RWTH
LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

Description Logics with Symbolic Number Restrictions

Franz Baader and Ulrike Sattler

LTCS-Report 96-03

This is an extended version of the article in: Proceedings of the 12th European Conference on Artificial Intelligence, 1996, Budapest, Hungary.

RWTH Aachen
LuFg Theoretische Informatik
http://www-lti.informatik.rwth-aachen.de

Ahornstr. 55
52074 Aachen
Germany

Description Logics with Symbolic Number Restrictions

Franz Baader and Ulrike Sattler

1 Motivation and Introduction

Terminological knowledge representation systems (TKR systems) are powerful tools not only to represent but also to reason about the knowledge on the terminology of an application domain. Their particular power lie in their ability to infer implicit knowledge from the knowledge explicitly stored in a knowledge base. Mainly, a TKR system consists of three parts: First, a terminological knowledge base which contains the explicit description of the concepts relevant for the application domain. Second, an assertional knowledge base which contains the description of concrete individuals and their relations. This description of concrete individuals is realized using the terminology fixed in the terminological knowledge base. Third, a TKR system comprises an inference engine which is able to infer implicit properties of the defined concepts and individuals such as

- subclass/superclass relations amongst concepts (subsumption),
- the classification of all defined concepts with respect to the subclass/superclass relation. This yields the class taxonomy.
- whether there exists an interpretation of the terminology where a given concept has at least one instance (satisfiability),
- to enumerate all individuals that are instances of a given concept (retrieval),

- given a concrete individual, to enumerate the most specific concepts of the terminology this individual is an instance of.

The language underlying a TKR system–the one that is used to build up those knowledge bases–is called a *concept language*. The more expressive power a concept language has, the more complex is the problem of inferring implicit from explicit knowledge. As satisfiability is the central problem to which–for most concept languages–the other ones can be reduced, it is subject to a great variety of investigations ([Donini *et al.*,1991b; 1991a; Baader and Hollunder,1991; 1991]). Most investigations of the complexity of deciding satisfiability of concepts consider *worst-case complexity* which turns out to be intractable for a large number of systems. Nevertheless, it could be shown that worst-case intractable languages may behave quite well in practice [Baader *et al.*,1994], which led to the use of expressive application-relevant operators that do not cause undecidability, but may increase the worst-case complexity.

To find–for a given application–a concept language which has enough expressive power to describe relevant properties of concepts of the application domain, but for which the inferences can be drawn using a reasonable amount of space or time is a difficult but important problem to solve.

Trying to describe concepts in an engineering application [Marquardt,1994] we observed that for the proper definition of complex objects–which are mainly determined by their components–it is important to describe not only which components an aggregate has, but also how many of them. This can be done using traditional number restrictions, [Hollunder *et al.*,1990; Donini *et al.*,1991a], if exact numbers are known. For example the following concept describes aggregates having five inputs and five outputs

$$\texttt{device} \sqcap (= 5\ \texttt{input}) \sqcap (= 5\ \texttt{output}).$$

Unfortunately an exact number has to be fixed[1], for example it is not possible to describe devices having the same number of inputs and outputs using traditional number restrictions. The extensions of $\mathcal{ALCN}$ presented here allow for using numerical variables inside number restrictions which

[1] As pointed out in [Levesque and Brachman,1987], an important aspect of expressiveness is, however, "what can be left unsaid" in a representation.

reach over the set of nonnegative integers. Substituting 5 by the numerical variable α in the above mentioned example yields a concept which describes aggregates having the same number of inputs and outputs. Any of these variables can be used at different levels of nested concepts, as for example in

$$\texttt{device} \sqcap (= \alpha\ \texttt{property}) \sqcap (\forall \texttt{implementat}.(\geq \alpha\ \texttt{parameter})),$$

which describes devices where each of their implementations has enough parameters to describe their properties.

Additionally, we allow explicit quantification of these variables using $\downarrow$. This quantification allows to describe both a device where for each component holds that the number of its inputs equals the number of its outputs as in (1) as well as devices where all components have the same number of inputs and outputs as in (2):

$$\texttt{device} \sqcap (\forall \texttt{component}.(\downarrow \alpha.(= \alpha\ \texttt{input}) \sqcap (= \alpha\ \texttt{output}))), \quad (1)$$

$$\texttt{device} \sqcap (\downarrow \alpha.(\forall \texttt{component}.(= \alpha\ \texttt{input}) \sqcap (= \alpha\ \texttt{output}))). \quad (2)$$

In the following, three extensions of $\mathcal{ALCN}$ by symbolic number restrictions are presented: a general extension and two different sub-languages. It is shown that satisfiability and subsumption of concepts of the general extension and one of its sub-languages is undecidable, whereas the other sub-language has a decidable satisfiability problem and an undecidable subsumption problem. Furthermore, we discuss the consequences of these observations for the use of symbolic number restrictions in TKR systems.

2 Preliminaries

The concept language underlying this investigation is $\mathcal{ALCN}$, a well–known concept language with strong expressive power. In this concept language, which was introduced by [Hollunder *et al.*,1990; Donini *et al.*,1991a; 1995], concepts can be built using boolean operators on concepts, number restriction and value restrictions.

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

- every concept name is a concept.
- if C and D are concepts, R is a role name and $n \in \mathbb{N}$, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$, $(\leq n\ R)$ and $(\geq n\ R)$ are concepts.

In the following, let rel $\in \{=, <, >, \leq, \geq\}$ and let $\#X$ denote the size of a set X.

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 a function $\cdot^\mathcal{I}$ which maps every concept to a subset of $\Delta^\mathcal{I}$ and every role to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ such that

$$\begin{aligned}
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
\neg C^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(\exists R.C)^\mathcal{I} &= \{d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I} : (d, e) \in R^\mathcal{I} \wedge e \in C^\mathcal{I}\} \\
(\forall R.C)^\mathcal{I} &= \{d \in \Delta^\mathcal{I} \mid \forall e \in \Delta^\mathcal{I} : (d, e) \in R^\mathcal{I} \Rightarrow e \in C^\mathcal{I}\} \\
(\mathrm{rel}\, n\ R)^\mathcal{I} &= \{d \in \Delta^\mathcal{I} \mid \#\{e \in \Delta^\mathcal{I} \mid (d, e) \in R^\mathcal{I}\}\, \mathrm{rel}\, n\}
\end{aligned}$$

The use of the general predicate “rel” in number restriction is appropriate although in the standard definition of $\mathcal{ALCN}$-concepts only $\geq$ and $\leq$ are allowed because of the following equivalences:

$$\begin{aligned}
(\neg(\leq n\ R)) &= (> n\ R) \\
(= n\ R) &= ((\leq n\ R) \sqcap (\geq n\ R)) \text{ etc.}
\end{aligned}$$

Other boolean operators will be used as abbreviations, as for example $A \Rightarrow B$ is short for $\neg A \sqcup B$. A concept C is called *satisfiable* iff there is some interpretation $\mathcal{I}$ such that $C^\mathcal{I} \neq \emptyset$. Such an interpretation is called a *model of* C. A concept D *subsumes* a concept C (written $C \sqsubseteq D$) iff for each interpretation $\mathcal{I}$ it holds that $C^\mathcal{I} \subseteq D^\mathcal{I}$. Two concepts C, D are said to be *equivalent* iff $C \sqsubseteq D$ and $D \sqsubseteq C$. If $x \in C^\mathcal{I}$, we say that x is an *instance of* C in $\mathcal{I}$.

3 The extension of $\mathcal{ALCN}$ by symbolic number restrictions

3.1 Definitions

Concepts of $\mathcal{ALCN}^S$ differ from $\mathcal{ALCN}$-concepts in that in $\mathcal{ALCN}^S$-concepts numerical variables are allowed inside number restrictions and that these variables can be quantified explicitly.

Definition 3 Let N_C be a set of *concept names*, N_R a set of *role names* and N_V a set of variables. The set of *concepts* of $\mathcal{ALCN}^S$ is the smallest set such that

- every concept name is a concept.
- if C and D are concepts, R is a role name, α is a variable and $n \in \mathbb{N}$ a nonnegative integer, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg C)$, $(\forall R.C)$, $(\exists R.C)$, $(\downarrow \alpha.C)$, $(\mathrm{rel}\, \alpha\ R)$, $(\mathrm{rel}\, n\ R)$ are concepts.

Concepts of the form $(\mathrm{rel}\, \alpha\ R)$ or $(\mathrm{rel}\, n\ R)$ are called number restrictions.

Before presenting the semantics of $\mathcal{ALCN}^S$-concepts, three examples are presented to give an intuitive idea of the meaning of these new operators and their expressive power.

Example 1 The first concept describes aggregates having more connections than devices as parts (by $\downarrow$, numerical variables are quantified existentially):

$$(\downarrow \alpha.(= \alpha\ \texttt{has\_device}) \sqcap (> \alpha\ \texttt{has\_connection})). \tag{3}$$

Without symbolic number restrictions, one could express that an aggregate has 1 device and more than 1 connections or 2 devices and more than 2 connections or... and had to stop this disjunction at some number n whereas the variable α reaches over all nonnegative integers.

The next concept describes aggregates whose components all have the same number of inputs and the same number of outputs:

$$(\downarrow \alpha \downarrow \beta.(\forall \texttt{component}.(= \alpha\ \texttt{input}) \sqcap (= \beta\ \texttt{output}))). \tag{4}$$

Since variables are quantified explicitly, their scope varies according to where this quantification occurs: commuting $\downarrow \beta$ and $\forall$`component` in 4 yields a different concept.

The last concept shows how variables can be used at different levels of nested concepts. Each device has several context-dependent implementations to describe its behaviour, each of which has to have enough parameters to describe the properties of the device:

$$\texttt{device} \sqcap (\downarrow \alpha.(= \alpha\ \texttt{property}) \sqcap (\forall \texttt{implementat}.(\geq \alpha\ \texttt{param}))) \tag{5}$$

Some notation has to be defined now:

Definition 4 Let $\text{sub}(C)$ denote the set of all subconcepts of a concept C. The occurrence of a variable $\alpha \in N_V$ is said to be *bound in* C iff α occurs inside the scope C' of a sub-term $(\downarrow \alpha.C') \in \text{sub}(C)$. Otherwise, the occurrence is said to be *free*. Please note that a variable can occur free and bound in a concept, as α in $((= \alpha\ R) \sqcap (\downarrow \alpha.(\exists R.(> \alpha\ R))))$. The set $\text{free}(C) \subseteq N_V$ denotes the set of variables which occur free in C. A concept C is *closed* iff $\text{free}(C) = \emptyset$. The concept $C[\frac{n}{\alpha}]$ is obtained from a concept C by substituting all free occurrences of α by n. For a role name R and some $x \in \Delta^{\mathcal{I}}$, let ${x_R}^{\mathcal{I}}$ denote the number of role fillers of x with respect to R in $\mathcal{I}$, this is

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

Semantics of $\mathcal{ALCN}^S$-concepts is defined as follows.

Definition 5 An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a closed $\mathcal{ALCN}^S$-concept is an interpretation as defined for $\mathcal{ALCN}$-concepts, which additionally satisfies

$$(\downarrow \alpha.C)^{\mathcal{I}} = \bigcup_{n \in \mathbb{N}} (C[\frac{n}{\alpha}])^{\mathcal{I}} \tag{6}$$

Semantics of concepts which are not closed is defined as follows:

$$\text{If free}(C) = \{\alpha_1, \ldots, \alpha_n\} \text{ for } n \geq 1, \text{ then } C^{\mathcal{I}} = (\downarrow \alpha_1. \ldots \downarrow \alpha_n.C)^{\mathcal{I}}.$$

A concept of the form $(\downarrow \alpha.C)$ is the only one where we do not have a non–negated concept for its complement. We will use $(\uparrow \alpha.C)$ as shorthand for $\neg(\downarrow \alpha.\neg C)$ and for each interpretation $\mathcal{I}$ we have

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

3.2 Satisfiability of $\mathcal{ALCN}^S$-concepts is undecidable

In the following, it will be shown that satisfiability of $\mathcal{ALCN}^S$-concepts is undecidable by reducing one of the classical domino problems to it. A direct consequence of this result is that subsumption of $\mathcal{ALCN}^S$-concepts is also undecidable.

Let $D = \{D_1, \ldots, D_m\}$ be a non-empty set of *domino types* and let $H \subseteq D \times D$, $V \subseteq D \times D$ define the horizontal and the vertical matching pairs of domino types. Then the triple $\mathcal{D} = (D, H, V)$ is called a *tiling system*. It has been shown [Berger,1966; Wang,1963] that, for a given tiling system, the problem whether the plane can be tiled using these domino types, i.e. whether there exists a mapping $t : \mathbb{N} \times \mathbb{N} \to D$ such that

$$\begin{aligned} \text{for all } (n,m) \in \mathbb{N} \times \mathbb{N} \text{ holds } (t(n,m), t(n+1,m)) &\in H \\ \text{and } (t(n,m), t(n,m+1)) &\in V \end{aligned}$$

is undecidable. There are several variants of this result, and for simplicity of our proof we will use a slightly modified result which states that, for a given tiling system $\mathcal{D}$, the question whether there exists a tiling for the second octant of the plane

$$(\mathbb{N} \times \mathbb{N})_{\leq} := \{(a,b) \mid a, b \in \mathbb{N} \text{ and } a \leq b\},$$

using $\mathcal{D}$, i.e. whether there exists a mapping $t : (\mathbb{N} \times \mathbb{N})_{\leq} \to D$ such that

$$\begin{aligned} \text{for all } (n,m) \in (\mathbb{N} \times \mathbb{N})_{\leq} \text{ with } n < m \text{ holds } (t(n,m), t(n+1,m)) &\in H \\ \text{and for all } (n,m) \in (\mathbb{N} \times \mathbb{N})_{\leq} \text{ holds } (t(n,m), t(n,m+1)) &\in V \end{aligned}$$

is undecidable.

The reduction works as follows: First, an $\mathcal{ALCN}^S$-concept $C_{\mathbb{N}}$ is defined in such a way that there is a natural injective mapping from $(\mathbb{N} \times \mathbb{N})_{\leq}$ to each model of $C_{\mathbb{N}}$. Second, given a tiling system $\mathcal{D}$, this concept $C_{\mathbb{N}}$ is used to construct a concept $C_{\mathcal{D}}$ in such a way that $C_{\mathcal{D}}$ is satisfiable iff there exists a tiling of $(\mathbb{N} \times \mathbb{N})_{\leq}$ using $\mathcal{D}$. As a consequence, satisfiability of $\mathcal{ALCN}^S$-concepts is undecidable.

$C_{\mathbb{N}}$ is defined in such a way that each $x \in C_{\mathbb{N}}{}^{\mathcal{I}}$ has infinitely many S-successors and each point $(n, m) \in (\mathbb{N} \times \mathbb{N})_{\leq}$ is represented by at least one of these S-successors of x, where an S-successors of x is said to represent a point $(n, m) \in (\mathbb{N} \times \mathbb{N})_{\leq}$ if it has n L-successors and m R-successors:

$$\begin{array}{rl} C_{\mathbb{N}} := (\uparrow \alpha. \uparrow \beta. & (\exists S.(= \alpha\ L)) \sqcap \\ & ((\exists S.(= \alpha\ L) \sqcap (\leq \beta\ L)) \Rightarrow (\exists S.(= \alpha\ L) \sqcap (= \beta\ R))) \sqcap \\ & (\forall S.((= \alpha\ L) \sqcap (= \beta\ R)) \Rightarrow (\leq \beta\ L))) \end{array}$$

In what follows, we will use the abbreviation C_1, C_2, C_3 for the three subconcepts of $C_{\mathbb{N}} = (\uparrow \alpha. \uparrow \beta. C_1 \sqcap C_2 \sqcap C_3)$.

Lemma 6 1. $C_{\mathbb{N}}$ is satisfiable.

2. Let $\mathcal{I}$ be a model of $C_{\mathbb{N}}$ with $x \in C_{\mathbb{N}}^{\mathcal{I}}$ and let $Y = \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in S^{\mathcal{I}}\}$.
   (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 $x \in C_{\mathbb{N}}{}^{\mathcal{I}}$, then there is an injective mapping $\phi : (\mathbb{N} \times \mathbb{N})_{\leq} \to Y$ from the second octant of the plane to the set of S-successors of x.

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

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

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

$x \in C_{\mathbb{N}}{}^{\mathcal{I}}$ iff for all $a, b \in \mathbb{N}$: $x \in (C_1[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$, $x \in (C_2[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ and $x \in (C_3[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$. Let $a, b \in \mathbb{N}$, then

- $x \in (C_1[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ since $(x, y_{a,b'}) \in S^{\mathcal{I}}$ for some $b' \geq a$.
- $x \in (C_2[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$: If $x \in (\exists S.(= a\ L) \sqcap (\leq b\ L))^{\mathcal{I}}$, then $a \leq b$ and $(x, y_{a,b}) \in S^{\mathcal{I}}$, which implies $x \in (\exists S.(= a\ L) \sqcap (= b\ R))^{\mathcal{I}}$.

- $x \in (C_3[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$: Let $(x,y) \in S^{\mathcal{I}}$. 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 $y \in Y$ with ${y_L}^{\mathcal{I}} = a$. C_2 ensures for all $a, b \in \mathbb{N}$ that, if $y \in Y$ with ${y_L}^{\mathcal{I}} = a$ and ${y_L}^{\mathcal{I}} \leq b$ (which implies $a \leq b$) then there is some $y' \in Y$ with ${y'_L}^{\mathcal{I}} = a$ and ${y'_R}^{\mathcal{I}} = b$.

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

3. This is a direct consequence of 2. (i): $\phi : (a,b) \mapsto y$ with ${y_L}^{\mathcal{I}} = a$ 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 ${y_R}^{\mathcal{I}} = b$.

Definition 7 Given a tiling system $\mathcal{D} = (\{D_1, \ldots, D_m\}, H, V)$, let

$$
\begin{array}{l}
C_{\mathcal{D}} := \\
(\forall S.(\sqcup_{1\leq i\leq m}(D_i \sqcap (\sqcap_{\substack{1\leq j\leq m\\ i\neq j}} \neg D_j))))\sqcap \\
(\uparrow \alpha.\uparrow \beta.C_1 \sqcap C_2 \sqcap C_3 \sqcap \\
\quad \sqcap_{1\leq i\leq m}(\exists S.((= \alpha\ \ L) \sqcap (= \beta\ R) \sqcap D_i)) \Rightarrow \\
\quad\quad ((\forall S.((\neq \alpha\ L) \sqcup (\neq \beta\ R) \sqcup D_i))\sqcap \\
\quad\quad\ (\uparrow \gamma.({<}(\alpha,\beta) \sqcap {=}(\alpha+1,\gamma)) \Rightarrow \\
\quad\quad\quad (\exists S.((= \gamma\ L) \sqcap (= \beta\ R) \sqcap \sqcup_{j\in H(D_i)} D_j))\sqcap \\
\quad\quad\ (\uparrow \gamma.({=}(\beta+1,\gamma) \Rightarrow \\
\quad\quad\quad (\exists S.(= \alpha\ L) \sqcap (= \gamma\ R) \sqcap \sqcup_{j\in V(D_i)} D_j)))))),
\end{array}
$$

where C_i are the subconcepts of $C_{\mathbb{N}}$ and the following abbreviations are used:

$$
\begin{array}{rcl}
{<}(\alpha,\beta) & := & (\exists S.((= \alpha\ L) \sqcap (= \beta\ R) \sqcap \neg(= \beta\ L))), \\
{=}(\alpha+1,\beta) & := & {<}(\alpha,\beta) \sqcap (\forall S.((\leq \alpha\ L) \sqcup (\geq \beta\ L))), \\
j \in H(D_i) & \text{iff} & (D_i, D_j) \in H, \\
j \in V(D_i) & \text{iff} & (D_i, D_j) \in V.
\end{array}
$$

Remark: From Lemma 6.2 follows directly that for $x \in {C_\mathbb{N}}^\mathcal{I}$ we have $x \in (<(\alpha, \beta)[\frac{a}{\alpha}][\frac{b}{\beta}])^\mathcal{I}$ iff $a < b$. Since $x \in {C_\mathbb{N}}^\mathcal{I}$ has some S-successor having a L-successors for each $a \in \mathbb{N}$, $x \in (=(\alpha+1, \beta)[\frac{a}{\alpha}][\frac{b}{\beta}])^\mathcal{I}$ iff $a + 1 = b$.

Lemma 8 $C_\mathcal{D}$ is satisfiable iff there exists a tiling of the first octant of the plane using $\mathcal{D}$.

Proof: From the definition of $C_\mathcal{D}$ follows immediately that $C_\mathcal{D}$ is subsumed by $C_\mathbb{N}$.

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

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

It has to be shown that t is indeed a tiling: Let $a, b \in \mathbb{N}$. Since

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

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

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

hence all S-successors of x having the same number of L-successors and the same number of R-successors are instances of the same $D_i \in D$. Together, this implies that $t : (\mathbb{N} \times \mathbb{N})_\leq \to D$ is well-defined. It remains to be shown that t is a tiling:

Let $a, b \in \mathbb{N}$, $a < b$ and $t(a, b) = D_i$. From Lemma 6.2.(i) follows that $x \in (\exists S.((= a\ L) \sqcap (= b\ R)))^\mathcal{I}$ and we have already seen that each S-successor of x is an instance of exactly one $D_j \in D$, hence $x \in (\exists S.((= a\ L) \sqcap (= b\ R) \sqcap D_i))^\mathcal{I}$ for some D_i. From $x \in {C_\mathcal{D}}^\mathcal{I}$ follows that

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

for some D_j with $(D_i, D_j) \in H$, hence

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

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

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

and from $x \in C_{\mathcal{D}}{}^{\mathcal{I}}$ follows that

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

for some D_j with $(D_i, D_j) \in V$, hence

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

which implies that $t(a, b+1) = D_j$ and $(D_i, D_j) \in V$. Hence t is a tiling.

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

$$\begin{aligned}
\Delta^{\mathcal{I}} &:= \{x\} \uplus \{y_{a,b} \mid a,b \in \mathbb{N} \text{ and } a \leq b\} \uplus \{l_a, r_b \mid a,b \in \mathbb{N}\},\\
S^{\mathcal{I}} &:= \{(x, y_{a,b}) \mid a,b \in \mathbb{N} \text{ and } a \leq b\},\\
L^{\mathcal{I}} &:= \{(y_{a,b}, l_{a'}) \mid a,a',b \in \mathbb{N} \text{ and } a' < a \leq b\},\\
R^{\mathcal{I}} &:= \{(y_{a,b}, r_{b'}) \mid a,b,b' \in \mathbb{N} \text{ and } a \leq b \text{ and } b' < b\},
\end{aligned}$$

$$\begin{aligned}
x &\in D_1,\\
\text{for all } a \in \mathbb{N},\ l_a &\in D_1,\\
\text{for all } a \in \mathbb{N},\ r_a &\in D_1,\\
y_{a,b} &\in D_i \text{ iff } t(a,b) = D_i.
\end{aligned}$$

Each S-successor of x is instance of exactly one $D_i \in D$, hence

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

Since the interpretation $\mathcal{I}$ is the same as given in the proof of Lemma 6.1 beside the interpretation of $D_i \in D$, it is clear that

$$x \in (\uparrow \alpha. \uparrow \beta. C_1 \sqcap C_2 \sqcap C_3)^{\mathcal{I}}.$$

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

Let $x \in ((\exists S.((= \alpha\ L) \sqcap (= \beta\ R) \sqcap D_i))[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$, then

- $x \in (\forall S.((\neq \alpha\ L) \sqcup (\neq \beta\ R) \sqcup D_i)[\frac{a}{\alpha}][\frac{b}{\beta}])^{\mathcal{I}}$ since x has only one single S-successor $y_{a,b} \in \Delta^{\mathcal{I}}$ having a L-successors and b R-successors, and for this $y_{a,b}$ we defined $y_{a,b} \in D_i$.
- if $x \in ((<(\alpha, \beta) \sqcap =(\alpha+1, \gamma))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$, then $a < b$ and $a+1 = g$, and from the definition of $\mathcal{I}$ follows that $x \in (\exists S.((= \gamma\ L) \sqcap (= \beta\ R) \sqcap D_j)[\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some D_j with $(D_i, D_j) \in H$, hence $x \in ((< (\alpha, \beta) \sqcap = (\alpha + 1, \gamma)) \Rightarrow (\exists S.((= \gamma\ L) \sqcap (= \beta\ R) \sqcap \sqcup_{j \in H(D_i)} D_j))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$.
- if $x \in (=(\beta+1, \gamma))[\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$, then $b+1 = g$, and from the definition of $\mathcal{I}$ follows that $x \in (\exists S.((= \alpha\ L) \sqcap (= \gamma\ R) \sqcap D_j)[\frac{a}{\alpha}][\frac{g}{\gamma}])^{\mathcal{I}}$ for some D_j with $(D_i, D_j) \in V$, hence $x \in ((=(\beta + 1, \gamma)) \Rightarrow (\exists S.((= \alpha\ L) \sqcap (= \gamma\ R) \sqcap \sqcup_{j \in V(D_i)} D_j))[\frac{a}{\alpha}][\frac{b}{\beta}][\frac{g}{\gamma}])^{\mathcal{I}}$.

Hence $x \in {C_{\mathcal{D}}}^{\mathcal{I}}$.

■

The central result of this section can now be given.

Theorem 9 Satisfiability and subsumption of $\mathcal{ALCN}^S$-concepts are undecidable.

Proof: Undecidability of satisfiability follows immediately from

- the fact that for a given tiling system $\mathcal{D}$ it is undecidable whether there exists a tiling of the second octant of the plane using $\mathcal{D}$,
- Lemma 8 and

- the fact that–given a tiling system $\mathcal{D}$–the concept $C_\mathcal{D}$ can effectively be constructed.

Subsumption is undecidable since C is unsatisfiable iff $C \sqsubseteq (A \sqcap \neg A)$.

■

Investigating the form of the above defined concept $C_\mathcal{D}$ more closely, it turns out that all numerical variables in $C_\mathcal{D}$ are quantified universally. Using some obvious abbreviations for subconepts of $C_\mathcal{D}$ in which no quantification occurs, this can easily been seen by noting that

$$\begin{aligned} C_\mathcal{D} &= G \sqcap (\uparrow \alpha. \uparrow \beta. \ (C_{1,2,3} \sqcap \sqcap_{1\leq i\leq m} \ (E_i \Rightarrow (F_i \sqcap (\uparrow \gamma.H_i) \sqcap (\uparrow \gamma.V_i))))) \\ &= G \sqcap (\uparrow \alpha. \uparrow \beta. \ (C_{1,2,3} \sqcap \sqcap_{1\leq i\leq m} \ (\neg E_i \sqcup (F_i \sqcap (\uparrow \gamma.H_i) \sqcap (\uparrow \gamma.V_i))))). \end{aligned}$$

Hence one well-known source of high complexity, namely alternation of existential and universal quantification, is not present in $C_\mathcal{D}$. Informally, the interaction of universally quantified numerical variables and disjunction is the source of the undecidability.

In order to make the undecidability result more precise, we will define a sublanguage $\mathcal{ALUEN}^\uparrow$ of $\mathcal{ALCN}^S$ where all numerical variables are quantified universally and where satisfiability and subsumption are undecidable, too. This is done by restricting negation to concept names and allowing for universal quantification only:

Definition 10 Let N_C be a set of *concept names*, N_R a set of *role names* and N_V a set of variables. The *concepts* of $\mathcal{ALUEN}^\uparrow$ are defined as follows:

1. Every concept name is a concept.
2. If C and D are concepts, $A \in N_C$, R is a role name, α is a variable and $n \in \mathbb{N}$ a nonnegative integer, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg A)$, $(\forall R.C)$, $(\exists R.C)$, $(\uparrow \alpha.C)$, $(\mathrm{rel}\, \alpha\ R)$, $(\mathrm{rel}\, n\ R)$ are concepts.

Corollary 11 Satisfiability and subsumption of $\mathcal{ALUEN}^\uparrow$-concepts are undecidable.

Proof: Undecidability of satisfiability of $\mathcal{ALUEN}^\uparrow$-concepts follows directly from the fact that for a tiling system $\mathcal{D}$ the concept $C_\mathcal{D}$ is in $\mathcal{ALUEN}^\uparrow$ and Lemma 8. Again, subsumption is undecidable since C is unsatisfiable iff $C \sqsubseteq (A \sqcap \neg A)$.

3.3 $\mathcal{ALUEN}^{\downarrow}$, a decidable sublanguage of $\mathcal{ALCN}^S$

Informally, one reason for the undecidability of satisfiability of $\mathcal{ALCN}^S$-concepts is the interaction of universal quantification of numerical variables and disjunction. In this section it will be shown that satisfiability of $\mathcal{ALCN}^S$-concepts is decidable when negation is restricted to concept names. The only consequence of this restriction to primitive negation is that universal quantification of variables can no longer be expressed, whereas $\mathcal{ALCN}$ remains a sublanguage of this restricted language.

Definition 12 Let N_C be a set of *concept names*, N_R a set of *role names* and N_V a set of variables. The *concepts* of $\mathcal{ALUEN}^{\downarrow}$ are defined as follows:

1. Every concept name is a concept.
2. If C and D are concepts, $A \in N_C$, R is a role name, α is a variable and $n \in \mathbb{N}$ a nonnegative integer, then $(C \sqcap D)$, $(C \sqcup D)$, $(\neg A)$, $(\forall R.C)$, $(\exists R.C)$, $(\downarrow \alpha.C)$, $(\mathrm{rel}\, \alpha\ R)$, $(\mathrm{rel}\, n\ R)$ are concepts.

Semantic of $\mathcal{ALUEN}^{\downarrow}$-concepts is the natural restriction of the semantic of $\mathcal{ALCN}^S$-concepts. In order to simplify the investigation of $\mathcal{ALUEN}^{\downarrow}$-concepts, in what follows we will restrict our attention to concepts where each variable occurs either bound or free, and where each variable is bound atmost once by $\downarrow$. It is easy to see that each $\mathcal{ALUEN}^{\downarrow}$-concept C can be transformed to an equivalent concept C' of this form by renaming of variables.

Decidability of satisfiability of $\mathcal{ALUEN}^{\downarrow}$-concepts will be shown by presenting a tableau based algorithm and showing that this algorithm is sound, complete and terminating. The basic structure this algorithm works on are constraints:

Definition 13 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 $\mathcal{ALUEN}^{\downarrow}$-concept D and some $x \in \tau$ where $\mathrm{free}(D) \subseteq N_V \times \tau$.

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

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

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

where $\nu(D)$ is obtained from D by replacing each free occurrence of a 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 $x \in \tau$ holds $\{x : A, x : \neg A\} \subseteq S$. 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 nonnegative integers.

A constraint system S is called *complete* iff S is clash-free, numerically consistent and none of the completion rules given in figure 1 can be applied to S.

The following algorithm decides whether an $\mathcal{ALUEN}^{\downarrow}$-concept is satisfiable:

Figure 1 shows the *completion rules* which are used to test the satisfiability for a closed concept C by constructing constraint systems. The *completion algorithm* works on a tree where each node is labelled with a constraint system. It starts with a tree consisting of a root labelled with $S = \{x : C\}$ for some closed concept C. A rule can only be applied to a leaf labelled with a clash–free constraint system. Applying a rule $S \to S_i$ for $1 \leq i \leq n$ to such a leaf leads to the creation of n new successors of this node labelled respectively with constraint system S_i. The algorithm terminates if none of the rules can be applied to any of the leafs. The algorithm answers with "C is satisfiable" iff it created a complete constraint system starting with $\{x : C\}$.

Please note that each of the completion rules adds constraints when applied to a constraint system, that none of the rules removes constraints, and that variables $x \in \tau$ are never identified nor substituted.

The algorithm presented here might seem inefficient because it explicitely tests for each number of role successors n between 1 and m in rule 4. This cannot be avoided because n cannot be deduced at this stage of the algorithm: it might be influenced by number restrictions occuring in constraints on individual variables generated later. For example, unsatisfiability of the following concept can only be proved by testing the cases where x has one and where x has two R-successors:

$$x :(\downarrow \alpha.((= \alpha\ R) \sqcap (\exists R.A) \sqcap (\exists R.\neg A) \sqcap (\forall R.((= \alpha\ S) \sqcap (\leq 1\ S))))).$$

Furthermore, for a fixed number n of role successors of x, one has to test for all possibilities of satisfying constraints of the form $x :(\exists R.D)$. In similar algorithms this is usually done by generating only one role successor at a time and eventually identifying role successors of a variable. The advantage of making this explicit is that our algorithm does not need any explicit inequality constraints like $x \neq y$ nor does it need a rule which identifies variables.

In the following, it will be shown that the algorithm always terminates and that it is sound and complete. More formally, decidability of satisfiability of $\mathcal{ALUEN}^{\downarrow}$-concepts is a consequence of the following four lemmata.

Lemma 14 [Termination] For each closed concept C_0, the completion algorithm terminates.

Lemma 15 [Local invariance] Let C_0 be a closed $\mathcal{ALUEN}^{\downarrow}$-concept and let S be obtained by applying the completion rules to $\{x_0 : C_0\}$. Then for each completion rule $\mathcal{R}$ which can be applied to S and for each interpretation $\mathcal{I}$ holds:

$$\mathcal{I} \text{ is a model of } S \text{ iff } \mathcal{R} \text{ yields some } S_i \text{ satisfied by } \mathcal{I}.$$

[2]Here, two n-partitions $P.P'$ are said to be equal if for each i with $1 \leq i \leq n$ there exists some j with $P_i = P'_j$

1. Intersection: If $x:(C_1 \sqcap C_2) \in S$ and $x:C_1 \notin S$ or $x:C_2 \notin S$
$S \rightarrow_{\sqcap} S \cup \{x:C_1, x:C_2\}$
2. Union: If $x:(C_1 \sqcup C_2) \in S$ and $x:C_1 \notin S$ and $x:C_2 \notin S$
$S \rightarrow_{\sqcup} S_1 = S \cup \{x:C_1\}$
$S \rightarrow_{\sqcup} S_2 = S \cup \{x:C_2\}$
3. Numerical Existential Quantification:
If $x:(\downarrow \alpha.D) \in S$ and $x:D[\frac{\alpha_x}{\alpha}] \notin S$
$S \rightarrow_{\downarrow} S \cup \{x:D[\frac{\alpha_x}{\alpha}]\}$
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[2] $P = \biguplus_{1 \leq i \leq n} P_i = \{1, \ldots, m\}$ of m let S_P be defined as follows:
$S \rightarrow_R S_P = \ S \cup \{xRy_i, \mid 1 \leq i \leq n\} \cup \{y_i : E_j \mid 1 \leq i \leq n, j \in P_i\} \cup$
$\{y_i : D_j \mid 1 \leq i \leq n, 1 \leq j \leq k\} \cup \{x:(\geq n\ R)\}$
where $y_i \in \tau$ are new variables.
5. Prophylactic new objects
If $xRy \notin S$ for all $y \in \tau$ and $x:(=\ 0\ \ R) \notin S$ and k maximal with $x:(\forall R.D_i) \in S$ for $1 \leq i \leq k$, $x:(\mathrm{rel}\, N\ R) \in S$ 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:
$S \rightarrow_n S_1 = S \cup \{x:(= 0\ R)\}$
$S \rightarrow_n S_2 = S \cup \{xRy\} \cup \{y:D_i \mid 1 \leq i \leq k\} \cup \{x:(> 0\ R)\}$
where $y \in \tau$ is a new variables.

Figure 1: The completion rules for $\mathcal{ALUEN}^{\downarrow}$

Lemma 16 [Model existence] If S is a complete constraint system obtained by applying the completion rules to $\{x_0 : C_0\}$ for some closed concept C_0, then there exists an interpretation $\mathcal{I}$ satisfying S.

Lemma 17 If S is a constraint system that contains either a clash or is not numerically consistent, then there exists no interpretation $\mathcal{I}$ satisfying S.

Proof of Lemma 14 [**Termination**]**:** According to the preconditions of the rules, a rule can be applied atmost once to each constraint in S: If one of the completion rules has been applied to some $x : D$, then S is modified in such a way that this rule is no longer applicable to any successor constraint system of S and $x : D$. For example the application of rule 5 to some S and x, R as described in the precondition of rule 6 leads to S_1, S_2 where $x :(= 0\ R) \in S_1$ and $xRy \in S_2$, hence rule 6 can no more be applied to S_1 nor to S_2 nor to any of their successor constraint systems with respect to x, R.

If a rule adds a new constraint $x : D$, then D is either a strictly shorter subterm (with eventually renamed variables) of one of those who caused the application of this rule or D is a number restriction to which no more rules can be applied. A concept C_0 has only finitely many subterms. Each rule adds only a finite set of new constraints to each of its finitely many successors, constraints are never removed from constraint systems and variables are never identified, hence the completion algorithm terminates. ∎

Proof of Lemma 15 [**Local invariance**]**:** Let $\mathcal{I}$ be a model of S with $\pi : \tau_S \to \Delta^{\mathcal{I}}$ and $\nu : N_V \times \Delta^{\mathcal{I}} \to \mathbb{N}$.

Intersection: If S' is obtained from S by application of rule 1, then $S' = S \cup \{x : C_1, x : C_2\}$ where $x :(C_1 \sqcap C_2) \in S$. $\mathcal{I}$ satisfies $x :(C_1 \sqcap C_2)$ iff $\pi(x) \in (C_1 \sqcap C_2)^{\mathcal{I}}$ iff $\mathcal{I}$ satisfies $x : C_1$ and $x : C_2$.

Union: If rule 2 can be applied to S, then $x :(C_1 \sqcup C_2) \in S$. $\mathcal{I}$ satisfies S iff $\mathcal{I}$ satisfies S and $\pi(x) \in {C_1}^{\mathcal{I}}$ or $\pi(x) \in {C_2}^{\mathcal{I}}$ iff $\mathcal{I}$ satisfies $S_1 = S \cup \{x : C_1\}$ or $S_2 = S \cup \{x : C_2\}$.

Numerical Existential Quantification: Let $\mathcal{I}, \pi, \nu$ satisfy S. $\mathcal{I}$ satisfies S with $x :(\downarrow \alpha.C) \in S$ iff $\mathcal{I}$ satisfies S and $x \in (C[\frac{n}{\alpha}])^{\mathcal{I}}$ for some $n \in \mathbb{N}$ iff $\mathcal{I}$ satisfies S and $x \in ((C[\frac{\alpha_x}{\alpha}])[\frac{\nu'(\alpha,x)}{\alpha_x}])^{\mathcal{I}}$ for

$$\nu'(\beta, y) = \begin{cases} n & \text{if } \beta = \alpha \text{ and } x = y \\ \nu(\beta, y) & \text{otherwise} \end{cases}$$

iff $\mathcal{I}$ satisfies $S' = S \cup \{x : C[\frac{\alpha_x}{\alpha}]\}$.

New Objects: Let x, R, k, m be as specified in the precondition of rule 5. $\mathcal{I}$ satisfies S iff for some $z_1, \ldots, z_\ell \in \Delta^{\mathcal{I}}$ holds

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

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

Because of the maximality of m, k and since rule 1-4 cannot be applied to S, the above mentioned constraints on R-successors of x are the only ones implied by S. There are two cases to distinguish: If $\ell \leq m$ then let P be the ℓ-partition of m with $j \in P_{j'}$ iff $z_{j'} \in E_j$. If $\ell > m$ then let P be the m-Partition of m with $j \in P_{j'}$ iff $z_{j'} \in E_j$ for $1 \leq j' \leq \ell$. Let $\pi(y_i) = z_i$. Then $\mathcal{I}$ satisfies S iff $\mathcal{I}$ satisfies S_P since

- $(\pi(x), \pi(y_i)) \in R^{\mathcal{I}}$ for all $1 \leq i \leq \min\{m, \ell\}$,
- for each $x : (\exists R.E_j) \in S$ holds $(\pi(x), \pi(y_{j'})) \in R^{\mathcal{I}}$ and $\pi(y_{j'}) \in E_j^{\mathcal{I}}$ for $j \in P_{j'}$,
- $\pi(y_i) \in D_j^{\mathcal{I}}$ for all $1 \leq i \leq \min\{m, \ell\}$, $1 \leq j \leq k$,
- $x_R^{\mathcal{I}} \geq \ell$.

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

■

Proof of Lemma 16 [**Model existence**]**:** Given a complete constraint system S obtained by applying the completion rules to $\{x_0 : C_0\}$ for some closed concept C_0, we will construct a model of S.

In the following, we will need *copies* of an interpretation: $\mathcal{I}'$ is a copy of an interpretation $\mathcal{I}$ iff there is a bijection $\phi : \Delta^{\mathcal{I}} \to \Delta^{\mathcal{I}'}$, for all concepts D holds $x \in D^{\mathcal{I}}$ iff $\phi(x) \in D^{\mathcal{I}'}$ and $(x, y) \in R^{\mathcal{I}}$ iff $(\phi(x), \phi(y)) \in R^{\mathcal{I}'}$. We say that $\phi(x)$ is a copy of x.

The construction starts by introducing an interpretation $\mathcal{I}_x$ containing a element x for each element in $\mathcal{I}$ whose properties can directly be deduced from S. Then, if all interpretations $\mathcal{I}_y$ for all successors y of z have been defined, the interpretation $\mathcal{I}_z$ is constructed as the union of all these $\mathcal{I}_y$ and possibly copies of them (if number restrictions imply more successors than actually present). More formally:

For all $\alpha \in N_V, x \in \tau_S$, for R occuring in S' let $\hat{x}_R, \hat{\alpha}_y \in \mathbb{N}$ be such that

$$\bigwedge_{\substack{x:(\mathrm{rel}\, n\ R) \in S \\ x \in \tau_{S'}, R \in N_R, n \in \mathbb{N}}} (\hat{x}_R \,\mathrm{rel}\, n) \wedge \bigwedge_{\substack{x:(\mathrm{rel}\, \alpha_y\ R) \in S \\ x, y \in \tau_{S'}, R \in N_R, \alpha \in N_V}} (\hat{x}_R \,\mathrm{rel}\, \hat{\alpha}_y)$$

is valid in $(\mathbb{N}, <)$. Since S is numerically consistent, these $\hat{x}_R, \hat{\alpha}_y \in \mathbb{N}$ exist.

The canonical model $\mathcal{I} = \mathcal{I}_{x_0}$ is inductively defined as follows:

For all $\alpha \in N_V, x \in \tau_S$, for all R occuring in S let $\nu(\alpha, x) = \hat{\alpha}_x$.

For all $x \in \tau_S$ with $xRy \notin S$ for all R occuring in C, define $\mathcal{I}_x$ by

$$\Delta^{\mathcal{I}_x} = \{x\} \cup \bigcup_{1 \le i \le \hat{x}_R} \{x_{i,R}\},$$

and for $y \in \Delta^{\mathcal{I}_x}$, for a concept name A define $y \in A^{\mathcal{I}_x}$ iff $y\!:\!A \in S$. Finally, define $(y, z) \in R^{\mathcal{I}_x}$ iff $x = y$ and $z = x_{i,R}$ for some $i \in \mathbb{N}$.

Now, for each $x \in \tau_S$ where $\mathcal{I}_x$ is not yet defined and where $\mathcal{I}_y$ is already defined for all y with $xRy \in S$ for all R occuring in C, let

$$\Delta^{\mathcal{I}_x} = \{x\} \cup \bigcup_{\substack{1 \le i \le \hat{x}_R \\ \text{and where forall } y \in \tau_S \,:\, xRy \notin S}} \{x_{i,R}\} \cup \bigcup_{y \text{ with } xRy \in S} \Delta^{\mathcal{I}_y} \cup M$$

where M is a set of copies of some $\Delta^{\mathcal{I}_y}$: If $xRy \in S$ and $\hat{x}_R > \#\{y \mid xRy \in S\}$ then M consists of $\hat{x}_R - \#\{y \mid xRy \in S\}$ copies of $\Delta^{\mathcal{I}_y}$ for some y with $xRy \in S$.

For all $y, z \in \Delta^{\mathcal{I}_x}$, for all concept names A define $y \in A^{\mathcal{I}_x}$ iff $y\!:\!A \in S$ or $z\!:\!A \in S$ for some z where y is a copy of. Furthermore, let $(y, z) \in R^{\mathcal{I}_x}$ iff $yRz \in S$ or z is a copy of z' and $yRz' \in S$ or $(y, z) \in R^{\mathcal{I}_w}$ for some $\mathcal{I}_w$ already defined or $x = y$ and $z = x_{i,R}$ for some $i \in \mathbb{N}$..

It has to be shown that the canonical model $\mathcal{I} = \mathcal{I}_{x_0}$ is well-defined and that it really satisfies S.

The canonical model is well-defined: It can easily be seen that the construction terminates and that for all concept names A and all $x \in \Delta^{\mathcal{I}}$ either $x \in A^{\mathcal{I}}$ or $x \notin A^{\mathcal{I}}$ holds.

The canonical model $\mathcal{I}$ satisfies S: This will be shown by induction on the structure of concepts:

Let $\pi : \tau_S \to \Delta^{\mathcal{I}}$ be the identity mapping.

For each concept name A, for each $(x : A) \in S$ holds $x \in A^{\mathcal{I}}$ and for each $xRy \in S$ holds $(x, y) \in R^{\mathcal{I}}$. If $x : \neg A \in S$, then $x : A \notin S$ because S is clash-free, hence $x \notin A^{\mathcal{I}}$ and $\mathcal{I}$ satisfies $x : \neg A$. For each $x :(\text{rel}\, n\ R) \in S$ holds ${x_R}^{\mathcal{I}} = \hat{x}_R$, hence $\mathcal{I}$ satisfies all $x :(\text{rel}\, n\ R) \in S$. For each $x :(\text{rel}\, \alpha_y\ R) \in S$ holds ${x_R}^{\mathcal{I}} = \hat{x}_R$ and since $\nu(\alpha, y) = \hat{\alpha}_y$ we have that $\mathcal{I}$ satisfies all $x :(\text{rel}\, \alpha_y\ R) \in S$.

If $x : C_1 \sqcap C_2 \in S$, then $\{x : C_1, x : C_2\} \subseteq S$, hence $\mathcal{I}$ satisfies $x : C_1 \sqcap C_2$ by induction.

If $x : C_1 \sqcup C_2 \in S$, then $\{x : C_1\} \in S$ or $\{x : C_2\} \in S$, hence $\mathcal{I}$ satisfies $x : C_1 \sqcup C_2$ by induction.

If $x :(\exists R.E) \in S$, then there is some $y \in \tau_S$ such that $\{xRy, y : E\} \subseteq S$, hence $\mathcal{I}$ satisfies $x :(\exists R.E)$ by induction.

If $x :(\forall R.E) \in S$ and $(x, y) \in R^{\mathcal{I}}$, then either $\{xRy, y : E\} \subseteq S$ or y is a copy of some y' with $\{xRy', y' : E\} \subseteq S$. Hence by induction $x :(\forall R.E)$ is satisfied by $\mathcal{I}$.

If $x :(\downarrow \alpha.D) \in S$, then $x : D[\frac{\alpha_x}{\alpha}] \in S$ and by induction we have that $\mathcal{I}$ satisfies $x :(\downarrow \alpha.D)$. ∎

Proof of Lemma 17: Let S be a constraint system. It is clear that no interpretation satisfies both $x : A$ and $x : \neg A$, hence a constraint system containing a clash is unsatisfiable.

Now let S be not numerically consistent. Suppose there is an interpretation satisfying S, this is to say that there exists $\pi : \tau_S \to \Delta^{\mathcal{I}}$ and $\nu : N_V \times \tau_S \to \mathbb{N}$ such that $\mathcal{I}, \pi, \nu$ satisfy S. Especially $\mathcal{I}, \pi, \nu$ satisfy each number restriction contained in S, i.e.

$$\bigwedge_{\substack{x :(\text{rel}\, n\ R) \in S \\ x \in \tau_S,\ \alpha \in N_V, n \in \mathbb{N}}} ({x_R}^{\mathcal{I}}\ \text{rel}\, n) \wedge \bigwedge_{\substack{x :(\text{rel}\, \alpha_y\ R) \in S \\ x, y \in \tau_S,\ \alpha \in N_V}} ({x_R}^{\mathcal{I}}\ \text{rel}\, \nu(\alpha, y))$$

is valid in $(\mathbb{N}, <)$ in contradiction to S beeing not numerically consistent.

∎

Theorem 18 Satisfiability of $\mathcal{ALUEN}^{\downarrow}$-concepts is decidable.

Proof: Using Lemma 14, the completion algorithm terminates for each closed concept C_0. Together with Lemma 15 this implies that if $x : C_0$ has a model, then there is at least one constraint system generated by the completion algorithm to which no more rules can be applied which has this model. And vice versa, if none of the constraint systems generated by the completion algorithm has a model, then $x : C_0$ has no model. From lemma 16 follows that complete constraint systems are satisfiable, and with Lemma 17 we have that a non-complete constraint system to which no more rules can be applied is unsatisfiable.

Collecting these facts, an $\mathcal{ALUEN}^{\downarrow}$-concept C is satisfiable iff a closed equivalent concept $C_0 = (\downarrow \alpha_1 . \ldots \downarrow \alpha_n . C)$ is satisfiable iff the constraint system $\{x_0 : C_0\}$ is satisfiable iff the completion algorithm generates at least one complete constraint system. It is decidable whether a constraint system contains a clash and whether it is numerically consistent. This second point can easily be seen by noting that numerical consistency can be tested using a modified cycle detection algorithm running in time cubic to the sice of the formula. ∎

Please note that for $\mathcal{ALUEN}^{\downarrow}$, decidability of satisfiability does not imply decidability of subsumption:

Theorem 19 Subsumption of $\mathcal{ALUEN}^{\downarrow}$-concepts is undecidable.

Proof: In section 3.2, Corollary 11 states that satisfiability of $\mathcal{ALUEN}^{\uparrow}$-concepts is undecidable. Since $(\uparrow \alpha . C) = \neg(\downarrow \alpha . \neg C)$, we have that the negation $\neg C$ of an $\mathcal{ALUEN}^{\uparrow}$-concept C is an $\mathcal{ALUEN}^{\downarrow}$-concept. Furthermore, an $\mathcal{ALUEN}^{\uparrow}$-concept C is unsatisfiable

$$\begin{array}{lrcl} \text{iff} & C & = & (\neg A \sqcap A) \\ \text{iff} & \neg C & = & (A \sqcup \neg A) \\ \text{iff} & (A \sqcup \neg A) & \sqsubseteq & \neg C \end{array}$$

∎

4 Conclusion

We have investigated one of the weakest concept languages–$\mathcal{ALCN}^S$–containing some sort of symbolic number restrictions and beeing propositionally closed. Nonetheless, satisfiability and subsumption of $\mathcal{ALCN}^S$-concepts turned out to be undecidable. There are mainly two reaons why $\mathcal{ALCN}^S$ is called weak:

First, it does not even allow for operations on numerical variables such as addition or multiplication. But the high amount of expressive power added to $\mathcal{ALCN}$ can be seen by noting that addition and multiplication can almost be expressed: This is to say that there exists an $\mathcal{ALCN}^S$-concept where each of its instances has more T-successors than R- and S-successors together. And another concept exists where each of its instances has more T-successors than the number of R- times the number of S-successors.

Second, it does not allow for any role forming operators like composition or conjunction of roles.

The undecidability results given here make one think of alternative ways of defining some sort of symbolic number restrictions that may have less expressive power but are easier to be handled algorithmically.

An alternative approach of comparing numbers of role successors would be to avoid the introduction of variables and to compare numbers of role successors directly, like in $(= R\ S)$ where R and S are roles and where $x \in (= R\ S)^{\mathcal{I}}$ iff $x_R{}^{\mathcal{I}} = x_S{}^{\mathcal{I}}$. Unfortunately, these concepts are only able to express some sort of symmetry conditions like "more neighbors to the right than to the left" or "the same number of daughters as of sons". In order to compare not only numbers of role successors but also of role-*path* successors, one has to allow for composition of roles inside these comparing concepts. For example "families where the number of rooms in the house they are living in is bigger than the number of children" could then be expressed by

$$\texttt{family} \sqcap (= 1\ \texttt{home}) \sqcap (\geq\ \texttt{home} \circ \texttt{has\_room}\ \texttt{children}).$$

This yields a completely different extension than the one presented here in that the number of role successors in different levels of nested concepts cannot be compared directly. For example, the following concept cannot be expressed in this extension:

$$\texttt{device} \sqcap (\downarrow \alpha.(\forall \texttt{component}.(= \alpha\ \texttt{input}))).$$

Since it is not even known whether satisfiability of $\mathcal{ALCN}$ extended by additionally allowing compositions of roles inside number restrictions like $(= n\ R_1 \circ \ldots \circ R_m)$ is decidable, this question is left open here but will be part of future work.

References

[Baader and Hollunder, 1991] F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithm. In *Proc. of the Workshop on Processing Declarative Knowledge, PDK-91*, volume 567 of *LNAI*, pages 67–86. Springer-Verlag, 1991.

[Baader *et al.*, 1994] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H. 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.

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

[Donini *et al.*, 1991a] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In *Proc. of KR-91*, Boston (USA), 1991.

[Donini *et al.*, 1991b] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Tractable concept languages. In *Proc. of IJCAI-91*, pages 458–463, Sydney, 1991.

[Donini *et al.*, 1995] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. Technical Report RR-95-07, DFKI, Kaiserslautern, Deutschland, 1995.

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

[Levesque and Brachman, 1987] H. Levesque and R. J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[Marquardt, 1994] W. Marquardt. Trends in computer-aided process modeling. In *Proc. of ICPSE '94*, pages 1–24, Kyongju, Korea, 1994.

[Wang, 1963] H. Wang. Dominoes and the AEA case of the Decision Problem. *Bell Syst. Tech. J.*, 40:1–41, 1963.