation $\mathcal{I}$ is a model of a constraint system S iff there is a mapping $\pi : \tau \to \Delta^{\mathcal{I}}$ and a mapping $\nu : N_V \times \tau \to \mathbb{N}$ such that $\mathcal{I}, \pi, \nu$ satisfy each constraint in S, i.e., we have

$$(\pi(x), \pi(y)) \in R^{\mathcal{I}} \quad \text{for all } xRy \in S,$$
$$\pi(x) \in \nu(D)^{\mathcal{I}} \quad \text{for all } x : D \in S,$$

where $\nu(D)$ is obtained from D by replacing each variable α_y by its ν-image $\nu(\alpha, y)$.

A constraint system S is said to contain a *clash* iff for some concept name A and some variable $x \in \tau$ we have $\{x : A, x : \neg A\} \subseteq S$. A constraint system S is said to be *numerically consistent* iff the conjunction of all numerical constraints in S, i.e.,

$$\bigwedge_{\substack{x : (\mathrm{rel}\, n\ R) \in S \\ x \in \tau, R \in N_R, n \in \mathbb{N}}} (x_R \,\mathrm{rel}\, n) \wedge \bigwedge_{\substack{x : (\mathrm{rel}\, \alpha_y\ R) \in S \\ x, y \in \tau, R \in N_R, \alpha \in N_V}} (x_R \,\mathrm{rel}\, \alpha_y),$$

is satisfiable in $(\mathbb{N}, <)$, where x_R, α_y are interpreted as variables for non-negative integers and rel stands for relations in $\{\leq, \geq, <, >, =\}$.

A constraint system S is called *complete* iff none of the completion rules of Figure 10 can be applied to S.

Like the algorithm presented in the previous section, the algorithm for $\mathcal{ALUEN}^S$ works on a tree where each node is labelled with a constraint system. It starts with a tree consisting of a root labelled with $S = \{x_0 : C_0\}$ for some closed concept C_0. A rule can only be applied to a leaf labelled with a clash-free constraint system. Applying a rule $S \rightarrow S_i$, for $1 \leq i \leq n$, to such a leaf leads to the creation of n new successors of this node, each labelled with the constraint systems S_i. The algorithm terminates if none of the rules can be applied to any of the leaves. The algorithm answers with "C_0 is satisfiable" iff one of the leaves obtained this way is a clash-free, numerically consistent, and complete constraint system.

Before showing that the completion algorithm described in Figure 10 yields a decision procedure for satisfiability of $\mathcal{ALUEN}^S$-concepts, let us make some comments on the rules. First, note that each of the completion rules adds constraints when applied to a constraint system, none of the rules removes constraints, and individual variables are never identified or substituted. With respect to this last property, the algorithm for $\mathcal{ALUEN}^S$ differs from the tableau-based algorithms for $\mathcal{ALCN}$ described in [9] and for $\mathcal{ALCN}(\circ)$ presented in the previous section. Unlike Rule 4 in Figure 10, these algorithms introduce, for each constraint of the form $x : \exists R.C$, a new R-successor of x. If x also has a constraint of the form $x : (\leq n\ R)$, and more than n R-successors have been introduced, then some of these individuals are identified. Rule 4 in Figure 10 avoids identification by "guessing" the number of allowed R-successors of x before introducing these successors. In fact, since we do not have explicit numbers, and since restrictions on numerical variables α_y in constraints $x : (\leq \alpha_y\ R)$ can derive from different parts of the constraint system, an identification on demand is not possible here. The second new feature is Rule 3. Given a constraint $x : (\downarrow\alpha.D)$, we substitute a new numerical variable α_x for α to make sure that the semantics of the existential quantifier $\downarrow\alpha$ is obeyed, i.e., that the valuation for α depends on x. If we would just use α, the difference between $\downarrow\alpha.\forall R.D$ and $\forall R.(\downarrow\alpha.D)$ would not be captured.

Again, correctness of this algorithm is an easy consequence of the following lemma.

Lemma 20 Let C_0 be a closed $\mathcal{ALUEN}^S$-concept, and let S be a constraint system obtained by applying the completion rules to $\{x_0 : C_0\}$. Then

1. The completion algorithm terminates when applied to $\{x_0 : C_0\}$.
2. For each completion rule $\mathcal{R}$ that can be applied to S, and for each interpretation $\mathcal{I}$ we have: $\mathcal{I}$ is a model of S iff $\mathcal{I}$ is a model of one of the systems S_i obtained by applying $\mathcal{R}$.
3. If S is clash-free, numerically consistent, and complete, then S has a model.
4. If S contains either a clash or is not numerically consistent, then S does not have a model.

PROOF. 1. The termination proof is similar to the one for the tableau-based algorithm for $\mathcal{ALCN}$ [9].

1. Conjunction: If $x :(C_1 \sqcap C_2) \in S$ and $x : C_1 \notin S$ or $x : C_2 \notin S$, then

$S \rightarrow_{\sqcap} S \cup \{x : C_1, x : C_2\}$

2. Disjunction: If $x :(C_1 \sqcup C_2) \in S$ and $x : C_1 \notin S$ and $x : C_2 \notin S$, then

$S \rightarrow_{\sqcup} S_1 = S \cup \{x : C_1\}$

$S \rightarrow_{\sqcup} S_2 = S \cup \{x : C_2\}$

3. Numerical Existential Quantification: If $x :(\downarrow\alpha.D) \in S$ and $x : D[\frac{\alpha_x}{\alpha}] \notin S$, then

$S \rightarrow_{\downarrow} S \cup \{x : D[\frac{\alpha_x}{\alpha}]\}$

4. New Objects:
If $xRy \notin S$ for all $y \in \tau$, and $m > 0$, $k \geq 0$ are *maximal* such that

$\{x :(\exists R.E_1), \ldots, x :(\exists R.E_m), x :(\forall R.D_1), \ldots, x :(\forall R.D_k)\} \subseteq S$

and Rules 1–3 cannot be applied to S, then for each n with $1 \leq n \leq m$ and for each n-Partition $P = \uplus_{1 \leq i \leq n} P_i$ of $\{1, \ldots, m\}$, let S_P be defined as follows:

$$\begin{aligned} S \rightarrow_R S_P \;=\; & S \cup \{xRy_i, \mid 1 \leq i \leq n\} \cup \{x :(\geq n\ R)\} \cup \\ & \{y_i : E_j \mid 1 \leq i \leq n, j \in P_i\} \cup \{y_i : D_j \mid 1 \leq i \leq n, 1 \leq j \leq k\} \end{aligned}$$

where $y_i \in \tau$ are new variables (i.e., variables not occurring in S).

5. Prophylactic new objects:
If $xRy \notin S$ for all $y \in \tau$ and $x :(\leq 0\ R) \notin S$ and k maximal with $x :(\forall R.D_i) \in S$ for $1 \leq i \leq k$, $x :(\text{rel}\, N\ R) \in S$ for $N \in \mathbb{N}$ or $N = \alpha_y$ for some $y \in \tau, \alpha \in N_V$ and Rules 1–4 cannot be applied to S, then S_1, S_2 are defined as follows:

$S \rightarrow_n S_1 = S \cup \{x :(\leq 0\ R)\}$

$S \rightarrow_n S_2 = S \cup \{xRy\} \cup \{y : D_i \mid 1 \leq i \leq k\} \cup \{x :(> 0\ R)\}$

where $y \in \tau$ is a new variables (i.e., a variable not occurring in S).

Figure 10: The *completion algorithm* for $\mathcal{ALUEN}^S$-concepts

2. We consider only Rules 3, 4 and 5 since Rules 1 and 2 are obvious. If S' is generated by the application of a completion rule to S, then $S \subseteq S'$. Hence every model of S' is also a model of S. Thus we must consider only the other direction.

Numerical Existential Quantification: Application of this rule adds the constraint $x : C[\frac{\alpha_x}{\alpha}]$ to S, where $x : \downarrow\alpha.C$ is contained in S. If $\mathcal{I}, \pi, \nu$ satisfy S, then we know that there exists an $n \in \mathbb{N}$ such that $\pi(x) \in \nu(C[\frac{n}{\alpha}])^{\mathcal{I}}$. Since the variable α_x does not occur in S (by our assumption that every variable is bound only once in the input concept), we can assume without loss of generality that $\nu(\alpha_x) = n$, and thus $\mathcal{I}, \pi, \nu$ satisfy $x : C[\frac{\alpha_x}{\alpha}]$.

New Objects: Let x, R, k, m be as specified in the precondition of Rule 4 and let $\mathcal{I}$ satisfy S. Since $\{x :(\exists R.E_1), \ldots, x :(\exists R.E_m), x :(\forall R.D_1), \ldots, x :(\forall R.D_k)\} \subseteq S$, there exist some $\ell \leq m$ and ℓ distinct elements $d_1, \ldots, d_\ell \in \Delta^{\mathcal{I}}$ such that

- $(\pi(x), d_i) \in R^{\mathcal{I}}$ for all i with $1 \leq i \leq \ell$,

- for all $1 \leq j \leq m$ there is some $j' \in \{1, \ldots, \ell\}$ with $d_{j'} \in E_j{}^{\mathcal{I}}$, and
- for all $1 \leq j \leq k$ and all $1 \leq i \leq \ell$ we have $d_i \in D_j{}^{\mathcal{I}}$.

The second item above implies that there exists at least one function $f\colon \{1, \ldots, m\} \rightarrow \{1, \ldots, \ell\}$ such that $d_{f(j)} \in E_j{}^{\mathcal{I}}$ for all $1 \leq j \leq \ell$. Let P be the ℓ-partition of $\{1, \ldots, m\}$ induced by f, i.e., $P_{j'} := \{j \mid f(j) = j'\}$. In the corresponding constraint system S_P, ℓ new variables y_i and the corresponding new constraints are introduced. Let $\pi(y_i) = d_i$ for $1 \leq i \leq \ell$. Then, by definition of P and the three items from above,

- $(\pi(x), \pi(y_i)) \in R^{\mathcal{I}}$ for all $1 \leq i \leq \ell$,
- for each of the new constraints $y_i : E_j$ in S_P we have $\pi(y_i) \in E_j{}^{\mathcal{I}}$ since $j \in P_i$ implies $f(j) = i$, and thus $\pi(y_i) = d_{f(j)} \in E_j{}^{\mathcal{I}}$,
- for each of the new constraints $y_i : D_j$ in S_P we have $\pi(y_i) = d_i \in D_j{}^{\mathcal{I}}$, and
- $x_R{}^{\mathcal{I}} \geq \ell$ since $\pi(x)$ has at least the R-successors $d_1, \ldots, d_\ell$.

Hence $\mathcal{I}$ satisfies S_P.

Prophylactic New Objects: Let x, R, k be as specified in the precondition of Rule 5 and assume that $\mathcal{I}$ satisfies S. Two cases are to be distinguished: If $x_R{}^{\mathcal{I}} = 0$, then clearly $\mathcal{I}$ satisfies S_1. Now let $x_R{}^{\mathcal{I}} > 0$ with $(\pi(x), d) \in R^{\mathcal{I}}$ for some $d \in \Delta^{\mathcal{I}}$. If we define $\pi(y) = d$, then $\mathcal{I}$ satisfies $S_2 = S \cup \{xRy\} \cup \{x : (> 0\ R)\} \cup \{y : D_i \mid 1 \leq i \leq k\}$.

3. As usual, we construct the canonical interpretation $\mathcal{I}_S$ induced by S: $\Delta^{\mathcal{I}_S}$ consists of the individual variables occurring in S; $(x, y) \in R^{\mathcal{I}_S}$ iff $xRy \in S$; and $x \in A^{\mathcal{I}_S}$ iff $x : A \in S$. This yields a tree-like interpretation. However, this Interpretation need not be a model of S since some number restrictions may be violated for one of the following reasons. Either (a) an individual does not have any role successors, but their existence is implied by number restrictions, or (b) it has some, but not sufficiently many role successors. Note that exact numerical restrictions on the number of role successors are given by a solution in $(\mathbb{N}, <)$ of the numerical constraints (which are satisfiable since S is numerically consistent). In the first case, S does not contain any constraints on such role successors since Rule 5 is not applicable. Thus, we can simply generate an appropriate number of them. In the second case, the idea is to add sufficiently many *copies* of some already existing role successor y. More precisely, we need to copy the whole subtree that has y as its root. Proceeding like this from the leaves to the root, we end up with a model of S. This can be shown by induction on the structure of concepts in constraints.

4. This is obvious. ■

Theorem 21 Satisfiability of $\mathcal{ALUEN}^S$-concepts is decidable.

PROOF. Lemma 20 implies that the completion algorithm always terminates. In addition, the second statement of the lemma shows that the original system $\{x_0 : C_0\}$ has a model iff one of the leaves of the tree obtained by the algorithm has a model. Thus, if none of the leaves is clash-free and numerically consistent, then the fourth statement of the lemma shows that $\{x_0 : C_0\}$ does not have a model. Otherwise, one of the leaves is a clash-free, numerically consistent, and complete, and thus the third statement of the lemma shows that $\{x_0 : C_0\}$ has a model. Obviously, $\{x_0 : C_0\}$ has a model iff C_0 is satisfiable. It remains to be shown that it is decidable whether a constraint system contains a clash and whether a constraint system is numerically consistent. Detecting clashes is trivial.

Numerical consistency can be tested using a modified cycle detection algorithm running in time polynomial in the size of the formula. To be more precise, a given formula is translated into a graph whose nodes correspond to the numerical variables and non-negative integers occurring in the formula. The edges are induced by the numerical constraints of the formula. For example, $\alpha \leq \beta$ yields an edge from the node corresponding to α to the node corresponding to β, and this edge is labelled with $\leq$. Obviously, if there is a cyclic path in the graph that is labelled with at least one strict inequality, then the formula is unsatisfiable. Because of the presence of concrete numbers, testing for cycles is not sufficient, though. Given nodes k_n, k_m corresponding to the numbers n, m, one must also check that a path from k_n to k_m does not contain more than $m - n$ strict inequalities. ■

Unfortunately, since $\mathcal{ALUEN}^S$ is not propositionally closed, subsumption cannot be reduced to satisfiability. A closer look at the specific form of the concept $C_\mathcal{D}$ introduced in Figure 9 reveals that it can be written as $C_\mathcal{D} = D_1 \sqcap \neg D_2$ for two $\mathcal{ALUEN}^S$-concepts D_1, D_2. In fact, D_1 is the first conjunct of $C_\mathcal{D}$ and D_2 is the negation of the remainder of $C_\mathcal{D}$. Note that D_1 does not contain numerical variables. Furthermore, all numerical variables occurring in the remainder of $C_\mathcal{D}$ are universally quantified, which shows that D_2 contains only existential quantification of numerical variables. Since $D_1 \sqcap \neg D_2$ is unsatisfiable iff $D_1 \sqsubseteq D_2$, this implies:

Theorem 22 Subsumption of $\mathcal{ALUEN}^S$-concepts is undecidable.

5 Related work

Some Modal Logics and Description Logics can be translated into first-order logic such that only two different variable names occur in the formulae obtained by this translation. Thus, decidability of subsumption and other inference problems for these logics follows from the known decidability result for $\mathcal{L}_2$, i.e., first-order logic with two variables and without function symbols [18, 10]. Recently, this decidability result has been extended to $\mathcal{C}_2$, i.e., first-order logic with 2 variables and counting quantifiers [11]. Independently, it has been proved in [19] that satisfiability of $\mathcal{C}_2$ formulae can be decided in nondeterministic doubly exponential time. As an immediate consequence, satisfiability and subsumption for $\mathcal{ALCN}(\sqcup, \sqcap, \neg, ^{-1})$, the extension of $\mathcal{ALC}$ by number restrictions with inversion and Boolean operators on roles, is still decidable. It should be noted, however, that expressing composition of roles in predicate logic requires more than two variables.

Using sophisticated techniques for translating Description Logic concepts into formulae of Propositional Dynamic Logics, it has been shown in [5] that deciding satisfiability and subsumption for a very expressive extension of $\mathcal{ALC}_{\mathsf{reg}}$ is ExpTime-complete. In particular, this extension allows for *qualifying* number restrictions on atomic and inverse roles, and thus it is an extension of the logic $\mathcal{ALC}_{\mathsf{reg}}\mathcal{N}(^{-1})$.

To the best of our knowledge, there are no (un)decidability or complexity results for logics that are similar to our DL with symbolic number restrictions.

6 Conclusion

The expressive power of traditional number restrictions is severely restricted for at least two reasons: only fixed non-negative integers may occur in number restrictions, and it is not possible to restrict the number of successors of a complex role. In this paper, we have tried to overcome these two restrictions by introducing two separate approaches for extending the expressiveness of number restrictions: symbolic number restrictions and number restrictions on complex roles. Although our goal was to obtain decidable Description Logics, it turned out that both types of extensions may easily cause undecidability.

For number restrictions on complex roles, we have considered extensions of $\mathcal{ALCN}$ and $\mathcal{ALC}_+\mathcal{N}$, and investigated decidability of the subsumption and the satisfiability problem. We could provide an almost complete classification of extensions of $\mathcal{ALCN}$ with number restrictions on complex roles, and a rather strong undecidability result for extensions of $\mathcal{ALC}_+\mathcal{N}$. Another inference problem of the decidable extension $\mathcal{ALCN}(\circ)$, namely checking the consistency of a concrete world description ("ABox-consistency"), was investigated in [17]. It was shown that consistency of ABoxes of a restricted form is decidable—whereas decidability of consistency of general $\mathcal{ALCN}(\circ)$-ABoxes is still an open problem.

To overcome the need to fix a non-negative integer in number restrictions, we introduced numerical variables to be used in number restrictions, where these variables can be existentially quantified. The propositionally closed extension (namely, the one that allows for full negation, and thus implicitly introduces universal quantification of numerical variables) turned out to be undecidable, whereas a restriction of this "full" extension to atomic negation turned out to have a decidable satisfiability problem. Unfortunately, the subsumption problem for this logic is still undecidable. The undecidability proof is also interesting from a theoretical point of view because symbolic number restrictions have the expressive power to enforce infinitely branching models, whereas the undecidability of other logics is usually due to the fact that infinite paths can be enforced.

Summing up, this paper almost completely answers the question how far the expressive power of number restrictions can be increased without losing decidability of the important inference problems. The decidable extensions, namely composition of roles in number restrictions and the decidable form of symbolic number restrictions, provide an expressive power that is useful in many applications, not only in the process engineering application that motivated this research.

References

[1] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, 1991.

[2] F. Baader. A formal definition for the expressive power of terminological knowledge representation languages. *Journal of Logic and Computation*, 6(1):33–54, 1996.

[3] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems, or: Making KRIS get a move on. *Applied Artificial Intelligence*, 4:109–132, 1994.

[4] R. Berger. The undecidability of the dominoe problem. *Mem. Amer. Math. Soc.*, 66, 1966.

[5] G. De Giacomo and M. Lenzerini. Tbox and Abox reasoning in expressive description logics. In *Proc. of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-96)*, pages 316–327. Morgan Kaufmann, Los Altos, 1996.

[6] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics (extended abstract). In *Proc. of the 12th Nat. Conf. on Artificial Intelligence (AAAI-94)*, 1994.

[7] G. De Giacomo and M. Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with mu-calculus. In *Proc. of the 11th European Conf. on Artificial Intelligence (ECAI-94)*, 1994.

[8] G. De Giacomo and M. Lenzerini. What's in an aggregate: Foundations for description logics with tuples and sets. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI-95)*, 1995.

[9] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134:1–58, 1997.

[10] E. Grädel, P. Kolaitis, and M. Vardi. On the Decision Problem for Two-Variable First-Order Logic. *Bulletin of Symbolic Logic*, 3:53–69, 1997.

[11] E. Grädel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *Proc. of the 12th Ann. IEEE Symp. on Logic in Computer Science (LICS-97)*, 1997. Available via `http://speedy.informatik.rwth-aachen.de/WWW/papers.html`.

[12] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-91)*, pages 335–346, Boston, MA, USA, 1991.

[13] B. Hollunder, W. Nutt, and M. Schmidt-Schauss. Subsumption algorithms for concept description languages. In *ECAI-90*, Pitman Publishing, London, 1990.

[14] D.E. Knuth. *The Art of computer programming*, volume 1. Addison Wesley Publ. Co., Reading, Massachussetts, 1968.

[15] N. Kurtonina and M. de Rijke. Classifying description logics. In M.-C. Rousset, R. Brachmann, F. Donini, E. Franconi, I. Horrocks, and A. Levy, editors, *Proceedings of the International Workshop on Description Logics*, Gif sur Yvette, France, 1997. Université Paris-Sud.

[16] R. MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[17] R. Molitor. Konsistenz von Wissensbasen in Beschreibungslogiken mit Rollenoperatoren. Diploma thesis, RWTH Aachen, Germany, 1997.

[18] M. Mortimer. On languages with two variables. *Zeitschr. f. math. Logik u. Grundlagen d. Math.*, 21:135–140, 1975.

[19] L. Pacholski, W. Szwast, and L. Tendera. Complexity of two-variable logic with counting. In *Proc. of the 12th Ann. IEEE Symp. on Logic in Computer Science (LICS-97)*, 1997.

[20] P. Patel-Schneider, D. McGuinness, R. Brachman, L. Resnick, and A. Borgida. The CLASSIC knowledge representation system: Guiding principles and implementation rationale. *SIGART Bulletin*, 2(3):108–113, 1991.

[21] C. Peltason. The BACK System - an Overview. *SIGART Bulletin*, 2(3):114–119, 1991.

[22] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 466–471, Sydney, 1991.

[23] K. Schild. Terminological cycles and the propositional μ-calculus. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 509–520, Bonn, 1994. Morgan Kaufmann, Los Altos.

[24] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

Appendix

Proof of Part 4 of Lemma 8 We must show that the tableau algorithm that tests satisfiability of $\mathcal{ALCN}(\circ)$-concepts always terminates. In the following, we consider only constraint systems S that are obtained by applying the completion rules to $\{x_0 : C_0\}$. For a concept C, we define its and/or-size $|C|_{\sqcap,\sqcup}$ as the number of occurrences of conjunction and disjunction constructors in C. The maximal role depth *depth*(C) of C is defined as follows:

$$
\begin{aligned}
&\mathit{depth}(A) := \mathit{depth}(\neg A) := 0 \text{ for } A \in N_C,\\
&\mathit{depth}(C_1 \sqcap C_2) := \mathit{max}\{\mathit{depth}(C_1), \mathit{depth}(C_2)\},\\
&\mathit{depth}(C_1 \sqcup C_2) := \mathit{max}\{\mathit{depth}(C_1), \mathit{depth}(C_2)\},\\
&\mathit{depth}(\forall R_1.C_1) := \mathit{depth}(\exists R_1.C_1) := 1 + \mathit{depth}(C_1),\\
&\mathit{depth}(\geq n\ R_1 \circ \ldots \circ R_m) := m,\\
&\mathit{depth}(\leq n\ R_1 \circ \ldots \circ R_m) := m.
\end{aligned}
$$

The following observations were made in Lemma 10:

1. *Every variable $x \neq x_0$ that occurs in S is an $R_1 \circ \ldots \circ R_m$-successor of x_0 for some role chain of length $m \geq 1$. In addition, every other role chain that connects x_0 with x has the same length.*

2. *If x can be reached in S by a role chain of length m from x_0, then for each constraint $x : C$ in S, the maximal role depth of C is bounded by the maximal role depth of C_0 minus m. Consequently, m is bounded by the maximal role depth of C_0.*

Let m_0 be the maximal role depth of C_0. Because of the first fact in Lemma 10, every individual x in a constraint system S (reached from $\{x_0 : C_0\}$ by applying completion rules) has a unique role level *level*(x), which is its distance from the root node x_0, i.e., the unique length of the role chains that connect x_0 with x. Because of the second fact, the level of each individual is an integer between 0 and m_0.

In the following, we define a mapping κ of constraint systems S to $5(m_0 + 1)$-tuples of non-negative integers such that $S \to S'$ implies $\kappa(S) \succ \kappa(S')$, where $\succ$ denotes the lexicographic ordering on $5(m_0+1)$-tuples. Since this lexicographic ordering is well-founded, this implies termination of our algorithm. In fact, if the algorithm did not terminate, then there would exist an infinite sequence $S_0 \to S_1 \to \ldots$, and this would yield an infinite descending $\succ$-chain of tuples.

Thus, let S be a constraint system that can be reached from $\{x_0 : C_0\}$ by applying completion rules. We define

$$\kappa(S) := (\kappa_0, \kappa_1, \ldots, \kappa_{m_0-1}, \kappa_{m_0}),$$

where $\kappa_\ell := (k_{\ell,1}, k_{\ell,2}, k_{\ell,3}, k_{\ell,4}, k_{\ell,5})$ and the components $k_{\ell,i}$ are obtained as follows:

- $k_{\ell,1}$ is the number of individual variables x in S with *level*$(x) = \ell$.
- $k_{\ell,2}$ is the sum of the and/or-sizes $|C|_{\sqcap,\sqcup}$ of all constraints $x : C \in S$ such that *level*$(x) = \ell$ and the conjunction or disjunction rule is applicable to $x : C$ in S.
- For a constraint $x :(\geq n\ R_1 \circ \ldots \circ R_m)$, let k be the maximal cardinality of all sets M of $R_1 \circ \ldots \circ R_m$-successors of x for which $y_i \neq y_j \in S$ for all pairs of distinct elements y_i, y_j of M. We associate with $x :(\geq n\ R_1 \circ \ldots \circ R_m)$ the number $r := n - k$, if $n \geq k$, and $r := 0$ otherwise. The component $k_{\ell,3}$ sums up all the numbers r associated with constraints of the form $x :(\geq n\ R_1 \circ \ldots \circ R_m)$ for variables x with *level*$(x) = \ell$.
- $k_{\ell,4}$ is the number of all constraints $x :(\exists R.C) \in S$ such that *level*$(x) = \ell$ and the existential restriction rule is applicable to $x :(\exists R.C)$ in S.
- $k_{\ell,5}$ is the number of all pairs of constraints $x :(\forall R.C)$, $xRy \in S$ such that *level*$(x) = \ell$ and the value restriction rule is applicable to $x :(\forall R.C)$, xRy in S.

In the following, we show for each of the rules of Figure 7 that $S \to S'$ implies $\kappa(S) \succ \kappa(S')$.

1. Conjunction: Assume that the rule is applied to the constraint $x : C_1 \sqcap C_2$, and let S' be the system obtained from S by its application. Let $\ell :=$ *level*(x).
First, we compare κ_ℓ and κ'_ℓ, the tuples respectively associated with level ℓ in S and S'. Obviously, the *first components* of κ_ℓ and κ'_ℓ agree since the number of individuals and their levels are not changed. The *second component* of κ'_ℓ is *smaller* than the second component of κ_ℓ: $|C_1 \sqcap C_2|_{\sqcap,\sqcup}$ is removed from the sum, and replaced by a number that is not larger than $|C_1|_{\sqcap,\sqcup} + |C_2|_{\sqcap,\sqcup}$ (depending on whether the top constructor of C_1 and C_2 is disjunction or conjunction, or some other constructor). Since tuples are compared with the lexicographic ordering, a decrease in this component makes sure that it is irrelevant what happens in later components.
For the same reason, we need not consider tuples κ_m for $m > \ell$. Thus, assume that $m < \ell$. In such a tuple, the first three components are not changed by application of the rule, whereas the remaining two components remain unchanged or decrease. Such a decrease can happen if *level*$(y) = m$ and S contains constraints yRx, $y :(\forall R.C_i)$ (or $y :(\exists R.C_i)$).

2. Disjunction: This rule can be treated like the conjunction rule.

3. Value restriction: Assume that the rule is applied to the constraints $x:(\forall R.C), xRy$, and let S' be the system obtained from S by its application. Let $\ell := \textit{level}(x)$. Obviously, this implies that $\textit{level}(y) = \textit{level}(x) + 1 > \ell$.
On level ℓ, the first three components of κ_ℓ remain unchanged; the fourth remains the same, or decreases (if S contains constraints zSy and $z:(\exists S.C)$ for an individual z with $\textit{level}(z) = \ell$); and the fifth decreases by at least one since the constraints $x:(\forall R.C), xRy$ are no longer counted. It may decrease by more than one if S contains constraints zSy and $z:(\forall S.C)$ for an individual z with $\textit{level}(z) = \ell$.
Because of this decrease at level ℓ, the tuples at larger levels (in particular, the one for level $\textit{level}(x) + 1$, where there might be an increase), need not be considered.
The tuples of levels smaller than ℓ are not changed by application of the rule. In particular, the third component of such a tuple does not change since no role constraints or inequality constraints are added or removed.

4. Existential restriction: Assume that the rule is applied to the constraint $x:(\exists R.C)$, and let $S' = S \cup \{xRy, y:C\}$ be the system obtained from S by its application. Let $\ell := \textit{level}(x)$. Obviously, this implies that $\textit{level}(y) = \textit{level}(x) + 1 > \ell$.
The first two components of κ_ℓ obviously remain unchanged. The third component may decrease (if y is the first successor for an at-least restriction) or it stays the same. Since the fourth component decreases, the possible increase of the fifth component is irrelevant.
For the same reason, the increase of the first component of $\kappa_{\ell+1}$ is irrelevant.
Tuples of levels smaller than ℓ are not increased by application of the rule. All components of such a tuple remain unchanged, with the possible exception of the third component, which may decrease.

5. Number restriction: Assume that the rule is applied to the constraint $x:(\geq n\ R_1 \circ \ldots \circ R_m) \in S$, let S' be the system obtained by rule application, and let $\ell = \textit{level}(x)$.
Similar to Rule 4, the first two components of κ_ℓ remain the same. In addition, there is a decrease in the third component of κ_ℓ, since the new individual z can now be added to the maximal sets of explicitly distinct $R_1 \circ \ldots \circ R_m$-successors of x. Note that these sets were previously smaller than n (because even the set of all $R_1 \circ \ldots \circ R_m$-successors of x was smaller than n).
For this reason, the possible increase in the fifth component of κ_ℓ and in the first components of tuples of levels larger than ℓ are irrelevant.Tuples of levels smaller than ℓ are either unchanged by application of the rule, or their third component decreases.

6. Number restriction: Assume that the rule is applied to the constraint $x:(\leq n\ R_1 \circ \ldots \circ R_m) \in S$, let $S' = S_{y_1,y_2}$ be the system obtained by rule application, and let $\ell = \textit{level}(x)$.
On level $\ell + m$, the first component of the tuple $\kappa_{\ell+m}$ decreases. Thus, possible increases in the other components of this tuple are irrelevant.
Tuples associated with smaller levels remain unchanged or decrease. In fact, since y_1 in S' has all its old constraints and the constraints of y_2 in S, some value restrictions or existential restrictions for individuals of the level immediately above level $\ell + m$ may become satisfied (in the sense that the corresponding rule no longer applies). Since

no constraints are removed, previously satisfied value restrictions or existential restrictions remain satisfied. The third component of tuples of smaller level cannot increase since the individuals y_1, y_2 that have been identified were not related by inequality constraints. ■

Proof of Lemma 16 We must show that $C_{\mathcal{D}}$ is satisfiable iff there exists a compatible tiling of the second eighth of the plane using $\mathcal{D}$. Note that the definition of $C_{\mathcal{D}}$ obviously implies that $C_{\mathcal{D}}$ is subsumed by $C_{\mathbb{N}}$, and thus Lemma 15 applies to instances of $C_{\mathcal{D}}$.

"⇒" Given a model $\mathcal{I}$ of $C_{\mathcal{D}}$ with $o \in C_{\mathcal{D}}{}^{\mathcal{I}}$, we define the mapping $t\colon (\mathbb{N} \times \mathbb{N})_{\leq} \to D$ as follows:

$$t(a, b) = D_i \text{ iff } o \in (\exists S.((= a\ L) \sqcap (= b\ R) \sqcap D_i))^{\mathcal{I}}.$$

First, we show that t is well-defined. Thus, let $a, b \in \mathbb{N}$. Since

$$o \in (\forall S.(\bigsqcup_{1 \leq i \leq m} (D_i \sqcap (\bigsqcap_{\substack{1 \leq j \leq m \\ i \neq j}} \neg D_j))))^{\mathcal{I}},$$

each S-successor of o is an instance of exactly one $D_i \in D$. For each $(a, b) \in (\mathbb{N} \times \mathbb{N})_{\leq}$ and each $D_i \in D$

$$o \in ((\exists S.((= a\ L) \sqcap (= b\ R) \sqcap D_i)) \Rightarrow (\forall S.((\neq a\ L) \sqcup (\neq b\ R) \sqcup D_i)))^{\mathcal{I}},$$

which implies that all S-successors of o having the same number of L-successors and the same number of R-successors are instances of the same $D_i \in D$. Thus t is well-defined, and it remains to be shown that t is indeed a compatible tiling.

Let $a, b \in \mathbb{N}$, $a < b$ and $t(a, b) = D_i$. From Lemma 15.2.(i) it follows that $o \in (\exists S.((= a\ L) \sqcap (= b\ R)))^{\mathcal{I}}$ and we have already seen that each S-successor of o is an instance of exactly one $D_i \in D$; hence $o \in (\exists S.((= a\ L) \sqcap (= b\ R) \sqcap D_i))^{\mathcal{I}}$ for some D_i. Now $o \in C_{\mathcal{D}}{}^{\mathcal{I}}$ implies that

$$o \in (\uparrow\gamma.((<(a, b) \sqcap =(a+1, \gamma)) \Rightarrow (\exists S.((= \gamma\ L) \sqcap (= \beta\ R) \sqcap D_j))))^{\mathcal{I}}$$

for some D_j with $(D_i, D_j) \in H$. Hence $o \in (\exists S.((= a+1\ L) \sqcap (= b\ R) \sqcap D_j))^{\mathcal{I}}$, which implies that $t(a+1, b) = D_j$ and $(D_i, D_j) \in H$.

Now let $a, b \in \mathbb{N}$ with $a \leq b$ and $t(a, b) = D_i$. Then again $o \in (\exists S.((= a\ L) \sqcap (= b\ R) \sqcap D_i))^{\mathcal{I}}$, and $o \in C_{\mathcal{D}}{}^{\mathcal{I}}$ implies that

$$o \in (\uparrow\gamma.((= (b+1, \gamma)) \Rightarrow (\exists S.((= a\ L) \sqcap (= \gamma\ R) \sqcap D_j))))^{\mathcal{I}}$$

for some D_j with $(D_i, D_j) \in V$. Hence $o \in (\exists S.((= a\ L) \sqcap (= b+1\ R) \sqcap D_j))^{\mathcal{I}}$, which implies that $t(a, b+1) = D_j$ and $(D_i, D_j) \in V$. To sum up, we have shown that t is a tiling.

"⇐" Given a tiling t, we define a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $C_{\mathcal{D}}$ as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}} &:= \{o\} \uplus \{y_{a,b} \mid a, b \in \mathbb{N} \text{ and } a \leq b\} \uplus \{\ell_a, r_b \mid a, b \in \mathbb{N}\}, \\
S^{\mathcal{I}} &:= \{(o, y_{a,b}) \mid a, b \in \mathbb{N} \text{ and } a \leq b\}, \\
L^{\mathcal{I}} &:= \{(y_{a,b}, \ell_{a'}) \mid a, a', b \in \mathbb{N} \text{ and } a' < a \leq b\}, \\
R^{\mathcal{I}} &:= \{(y_{a,b}, r_{b'}) \mid a, b, b' \in \mathbb{N} \text{ and } a \leq b \text{ and } b' < b\}, \\
D_i^{\mathcal{I}} &:= \{y_{a,b} \mid t(a, b) = D_i\}
\end{aligned}$$

By definition of $D_i^{\mathcal{I}}$, each S-successor of o is an instance of exactly one $D_i \in D$, and hence

$$o \in (\forall S.(\bigsqcup_{1 \le i \le m} (D_i \sqcap (\bigsqcap_{\substack{1 \le j \le m \\ i \ne j}} \neg D_j))))^{\mathcal{I}}.$$

The interpretation $\mathcal{I}$ defined above just extends the one constructed in the proof of Lemma 15.1 by the interpretation of the atomic concepts D_i. Thus, Lemma 15.1 yields $o \in (\uparrow\alpha.\uparrow\beta.(C_1 \sqcap C_2 \sqcap C_3))^{\mathcal{I}}$.

Now let $a, b \in \mathbb{N}$. Then $o \in ((\exists S.((=\alpha\ L) \sqcap (=\beta\ R) \sqcap D_i))[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ iff $a \le b$ and $t(a,b) = D_i$.

For all a, b such that $o \in ((\exists S.((=\alpha\ L) \sqcap (=\beta\ R) \sqcap D_i))[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$, we must show that o also belongs to the concepts on the right-hand side of the implication (see lines (1), (2), (3) in Figure 4).

- $o \in (\forall S.((\ne \alpha\ L) \sqcup (\ne \beta\ R) \sqcup D_i)[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ since o has exactly one S-successor $y_{a,b} \in \Delta^{\mathcal{I}}$ having a L-successors and b R-successors, and for this S-successor $y_{a,b}$ we know $y_{a,b} \in D_i^{\mathcal{I}}$ by assumption.
- If $o \in ((<(\alpha,\beta) \sqcap =(\alpha+1,\gamma))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some $g \in \mathbb{N}$, then $a < b$ and $a + 1 = g$. The definition of $\mathcal{I}$ and the fact that t is a compatible tiling entail that $o \in (\exists S.((=\gamma\ L) \sqcap (=\beta\ R) \sqcap D_j)[\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some D_j with $(D_i, D_j) \in H$, and hence $o \in ((<(\alpha,\beta) \sqcap =(\alpha+1,\gamma)) \Rightarrow (\exists S.((=\gamma\ L) \sqcap (=\beta\ R) \sqcap \sqcup_{j \in H(D_i)} D_j))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$.
- If $o \in (=(\beta+1,\gamma))[\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some $g \in \mathbb{N}$, then $b + 1 = g$. Again, the definition of $\mathcal{I}$ and the fact that t is a compatible tiling entail that $o \in (\exists S.((=\alpha\ L) \sqcap (=\gamma\ R) \sqcap D_j)[\frac{a}{\alpha}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some D_j with $(D_i, D_j) \in V$, and hence $o \in ((=(\beta+1,\gamma)) \Rightarrow (\exists S.((=\alpha\ L) \sqcap (=\gamma\ R) \sqcap \sqcup_{j \in V(D_i)} D_j))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$.

To sum up, we have shown that $o \in {C_{\mathcal{D}}}^{\mathcal{I}}$, and thus $C_{\mathcal{D}}$ is satisfiable. ∎