A tableau algorithm for reasoning about concepts and similarity

Carsten Lutz[1], Frank Wolter[2], and Michael Zakharyaschev[3]

[1] Institut für Theoretische Informatik, TU Dresden,
Fakultät Informatik, 01062 Dresden, Germany;
lutz@tcs.inf.tu-dresden.de
[2] Department of Computer Science, University of Liverpool
Liverpool L69 7ZF, U.K.
frank@csc.liv.ac.uk
[3] Department of Computer Science, King's College London,
Strand, London WC2R 2LS, U.K.
mz@dcs.kcl.ac.uk

Abstract. We present a tableau-based decision procedure for the fusion (independent join) of the expressive description logic $\mathcal{ALCQO}$ and the logic $\mathcal{MS}$ for reasoning about distances and similarities. The resulting 'hybrid' logic allows both precise and approximate representation of and reasoning about concepts. The tableau algorithm combines the existing tableaux for the components and shows that the tableau technique can be fruitfully applied to fusions of logics with nominals—the case in which no general decidability transfer results for fusions are available.

1 Introduction

Undoubtedly, there will come a day when, to attract submissions, organisers will be trying to annotate their conference sites with machine readable information. Imagine, for instance, that we want to do this now for *Tableaux 2003*. Choosing a formalism for representation of and reasoning about the terminology used in the Tableaux 2003 site, we may naturally try the description logic $\mathcal{ALCQO}$ underlying the DAML+OIL language of the semantic web [7, 1]. Then we start with a definition of tableau-style algorithms and, as a first attempt, write something like this:

$$\mathsf{Tableau\_style\_algorithm} = \mathsf{Algorithm} \sqcap \exists \mathsf{comprises.Rule}, \tag{$\dagger$}$$

saying that tableau-style algorithms are precisely those algorithms that are equipped with rules. Well, it seems unlikely that any potential participant of *Tableaux 2003* would be happy with this provocative definition (according to which almost all reasoning procedures may be called tableau-based). Then how to improve it? Do we really have a good, clear and concise definition (which is better than 'lots of rules, but few axioms')? How can we represent in $\mathcal{ALCQO}$ many other 'vague' concepts from the site, such as 'related techniques,' 'related methods,' 'new calculi,' etc.?

One of the possible solutions to these problems is to introduce a *similarity measure* between the objects of the application domain—in our case the reasoning procedures (which can be based on common sense, or defined by an expert, or automatically generated using certain algorithms). Then, by taking a role name $\mathsf{similar\_to\_degree} \leq 1$, we could say, for instance, that tableau-style algorithms are similar to degree ≤ 1 to at least one of the prototypical tableau algorithms $\mathsf{ta}_1, \ldots, \mathsf{ta}_7$. However, this approach is in conflict with the expressive capabilities of standard description logics (DLs) such as $\mathcal{ALC}$ or DAML+OIL because usually similarity measures are supposed to satisfy a number of natural axioms like the axioms of *metric spaces*, in particular, a sort of 'triangular inequality' which is *not expressible in standard DLs.*

The main idea of this paper is not to extend the family of DLs by introducing a new one, but rather to *combine* the existing knowledge representation formalisms, viz.,

- the standard description logic $\mathcal{ALCQO}$—i.e., the basic DL $\mathcal{ALC}$ extended with qualified number restrictions, nominals and general TBoxes [5], and
- the logic $\mathcal{MS}$ [13] for reasoning about metric spaces[4]

in order to achieve the desirable expressivity.

To illustrate the expressive power of the resulting 'hybrid' logic *sim*-$\mathcal{ALCQO}$, we show how one can further 'approximate' the definition of tableau-style algorithms. First, we add to the right-hand side of (†) the conjunct

$$\mathsf{E}^{\leq 1}(\mathsf{ta}_1 \sqcup \cdots \sqcup \mathsf{ta}_7)$$

which is an $\mathcal{MS}$-formula saying that tableau-style algorithms should be similar to degree ≤ 1 to at least one of $\mathsf{ta}_1, \ldots, \mathsf{ta}_7$. If this 'positive information' is still not enough, one can add some 'negative' bit. For example, it may be natural to say that tableau-style algorithms are neither similar to degree ≤ 0.5 to a certain Hilbert-style algorithm ha, nor similar to degree ≤ 0.5 to any resolution-based decision procedure:

$$\neg\mathsf{E}^{\leq 0.5}\mathsf{ha} \sqcap \neg\mathsf{E}^{\leq 0.5}\mathsf{Resolution\_based\_algorithm}.$$

Of course, the individual algorithms such as ha can also be described by means of concepts, possibly involving similarity measures:

$$\mathsf{ha} : \mathsf{Algorithm} \sqcap \neg\exists\mathsf{feature.Termination} \sqcap \mathsf{A}^{\leq 0.5}(\exists\mathsf{comprises.Modus\_ponens})$$

(i.e., ha does not necessarily terminate and all ≤ 0.5 similar algorithms use a kind of *modus ponens* as one of their inference rules). It may seem more natural to specify similarity in terms of a finite set of symbolic similarity measures such as 'close' and 'far' rather than in terms of rational numbers as above. In

[4] This metric logic differs considerably from the metric logics investigated in [9]. Here we quantify over open and closed 'balls,' while in [9] over closed balls and their complements. The expressive power of the two languages is, therefore, incomparable.

our approach, however, the user is free to choose either option: one may fix a rational number for each symbolic similarity measure, say, 1 for 'close' and 10 for 'far' (or the other way round), and then work with the symbolic names.

In this paper, we provide a tableau-style decision procedure for the new logic *sim*-$\mathcal{ALCQO}$. Technically, this logic is the *fusion* (or *independent join*) [8, 3] of $\mathcal{ALCQO}$ and $\mathcal{MS}$. We believe that this is a reasonable starting point, since many similarity measures are indeed metric, and our approach without any problems can be adapted to similarity measures which do *not* satisfy all of the axioms of metric spaces. Moreover, we can easily extend *sim*-$\mathcal{ALCQO}$ and the tableau algorithm with additional similarity measures (say, between inference rules).

In our opinion, *sim*-$\mathcal{ALCQO}$ provides just the right compromise between expressive power and computational cost:

(1) In *sim*-$\mathcal{ALCQO}$, we can mix constructors of $\mathcal{ALCQO}$ and $\mathcal{MS}$ in order to define concepts based on similarity measures as illustrated above. Moreover, as our tableau algorithm shows, reasoning in *sim*-$\mathcal{ALCQO}$ is decidable. It is of interest to contrast this with the fact that a tighter coupling of $\mathcal{ALCQO}$ and $\mathcal{MS}$ leads to undecidability: as we also show, the extension of $\mathcal{MS}$ with qualified number restrictions such as 'there exists at most 1 point x with property P within distance ≤ 1' results in an undecidable logic. Therefore, the fusion of the two formalisms seems to be a good starting point for investigating the interaction between concepts and similarity measures.

(2) Although there exists a number of general results regarding the transfer of decidability from the components of a fusion to the fusion itself [8, 3, 12, 2, 11], these results do not apply to logics with nominals (atomic concepts interpreted as singleton sets) such as $\mathcal{ALCQO}$. In fact, no transfer result is available from which we could derive the decidability of *sim*-$\mathcal{ALCQO}$ using the decidability of both $\mathcal{ALCQO}$ and $\mathcal{MS}$. Despite the fact that they are not applicable, it is of interest to note that our algorithm has an important advantage over general approaches to proving decidability: structurally, it is very similar to the tableau algorithms for $\mathcal{SHIQ}$ and $\mathcal{SHOQ}$ proposed in [6, 5]. Since these algorithms have turned out to be implementable in efficient reasoning systems, we do hope that our algorithm shares this attractive property as well.

The paper is organised as follows: in Section 2, we introduce the description logic *sim*-$\mathcal{ALCQO}$. In Section 3, we describe the tableau algorithm for deciding the satisfiability of *sim*-$\mathcal{ALCQO}$-knowledge bases, whose correctness is then proved in Section 4. Section 5 is concerned with the undecidability of $\mathcal{MS}$ extended with qualifying number restrictions. A version of this paper with detailed proofs is available at `http://www.csc.liv.ac.uk/~frank`.

2 The logic *sim*-$\mathcal{ALCQO}$

In this section, we introduce the combined logic *sim*-$\mathcal{ALCQO}$. The *alphabet* for forming concepts and assertions consists of the following elements:

– a countably infinite list of *concept names* $A_1, A_2, \ldots$;

- a countably infinite list of *object names* $\ell_1, \ell_2, \ldots$;
- binary *distance* ($\boldsymbol{\delta}$), *equality* ($=$) and *membership* ($:$) *predicates*;
- the *Boolean operators* $\sqcap$, $\sqcup$, $\neg$;
- two *distance quantifiers* $\mathsf{E}^{<a}$, $\mathsf{E}^{\leq a}$ and their duals $\mathsf{A}^{<a}$, $\mathsf{A}^{\leq a}$, for every positive rational number a (i.e., $a \in \mathbb{Q}^+$);
- *role names* $R_1, R_2, \ldots$;
- *qualified number restrictions* $(\leq nR.C)$ and $(\geq nR.C)$, for every natural n, every role name R, and every concept C.

Using this alphabet, sim-$\mathcal{ALCQO}$*-concepts* are defined by the formation rule:

$$C ::= A_i \mid \ell_i \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \mathsf{E}^{<a}C \mid \mathsf{E}^{\leq a}C \mid \mathsf{A}^{<a}C \mid \\ \mid \mathsf{A}^{\leq a}C \mid (\leq nR_i.C) \mid (\geq nR_i.C).$$

As usual, we write $\top$ as an abbreviation for an arbitrary propositional tautology, $\bot$ for $\neg\top$, $\exists R.C$ for $(\geq 1R.C)$, and $\forall R.C$ for $(\leq 0R.\neg C)$. At first sight, it may seem strange to have both strict and non-strict versions of the E and A constructors available for talking about similarity measures. Note, however, that this allows us to define the concept $\mathsf{E}^{\leq a}C \sqcap \neg\mathsf{E}^{<a}C$ which states that the most similar object from C is located precisely at distance a. Object names occurring in concepts will also be called *nominals*.

Now we define *sim-*$\mathcal{ALCQO}$*-assertions* as expressions of the following forms:

- $\ell : C$, where ℓ is an object name and C a concept;
- $C_1 = C_2$, where C_1 and C_2 are concepts;
- $\boldsymbol{\delta}(k, \ell) < a$, $\boldsymbol{\delta}(k, \ell) \leq a$, $\boldsymbol{\delta}(k, \ell) > a$, $\boldsymbol{\delta}(k, \ell) \geq a$, where k, ℓ are object names and $a \in \mathbb{Q}^+$.

Assertions of the third form are called *distance assertions*. A *sim-*$\mathcal{ALCQO}$*-knowledge base* is a finite set of *sim-*$\mathcal{ALCQO}$-assertions.

Observe that knowledge bases subsume both general TBoxes and ABoxes. In particular, the rather common ABox assertions of the form $(\ell_1, \ell_2) : R$, where ℓ_1 and ℓ_2 are object names and R a role name, can be viewed as abbreviations for $\ell_1 : \exists R.\ell_2$.

The semantics of *sim-*$\mathcal{ALCQO}$-concepts is a blend of the semantics of the logic of metric spaces [13] and the usual set-theoretic semantics of description logics. A *concept-distance model* (a *CD-model*, for short) is a structure of the form

$$\mathfrak{B} = \left\langle W, d, A_1^{\mathfrak{B}}, A_2^{\mathfrak{B}}, \ldots, R_1^{\mathfrak{B}}, R_2^{\mathfrak{B}}, \ldots, \ell_1^{\mathfrak{B}}, \ell_2^{\mathfrak{B}} \ldots \right\rangle,$$

where $\langle W, d\rangle$ is a *metric space* with a *distance function* d satisfying, for all $x, y, z \in W$, the axioms

$$d(x, y) = 0 \ \text{ iff } \ x = y, \tag{1}$$

$$d(x, z) \leq d(x, y) + d(y, z), \tag{2}$$

$$d(x, y) = d(y, x), \tag{3}$$

the $A_i^{\mathfrak{B}}$ are subsets of W, the $R_i^{\mathfrak{B}}$ are binary relations on W, and the $\ell_i^{\mathfrak{B}}$ are singleton subsets of W such that $i \neq j$ implies $\ell_i^{\mathfrak{B}} \neq \ell_j^{\mathfrak{B}}$.

The extension $C^{\mathfrak{B}}$ of a *sim*-$\mathcal{ALCQO}$-concept C is computed inductively:

$$(C_1 \sqcap C_2)^{\mathfrak{B}} = C_1^{\mathfrak{B}} \cap C_2^{\mathfrak{B}}, \quad (C_1 \sqcup C_2)^{\mathfrak{B}} = C_1^{\mathfrak{B}} \cup C_2^{\mathfrak{B}}, \quad (\neg C)^{\mathfrak{B}} = W - C^{\mathfrak{B}},$$

$$\begin{aligned}
(\mathsf{E}^{\leq a} C)^{\mathfrak{B}} &= \{x \in W \mid \exists y \in W \left(d(x,y) \leq a \ \wedge y \in C^{\mathfrak{B}}\right)\},\\
(\mathsf{E}^{< a} C)^{\mathfrak{B}} &= \{x \in W \mid \exists y \in W \left(d(x,y) < a \ \wedge y \in C^{\mathfrak{B}}\right)\},\\
(\mathsf{A}^{\leq a} C)^{\mathfrak{B}} &= \{x \in W \mid \forall y \in W \left(d(x,y) \leq a \ \rightarrow y \in C^{\mathfrak{B}}\right)\},\\
(\mathsf{A}^{< a} C)^{\mathfrak{B}} &= \{x \in W \mid \forall y \in W \left(d(x,y) < a \ \rightarrow y \in C^{\mathfrak{B}}\right)\},\\
(\leq nR.C)^{\mathfrak{B}} &= \{x \in W \mid \left|\{y \in W \mid (x,y) \in R^{\mathfrak{B}} \wedge y \in C^{\mathfrak{B}}\}\right| \leq n\},\\
(\geq nR.C)^{\mathfrak{B}} &= \{x \in W \mid \left|\{y \in W \mid (x,y) \in R^{\mathfrak{B}} \wedge y \in C^{\mathfrak{B}}\}\right| \geq n\}.
\end{aligned}$$

We still have to specify when a CD-model satisfies a *sim*-$\mathcal{ALCQO}$-assertion: the *truth-relation* $\models$ between CD-models $\mathfrak{B}$ and assertions φ is defined as follows:

- $\mathfrak{B} \models \ell : C$ iff $\ell^{\mathfrak{B}} \subseteq C^{\mathfrak{B}}$,
- $\mathfrak{B} \models C_1 \doteq C_2$ iff $C_1^{\mathfrak{B}} = C_2^{\mathfrak{B}}$,
- $\mathfrak{B} \models \boldsymbol{\delta}(k,\ell) \leq a$ iff $d(k^{\mathfrak{B}}, \ell^{\mathfrak{B}}) \leq a$,
- $\mathfrak{B} \models \boldsymbol{\delta}(k,\ell) < a$ iff $d(k^{\mathfrak{B}}, \ell^{\mathfrak{B}}) < a$, and similar for $\geq$ and $>$.

Finally, a *sim*-$\mathcal{ALCQO}$-knowledge base Σ is called *satisfiable* if there exists a CD-model $\mathfrak{B}$ such that $\mathfrak{B} \models \varphi$ for all $\varphi \in \Sigma$. In this case we write $\mathfrak{B} \models \Sigma$.

Note that we make the *unique name assumption* (*UNA*), i.e., different object names denote distinct domain elements. The sole purpose of this assumption is to allow a clearer presentation of our tableau algorithm. It is, however, easily seen that the UNA has no influence on decidability, and that our tableau algorithm can be extended to deal with *sim*-$\mathcal{ALCQO}$ without UNA.

3 The tableau algorithm

Now we present a sound, complete and terminating algorithm for checking the satisfiability of *sim*-$\mathcal{ALCQO}$-knowledge bases. In fact, it is a (labelled) tableau algorithm that generalises the existing tableau algorithms for metric logics [13] and for the description logic $\mathcal{ALCQO}$ [5]. Before formulating the algorithm and proving its correctness, we introduce some notations and auxiliary definitions.

Supose we are given a *sim*-$\mathcal{ALCQO}$-knowledge base Σ. Denote by $con(\Sigma)$ the set of concepts occurring in Σ (including all subconcepts), by $rol(\Sigma)$ the set of role names occurring in Σ, by $par(\Sigma)$ the set of rational numbers occurring in Σ (either in E/A concepts or in distance assertions), and by $ob(\Sigma)$ we denote the set of object names occurring in Σ. Without loss of generality, we may assume that neither $par(\Sigma)$ nor $ob(\Sigma)$ are empty: if this is not the case, we can always add an assertion $\ell : \mathsf{A}^{<a}\top$ with a fresh object name ℓ. To simplify presentation, it is convenient to make three assumptions:

(1) A concept C is in *negation normal form* (*NNF*) if negation occurs only in front of concept names and nominals. Each concept can be transformed into an equivalent one in NNF by pushing negation inwards: for example, $\neg\mathsf{E}^{<a}C$ is equivalent to $\mathsf{A}^{<a}\neg C$. So, without loss of generality, we may assume that all concepts are in NNF. In what follows, we use $\dot{\neg}C$ to denote the NNF of $\neg C$.

(2) We may also assume that knowledge bases contain only assertions of the form $\ell : C$ and $C \doteq \top$. To see this, note first that distance assertions can be expressed using nominals and distance quantifiers:

$$\begin{array}{ll} \boldsymbol{\delta}(k,\ell) < a \text{ is equivalent to } k : \mathsf{E}^{<a}\ell, & \boldsymbol{\delta}(k,\ell) \leq a \text{ is equivalent to } k : \mathsf{E}^{\leq a}\ell, \\ \boldsymbol{\delta}(k,\ell) > a \text{ is equivalent to } k : \mathsf{A}^{\leq a}\neg\ell, & \boldsymbol{\delta}(k,\ell) \geq a \text{ is equivalent to } k : \neg\mathsf{A}^{<a}\neg\ell. \end{array}$$

Assertions of the form $C_1 \doteq C_2$ can be rewritten as $(C_1 \sqcap C_2) \sqcup (\dot{\neg}C_1 \sqcap \dot{\neg}C_2) \doteq \top$.

(3) Without loss of generality, we may assume that $par(\Sigma)$ contains only natural numbers: given a knowledge base Σ with $par(\Sigma) \subseteq \mathbb{Q}^+$, we may replace every element q of $par(\Sigma)$ with $q \cdot x$, where x is the least common multiple of the denominators of all elements of $par(\Sigma)$. It is then straightforward to show that any CD-model of the resulting knowledge base can be converted into a CD-model of Σ and vice versa.

We use α_Σ to denote the largest natural number that occurs in $par(\Sigma)$ and $M[\Sigma]$ to denote the smallest set satisfying the following conditions:

- $par(\Sigma) \subseteq M[\Sigma]$;
- if $a, b \in M[\Sigma]$ and $a + b < \alpha_\Sigma$, then $a + b \in M[\Sigma]$;
- if $a, b \in M[\Sigma]$ and $a - b > 0$, then $a - b \in M[\Sigma]$.

Having started on the input knowledge base Σ (in the form described above), the tableau algorithm considers only certain 'relevant' concepts. More precisely, we define the *closure* $cl(\Sigma)$ of Σ to be the (finite) set of concepts

$$\begin{array}{l} con(\Sigma) \cup \{\dot{\neg}C \mid C \in con(\Sigma)\} \cup \\ \qquad \{\mathsf{A}^{<a}C, \mathsf{A}^{\leq a}C \mid a \in M[\Sigma] \text{ and } \exists b \geq a \ \{\mathsf{A}^{\leq b}C, \mathsf{A}^{<b}C\} \cap con(\Sigma) \neq \emptyset\}. \end{array}$$

Similar to the set $cl(\Sigma)$ of relevant concepts, $M[\Sigma]$ describes the set of relevant numbers. However, the numbers in $M[\Sigma]$ are not enough: to distinguish between '$\leq a$' and '$< a$,' we require some additional symbols that will be used in the same way as numbers, namely, $M[\Sigma]^- = \{a^- \mid a \in M[\Sigma]\}$. Define a strict linear order $\prec$ on $M[\Sigma] \cup M[\Sigma]^-$ by setting

$$a_1^- \prec a_1 \prec a_2^- \prec a_2 \prec \cdots \prec a_n^- \prec a_n,$$

where $a_1 < a_2 < \cdots < a_n$.

We are in a position now to describe our tableau algorithm. Starting with Σ, it operates on *constraint systems* $\mathcal{S} = \langle T, <, L, S, E \rangle$, where

- $\langle T, < \rangle$ is a forest whose set of roots coincides with $ob(\Sigma)$;

- S is a *node labelling function* which associates with each $x \in T$ a set

$$S(x) \subseteq cl(\Sigma) \cup \{(R, \ell), (a, \ell), (a^-, \ell) \mid \ell \in ob(\Sigma), R \in rol(\Sigma), a \in M[\Sigma]\};$$

- L is a *labelling function* which associates with each pair $x, y \in T$ such that $x < y$ either a role name or a number from $M[\Sigma]$, or a symbol from $M[\Sigma]^-$;
- E is a set of inequalities between members of T.

Intuitively, we have $x < y$ if either x and y are related by some role R or the distance between x and y is known to be smaller than some value from $M[\Sigma]$. The purpose of the extra elements (R, ℓ) and (a, ℓ) in node labels is to represent additional edges that lead to nominals (roots in the forest), and whose explicit representation would destroy the forest structure.

The algorithm starts with $\mathcal{S}_0 = \langle T_0, <_0, L_0, S_0, E_0\rangle$, the *initial constraint system* for Σ, where

- $T_0 = ob(\Sigma)$,
- $S_0(\ell) = \{\ell\} \cup \{C \mid \ell : C \in \Sigma\}$, for every $\ell \in ob(\Sigma)$,
- $E_0 = \{\ell \neq \ell' \mid \ell \neq \ell',\ \ \ell, \ell' \in ob(\Sigma)\}$, and
- $<_0 = L_0 = \emptyset$.

Before describing the completion rules, we introduce some simplifying notation required to deal with edges represented via node labels. We write $\overline{L}(x, y) = a$ to express that either $x < y$ and $L(x, y) = a$ or that a is the $\prec$-minimum of $\{c \mid (c, y) \in S(x)\}$.[5] To account for the fact that, for some rules, it is not important whether a node is a predecessor or a successor, we write $L^o(\{x, y\}) = a$ if a is the $\prec$-minimum of $\{\overline{L}(x, y), \overline{L}(y, x)\}$. Finally, for a role name R, we say that y is an R-successor of x if either $x < y$ and $L(x, y) = R$ or $(R, y) \in S(x)$.

The *completion rules* are shown in Fig. 1. Constraint systems obtained by applying the completion rules to the initial constraint system for Σ will be called *constraint systems for Σ*. The terms 'blocked' and 'indirectly blocked' in the rule premises refer to a cycle detection mechanism that is needed to ensure termination of the algorithm. Before discussing the completion rules in more detail, let us formally introduce this mechanism. The general idea is that we stop the expansion of node labels if a node is labelled with exactly the same set of concepts as one of its $<$-ancestors. This simple approach works perfectly well, but it is not the most sensible thing we can do: the problem is that, due to the 'extra' concepts $\mathsf{A}^{<a}C$ and $\mathsf{A}^{\leq a}C$, the size of $cl(\Sigma)$ is *exponential* in the size of Σ rather than polynomial, and thus paths of the forest may grow to a length doubly exponential in Σ before the blocking occurs. Fortunately, this worst case can be avoided. When comparing node labels to check for a blocking situation, it is not necessary to take into account *all* of the extra $\mathsf{A}^{<a}C$ and $\mathsf{A}^{\leq a}C$ concepts: if, for example, we find $\mathsf{A}^{\leq a}C \in S(x)$, then it is clear that the object x also satisfies the concepts $\mathsf{A}^{\leq b}C$ for all $b \leq a$, even if they do not explicitly appear in the node label $S(x)$. This observation leads to the following, refined variant of blocking.

[5] This gives a well-defined value for $\overline{L}(x, y)$, as $(c, y) \in S(x)$ implies that y is a root.

For a node $x \in T$, we use $S^*(x)$ to denote the set of concepts $C \in S(x)$ such that one of the following conditions is satisfied:

1. C is not of the form $\mathsf{A}^{<a}D$ or $\mathsf{A}^{\leq a}D$;
2. C is of the form $\mathsf{A}^{\leq a}D$ and there is no $b > a$ such that $\mathsf{A}^{\leq b}D \in S(x)$;
3. C is of the form $\mathsf{A}^{<a}D$ and there is no $b > a$ such that $\mathsf{A}^{<b}D \in S(x)$.

Denote by $<^+$ the transitive closure of $<$. We say that a node $x \in T$ is *directly blocked by a node* y if $y <^+ x$, $S^*(x) = S^*(y)$, but for no distinct $u <^+ x$ and $v <^+ x$ do we have $S^*(u) = S^*(v)$. The $<^+$-successors of directly blocked nodes are called *indirectly blocked*. All directly or indirectly blocked nodes comprise the set of *blocked nodes*. Observe that the elements (R, ℓ) and (a, ℓ) of node labels are not taken into account for blocking.

Note that this blocking condition can be refined even further by taking into account implications between $\mathsf{A}^{\leq a}C$ and $\mathsf{A}^{<b}C$ concepts. We prefer to work with the above variant, since it suffices to restrict paths in forests to exponential length, and the more elaborate version makes proofs rather unreadable due to many additional case distinctions.

Let us now return to the completion rules. In what follows we assume that a rule can be applied to a tableau only if the tableau is changed. Such a rule will be called *applicable* to the tableau. The tableau algorithm applies the rules until either the obtained constraint system contains an obvious contradiction or no more rules are applicable. To be more precise, say that a constraint system $\mathcal{S}$ *contains a clash* if it contains a node x such that one of the following conditions hold:

1. $\{A, \dot{\neg} A\} \subseteq S(x)$, for some concept name A;
2. $\{\ell, \neg\ell\} \subseteq S(x)$ for some object name ℓ;
3. $\ell' \in S(\ell)$ for some object names $\ell' \neq \ell$;
4. $(x \neq x) \in E$;
5. for some R, $(\leq nR.C) \in S(x)$ and there are $n+1$ R-successors $y_0, \dots, y_n$ of x with $C \in L(y_i)$, for each $0 \leq i \leq n$ and $y_i \neq y_j \in E$ for each $0 \leq i < j \leq n$.

A constraint system $\mathcal{S}$ is *complete* if it either contains a clash or none of the rules in Fig. 1 is applicable to $\mathcal{S}$.

4 Termination, soundness and completeness

We show now that the tableau algorithm above always *terminates*, is *sound* (i.e., if there is a complete and clash-free constraint system for Σ, then Σ is satisfiable), and *complete* (i.e., if Σ is satisfiable, then the tableau algorithm eventually succeeds in finding a complete and clash-free complete system).

Termination

Theorem 1. *Any sequence of applications of tableau rules to the initial constraint system for Σ terminates after finitely many steps.*

$\mathsf{R}_\sqcap$ If $C_1 \sqcap C_2 \in S(x)$ and x is not indirectly blocked,
then set $S(x) := S(x) \cup \{C_1, C_2\}$.

$\mathsf{R}_\sqcup$ If $C_1 \sqcup C_2 \in S(x)$ and x is not indirectly blocked,
then set either $S(x) := S(x) \cup \{C_1\}$ or $S(x) := S(x) \cup \{C_2\}$.

$\mathsf{R}_=$ If $C = \top \in \Sigma$ and x is not indirectly blocked, then set $S(x) := S(x) \cup \{C\}$.

R_A If $\mathsf{A}^{<a}C \in S(x)$ or $\mathsf{A}^{\leq a}C \in S(x)$ and x is not indirectly blocked,
then set $S(x) := S(x) \cup \{C\}$.

$\mathsf{R}_{\mathsf{A}<}$ Let $\mathsf{A}^{<a}C \in S(x)$ and x is not indirectly blocked. Then:
if $L^o(\{y, x\}) = a^-$, then set $S(y) := \{C\} \cup S(y)$;
if $L^o(\{y, x\}) = b < a$, then set $S(y) := \{\mathsf{A}^{<a-b}C\} \cup S(y)$;
if $L^o(\{y, x\}) = b^-$ with $b < a$, then set $S(y) := \{\mathsf{A}^{\leq a-b}C\} \cup S(y)$.

$\mathsf{R}_{\mathsf{A}\leq}$ Let $\mathsf{A}^{\leq a}C \in S(x)$, $L^o(\{y, x\}) \in \{b, b^-\}$ and x is not indirectly blocked. Then:
if $b = a$, then set $S(y) := \{C\} \cup S(y)$;
if $b < a$, then set $S(y) := \{\mathsf{A}^{\leq a-b}C\} \cup S(y)$.

$\mathsf{R}_{\mathsf{E}<}$ If $\mathsf{E}^{<a}C \in S(x)$, x is not blocked, and
$\overline{L}(x, y) \notin \{b \mid b < a\} \cup \{b^- \mid b \leq a\}$ for any y with $C \in S(y)$,
then create a new node $y > x$ and set $L(x, y) := a^-$ and $S(y) := \{C\}$.

$\mathsf{R}_{\mathsf{E}\leq}$ If $\mathsf{E}^{\leq a}C \in S(x)$, x is not blocked and
$\overline{L}(x, y) \notin \{b \mid b \leq a\} \cup \{b^- \mid b \leq a\}$ for any y with $C \in S(y)$,
then create a new node $y > x$ and set $L(x, y) := a$ and $S(y) := \{C\}$.

R_{ch} If $\{(\geq nR.C), (\leq nR.C)\} \cap S(x) \neq \emptyset$, x is not blocked and y is an
R-successor of x, then set $S(y) := S(y) \cup \{C\}$ or $S(y) = S(y) \cup \{\dot{\neg} C\}$.

$\mathsf{R}_\geq$ If $(\geq nR.C) \in S(x)$, x is not blocked, and there are no R-successors $y_1, \ldots, y_n$
with $C \in S(y_i)$ and $y_i \neq y_j \in E$, for all $i \neq j$, then take new $y_1 > x, \ldots, y_n > x$
and set $L(x, y_i) := R$, $S(y_i) := \{C\}$, $E := E \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$.

$\mathsf{R}_\leq$ If $(\leq nR.C) \in S(x)$, x is not blocked, has $n+1$ R-successors $y_0, \ldots, y_n$
with $C \in S(y_i)$ for all i, and, for some $i, j \leq n$, $y_i \neq y_j \notin E$ and $y_j \notin ob(\Sigma)$,
then set $E := E \cup \{y \neq y_i \mid y \neq y_j \in E\}$, $S(y_i) := S(y_i) \cup S(y_j)$,
$S(x) := S(x) \cup \{(R', \ell) \mid R' = L(x, y_j)\}$, if $y_i = \ell \in ob(\Sigma)$,
and finally delete y_j and all z with $y_j <^+ z$ from T.

R_ℓ If $\ell \in S(x)$, $x \notin ob(\Sigma)$, and x is not indirectly blocked,
Then set $S(\ell) := S(\ell) \cup S(x)$, and, for every y,
$S(y) := S(y) \cup \{(c, \ell) \mid c = \overline{L}(y, x)$ or $c = R$ a role and x an R-successor of $y\}$,
$E := E \cup \{y \neq \ell \mid y \neq x \in E\}$, and delete x and all z with $x <^+ z$ from T.

Fig. 1. Tableau rules.

Proof. Let $m_0 = |con(\Sigma)|$ and m_q be the maximal number occurring in qualified number restrictions of Σ. Termination follows from the following five observations.

(1) Each rule except $\mathsf{R}_\leq$ and R_ℓ strictly extends the constraint system. Moreover, neither R_ℓ nor $\mathsf{R}_\leq$ removes concepts from nodes.

(2) None of the generating rules $\mathsf{R}_{\mathsf{E}<}$, $\mathsf{R}_{\mathsf{E}\leq}$, $\mathsf{R}_\geq$ can be applied more than once to a given node and a given concept.

Suppose that $\mathsf{R}_{\mathsf{E}<}$ is applied to a node x, generates y with $x < y$ and updates $L(x, y) = a^-$ and $S(y) = \{C\}$. The only reason why $\mathsf{R}_{\mathsf{E}<}$ could be applied once again to x and $\mathsf{E}^{<a}C$ is that later on y is removed by an application of $\mathsf{R}_\leq$ or R_ℓ. However, unless x is removed (in this case the claim is trivial) y cannot be removed by an application of $\mathsf{R}_\leq$ because we do not find a z and a role R with $R = L(z, y)$. Suppose y is removed by an application of R_ℓ because $\ell \in S(y)$. Then, after the application of R_ℓ, we have $(a^-, \ell) \in S(x)$ and $C \in S(\ell)$, since $a^- = L(x, y)$. But then, since a node of the form ℓ is never removed, the rule $\mathsf{R}_{\mathsf{E}<}$ is not applicable to x and $\mathsf{E}^{<a}C$ afterwards. The rule $\mathsf{R}_{\mathsf{E}\leq}$ is considered analogously.

Suppose that $\mathsf{R}_\geq$ is applied to a node x, generates $y_1, \ldots, y_n$ with $x < y_i$ and updates $L(x, y_i) = R$, $S(y_i) = \{C\}$, and $E = E \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$. Now, whenever some y_j is removed by $\mathsf{R}_\geq$ or R_ℓ and x is not removed, after the removal of y_j we still have n R-successors $z_1, \ldots, z_n$ of x such that $C \in S(z_i)$, $E \supseteq \{z_i \neq z_j \mid 1 \leq i < j \leq n\}$. So, $\mathsf{R}_\geq$ is not applied to x after such a removal.

(3) The out-degree of the forest constructed using the tableaux rules is bounded by $m_0 + m_q \cdot m_0$. This follows from (2) and the fact that nodes are labelled with subsets of the set

$$cl(\Sigma) \cup \{(R, \ell), (a, \ell), (a^-, \ell) \mid \ell \in ob(\Sigma), R \in rol(\Sigma), a \in M[\Sigma]\}.$$

(4) If a node x is removed, then all z with $x <^+ z$ are removed as well

(5) No $<$-branch in any constraint system for Σ can ever be of length exceeding $2^{m_0} \cdot |M[\Sigma]|^2$, since no node introducing rule can be applied to a node x such that $S^*(y) = S^*(z)$ for two distinct $y, z \leq x$.

Soundness

Before proving the soundness of the tableau algorithm, we introduce a relational semantics for *sim*-$\mathcal{ALCQO}$. This semantics comprises, for each $a \in M[\Sigma]$, additional binary relations R_a and S_a such that, intuitively, we have uR_av if the distance between u and v is at most a, and uS_av if the distance between u and v is less than a. Formally, a *Kripke model for* Σ is a structure of the form

$$\mathfrak{M} = \left\langle W, A_1^{\mathfrak{M}}, \ldots, R_1^{\mathfrak{M}}, \ldots, (R_a)_{a \in M[\Sigma]}, (S_a)_{a \in M[\Sigma]}, \ell_1^{\mathfrak{M}}, \ldots \right\rangle$$

satisfying, for all $u, v, w \in W$ and all $a, b \in M[\Sigma]$, the following conditions:

($\mathrm{S1}_R$) if uR_av and $a \leq b$, then uR_bv,
($\mathrm{S2}_R$) uR_av iff vR_au,

(S3$_R$) $uR_a u$,
(S4$_R$) if $uR_a v$, $vR_b w$ and $a+b \in M[\Sigma]$, then $uR_{a+b}w$,
(S1$_S$) if $uS_a v$ and $a \leq b$, then $uS_b v$;
(S2$_S$) $uS_a v$ iff $vS_a u$;
(S3$_S$) $uS_a u$,
(S4$_S$) if $uS_a v$, $vS_b w$ and $a+b \in M[\Sigma]$, then $uS_{a+b}w$,
(C1) if $uS_a v$ then $uR_a v$,
(C2) if $uR_a v$ and $a < b$, then $uS_b v$,
(C3) if $uR_a v$, $vS_b w$ and $a+b \in M[\Sigma]$, then $uS_{a+b}w$,
(C4) if $uS_a v$, $vR_b w$ and $a+b \in M[\Sigma]$, then $uS_{a+b}w$.

The *value* $C^{\mathfrak{M}}$ of a concept C in $\mathfrak{M}$ and the *truth-relation* $\mathfrak{M} \models C_1 \doteq C_2$ are defined in almost the same way as for CD-models: we only replace $\mathfrak{B}$ with $\mathfrak{M}$ and define the clauses for the distance quantifiers as follows:

$$
\begin{aligned}
(\mathsf{E}^{\leq a}C)^{\mathfrak{M}} &= \{x \in W \mid \exists y \in W \left(xR_a y \wedge y \in C^{\mathfrak{M}}\right)\},\\
(\mathsf{E}^{<a}C)^{\mathfrak{M}} &= \{x \in W \mid \exists y \in W \left(xS_a y \wedge y \in C^{\mathfrak{M}}\right)\},\\
(\mathsf{A}^{\leq a}C)^{\mathfrak{M}} &= \{x \in W \mid \forall y \in W \left(xR_a y \rightarrow y \in C^{\mathfrak{M}}\right)\},\\
(\mathsf{A}^{<a}C)^{\mathfrak{M}} &= \{x \in W \mid \forall y \in W \left(xS_a y \rightarrow y \in C^{\mathfrak{M}}\right)\}.
\end{aligned}
$$

The next theorem ensures that the alternative Kripke semantics is 'equivalent' to the original one.

Theorem 2. *The knowledge base Σ is satisfiable in a CD-model iff it is satisfiable in a Kripke model for Σ.*

Proof. ($\Rightarrow$) Suppose that Σ is satisfied in a CD-model

$$\mathfrak{B} = \left\langle W, d, A_1^{\mathfrak{B}}, \dots, R_1^{\mathfrak{B}}, \dots, \ell_1^{\mathfrak{B}}, \dots \right\rangle .$$

Define a Kripke model

$$\mathfrak{M} = \left\langle W, A_1^{\mathfrak{M}}, \dots, R_1^{\mathfrak{M}}, \dots, (R_a)_{a \in M[\Sigma]}, (S_a)_{a \in M[\Sigma]}, \ell_1^{\mathfrak{M}}, \dots \right\rangle$$

for Σ by taking, for $a \in M[\Sigma]$,

- $A_i^{\mathfrak{M}} = A_i^{\mathfrak{B}}$, $\ell_i^{\mathfrak{M}} = \ell_i^{\mathfrak{B}}$, and $R_i^{\mathfrak{M}} = R_i^{\mathfrak{B}}$;
- $xR_a y$ iff $d(x,y) \leq a$;
- $xS_a y$ iff $d(x,y) < a$.

It is not difficult to see that $\mathfrak{M}$ is a Kripke model for Σ and to prove by induction that $C^{\mathfrak{M}} = C^{\mathfrak{B}}$, for all $C \in cl(\Sigma)$. It follows that $\mathfrak{M}$ satisfies Σ.

($\Leftarrow$) Suppose now that Σ is satisfied in a Kripke model

$$\mathfrak{M} = \left\langle W, A_1^{\mathfrak{M}}, \dots, R_1^{\mathfrak{M}}, \dots, (R_a)_{a \in M[\Sigma]}, (S_a)_{a \in M[\Sigma]}, \ell_1^{\mathfrak{M}}, \dots \right\rangle$$

for Σ. Let $M[\Sigma] = \{a_1, \dots, a_N\}$ with $0 < a_1 < a_2 < \dots < a_N$. Choose a rational number $\gamma_\Sigma > a_N$ in such a way that there are no $a_1, a_2 \in M[\Sigma]$ with $a_N < a_1 + a_2 \leq \gamma_\Sigma$. Let D be the minimal number in the set

$$M[\Sigma] \cup \{a_1 + a_2 - \gamma_\Sigma \mid a_1, a_2 \in M[\Sigma] - \{\gamma_\Sigma\} \ \&\ a_1 + a_2 > \gamma_\Sigma\}.$$

Take some positive $\epsilon < \frac{D}{2^{N+1}}$. Define a function $d : W \times W \to \mathbb{R}$ by taking $d(u,v) = 0$ if $u = v$ and otherwise

$$d(u,v) = \begin{cases} \gamma_\Sigma, & \text{if } \neg\exists a \in M[\Sigma]\ uR_a v, \\ a, & \text{if } \exists a \in M[\Sigma]\ (uR_a v \wedge \neg uS_a v), \\ a_i - 2^i \cdot \epsilon, & \text{if } \exists a_i \in M[\Sigma]\ (uS_{a_i} v \wedge \forall j\ (0 < j < i \to \neg uR_{a_j} v)). \end{cases}$$

Consider the model

$$\mathfrak{B} = \left\langle W, d, A_1^{\mathfrak{B}}, \ldots, R_1^{\mathfrak{B}}, \ldots, \ell_1^{\mathfrak{B}}, \ldots \right\rangle .$$

where $A_i^{\mathfrak{B}} = A_i^{\mathfrak{M}}$, $R_i^{\mathfrak{B}} = R_i^{\mathfrak{M}}$, and $\ell_i^{\mathfrak{B}} = \ell_i^{\mathfrak{M}}$ for all i. One can show now that $\mathfrak{B}$ is a CD-model satisfying Σ.

Thus, it suffices to prove soundness with respect to Kripke semantics.

Theorem 3. *If there exists a complete and clash-free constraint system for Σ, then Σ is satisfiable in a Kripke model for Σ.*

Proof. Suppose that $\mathcal{S} = \langle T, <, S, L, E \rangle$ is a complete and clash-free constraint system for Σ that is obtained by repeatedly applying completion rules from Fig. 1 to the initial constraint system $\langle T_0, <_0, S_0, L_0, E_0 \rangle$. We use this constraint system to construct a Kripke model

$$\mathfrak{M} = \left\langle W, A_1^{\mathfrak{M}}, \ldots, R_1^{\mathfrak{M}}, \ldots, (R_a)_{a \in M[\Sigma]}, (S_a)_{a \in M[\Sigma]}, \ell_1^{\mathfrak{M}}, \ldots \right\rangle$$

satisfying Σ. Denote by T^i the set of nodes from T that are not indirectly (but possible directly) blocked. The domain W of $\mathfrak{M}$ consists of all sequences of the form $\langle \ell, x_1, \ldots, x_k \rangle$, where $\ell \in ob(\Sigma)$ and $x_1, \ldots, x_k \in T^i$ (with $k \geq 0$) such that $\ell < x_1$ and, for $1 \leq i < k$, either (i) x_i is unblocked and $x_i < x_{i+1}$ or (ii) there is a z such that z directly blocks x_i and $z < x_{i+1}$. Role names R are interpreted by setting

- $(\langle \ell_1, x_1, \ldots, x_k \rangle, \langle \ell_2 \rangle) \in R^{\mathfrak{M}}$ iff x_k is not blocked and $(R, \ell_2) \in S(x_k)$, or there exists z which directly blocks x_k such that $(R, \ell_2) \in S(z)$;
- $(\langle \ell, x_1, \ldots, x_k \rangle, \langle \ell, x_1, \ldots, x_{k+1} \rangle) \in R^{\mathfrak{M}}$ iff one of the following holds:
  - x_i is not blocked, $x_k < x_{k+1}$, and $L(x_k, x_{k+1}) = R$;
  - there is z which directly blocks x_k, $z < x_{k+1}$ and $L(z, x_{k+1}) = R$.

Given $\overline{x} = \langle \ell, x_1, \ldots, x_k \rangle \in W$, let $S(\overline{x})$ denote $S(x_k)$. We now define the relations R_a and S_a. Let R_a be the set of pairs $(\overline{x}, \overline{y}) \in W \times W$ such that, for $\{\overline{u}, \overline{v}\} = \{\overline{x}, \overline{y}\}$, the following conditions are satisfied:

(a) $\mathsf{A}^{\leq a} C \in S(\overline{u})$ implies $C \in S(\overline{v})$;
(b) $\mathsf{A}^{\leq b} C \in S(\overline{u})$ and $b > a$ imply that $\mathsf{A}^{\leq c} C \in S(\overline{v})$ for some $c \geq b - a$;
(c) $\mathsf{A}^{< b} C \in S(\overline{u})$ and $b > a$ imply that $\mathsf{A}^{< c} C \in S(\overline{v})$ or $\mathsf{A}^{\leq c} C \in S(\overline{v})$ for some $c \geq b - a$.

Similarly, S_a is comprised of the pairs $(\overline{x}, \overline{y}) \in W \times W$ such that, for $\{\overline{u}, \overline{v}\} = \{\overline{x}, \overline{y}\}$, the following conditions are satisfied:

(d) $\mathsf{A}^{<a}C \in S(\overline{u})$ implies $C \in S(\overline{v})$;
(e) $\mathsf{A}^{\leq b}C \in S(\overline{u})$ and $b > a$ imply that $\mathsf{A}^{\leq c}C \in S(\overline{v})$ for some $c \geq b - a$;
(f) $\mathsf{A}^{<b}C \in S(\overline{u})$ and $b > a$ imply that $\mathsf{A}^{<c}C \in S(\overline{v})$ or $\mathsf{A}^{\leq c}C \in S(\overline{v})$ for some $c \geq b - a$.

For all $\ell \in ob(\Sigma)$, we set $\ell^{\mathfrak{M}} = \{\langle \ell \rangle\}$. This is well-defined, since no nominal is removed from the tableau. Finally, for all concept names A_i and $\overline{x} \in W$, we set $\overline{x} \in A_i^{\mathfrak{M}}$ iff $A_i \in S(\overline{x})$. $\mathfrak{M}$ is a Kripke models for Σ which Σ. A proof of this claim can be found in the full version of this paper.

Completeness

Let us say that a model $\mathfrak{B} = \langle W, d, A_1^{\mathfrak{B}}, \ldots, \ell_1^{\mathfrak{B}}, \ldots \rangle$ *realises* a constraint system $\langle T, <, L, S, E \rangle$ for Σ if $\mathfrak{B} \models \Sigma$ and there exists a map $\rho : T \to W$ such that

- $C \in S(x)$ implies $\rho(x) \in C^{\mathfrak{B}}$;
- $L^o(\{x, y\}) = a \in M[\Sigma]$ implies $d(\rho(x), \rho(y)) \leq a$;
- $L^o(\{x, y\}) = a^- \in M[\Sigma]^-$ implies $d(\rho(x), \rho(y)) < a$;
- $x \neq y \in E$ implies $\rho(x) \neq \rho(y)$;
- if y is an R-successor of x, then $(\rho(x), \rho(y)) \in R^{\mathfrak{B}}$.

The following lemma is an immediate consequence of the definitions:

Lemma 1. *If a knowledge base Σ is satisfied in a CD-model $\mathfrak{B}$, then the initial constraint system for Σ is realisable in $\mathfrak{B}$.*

Lemma 2. *Suppose that $\mathfrak{B}$ realises a constraint system $\mathcal{S} = \langle T, <, L, S, E \rangle$ for Σ and a completion rule R is applicable to $\mathcal{S}$. Then R can be applied in such a way that $\mathfrak{B}$ realises the resulting constraint system $\mathcal{S}' = \langle T', <', S', L', E' \rangle$ as well.*

Proof. Let $\mathfrak{B} = \langle W, d, A_1^{\mathfrak{B}}, \ldots, \ell_1^{\mathfrak{B}}, \ldots \rangle$ realise $\mathcal{S}$ by means of a map $\rho : T \to W$ and let $\mathcal{S}'$ be obtained from $\mathcal{S}$ using some rule R. We consider only two rules, $\mathsf{R} = \mathsf{R}_{\mathsf{E}\leq}$ and $\mathsf{R} = \mathsf{R}_{\mathsf{A}<}$, and and leave the remaining cases to the reader.

$\mathsf{R}_{\mathsf{E}\leq}$: Suppose that $\mathsf{E}^{\leq a}C \in S(x)$, $T' = T \cup \{y\}$, $L'(\{x, y\}) = a$, $<' = < \cup \{(x, y)\}$, and $S(y) = \{C\}$. We know that $\rho(x) \in (\mathsf{E}^{\leq a}C)^{\mathfrak{B}}$. So we can find $v \in W$ such that $d(\rho(x), v) \leq a$ and $v \in C^{\mathfrak{B}}$. Define a map $\rho' : T' \to W$ by taking $\rho'(z) = \rho(z)$ for all $z \in T$ and $\rho'(y) = v$. It should be clear that $\mathfrak{B}$ realises $\mathcal{S}'$ my means of ρ'.

$\mathsf{R}_{\mathsf{A}<}$: Let $\mathsf{A}^{<a}C \in S(x)$, $x \in T$. Suppose that the rule is applied to some $y \in T$. Consider three possible cases.

(i) If $L^o(\{x, y\}) = a^-$ then $d(\rho(x), \rho(y)) < a$ and $S(y) = \{C\} \cup S(y)$. We need to show that $\rho(y) \in C^{\mathfrak{B}}$. But this follows immediately from $\rho(x) \in (\mathsf{A}^{<a}C)^{\mathfrak{B}}$.

(ii) If $L^o(\{y, x\}) = b < a$ then $d(\rho(x), \rho(y)) \leq b$ and $S(y) = \{\mathsf{A}^{<a-b}C\} \cup S(y)$. To show that $\rho(y) \in (\mathsf{A}^{<a-b}C)^{\mathfrak{B}}$, take any $v \in W$ such that $d(\rho(y), v) < a - b$. By the triangular inequality, we then have $d(\rho(y), v) < a$ and so $v \in C^{\mathfrak{B}}$.

(iii) The case of $L^o(\{y, x\}) = b^-$ and $b < a$ is considered similarly to (ii).

As a consequence of these two lemmas and Theorem 1 we obtain

Theorem 4. *If Σ is satisfiable, then there exists a complete clash-free constraint system for Σ.*

5 Undecidability

We show now that a rather natural and closer integration of distance quantifiers and qualified number restrictions results in an undecidable logic. Denote by sim_f the language with the following concept formation rule:

$$C ::= A_i \mid \ell_i \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \mathsf{E}^{\leq a} C \mid (\leq_a^1 .C),$$

where $(\leq_a^1 .C)$ is interpreted in concept distance models $\mathfrak{B}$ as follows

$$(\leq_a^1 .C)^{\mathfrak{B}} = \{x \in W \mid \left|\{y \mid d(x,y) \leq a, y \in C^{\mathfrak{B}}\}\right| \leq 1\}.$$

Theorem 5. *The satisfiability problem for sim_f-knowledge bases in concept distance models is undecidable.*

Proof. (sketch) We can simulate the undecidable $\mathbb{N} \times \mathbb{N}$-tiling problem in almost the same way as in the undecidability proof of [9] for the language $\mathcal{MS}_1$ with the operators $\mathsf{A}^{\leq a}$, $\mathsf{A}^{>0}_{\leq a}$ and their duals: just replace everywhere in the proof of Theorem 3.1 the concept $\mathsf{A}^{>0}_{\leq 80} \neg \chi_{i,j}$ by the concept $(\leq_{80}^1 .\chi_{i,j})$.

6 Conclusion

We have introduced the description-metric logic *sim*-$\mathcal{ALCQO}$ for defining concepts based on similarity measures, and have proposed a tableau algorithm for deciding the satisfiability of *sim*-$\mathcal{ALCQO}$-knowledge bases. This algorithm unifies the tableau algorithms for $\mathcal{SHOQ}$ (a superlogic of $\mathcal{ALCQO}$) presented in [5] and for the logic of metric spaces $\mathcal{MS}$ as defined in [13]. It is of interest to note that, in contrast to what is done in [13], we need a different soundness proof, since the presence of number restrictions prohibits the use of filtration techniques.

We regard the presented logic only as a first step towards DLs that allow definitions of concepts based on similarity measures. Although we believe that the expressive power provided by *sim*-$\mathcal{ALCQO}$ is quite natural and useful, an in-depth investigation of the expressive means that are useful for defining vague concepts are in order. Some possible extensions of *sim*-$\mathcal{ALCQO}$ are the following:

(1) New constructors $\mathsf{E}^{<a} R.C$ and $\mathsf{A}^{<a} R.C$, where the former expresses that there exists an R-successor at distance smaller than a satisfying C, and the latter is its dual. Such constructors would, e.g., allow us to say that a person is very similar to his father: $\mathsf{E}^{<0.5}\mathsf{parent}.\mathsf{Male}$. The presented algorithm should be extendable to this case without any problems.

(2) New constructors $\mathsf{E}^{>a} C$ and $\mathsf{E}^{\geq a} C$ (and their duals) with the obvious semantics. Although these constructors do not seem to be so natural as the variants based on $<$ and $\leq$, they could, e.g., be used to express that a propotypical tableau algorithm pta is very close to *all* other tableau algorithms: $\mathsf{pta} : \mathsf{A}^{>0.5} \neg \mathsf{Tableau\_algorithm}$. While [9] proves the decidability of the metric logic with the operators $\mathsf{E}^{\leq a} C$ and $\mathsf{E}^{>a} C$ (and their duals), nothing is currently known about the extension of $\mathcal{MS}$ with all four possible constructors.

Acknowledgements:

The work of the second author was supported by Deutsche Forschungsgemeinschaft (DFG) grant Wo583/3-3. The work of the third author was partially supported by U.K. EPSRC grants no. GR/R45369/01 and GR/R42474/01.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *J. of Artificial Intelligence Research*, 16:1–58, 2002.
3. K. Fine and G. Schurz. Transfer theorems for stratified modal logics. In J. Copeland, editor, *Logic and Reality, Essays in Pure and Applied Logic. In memory of Arthur Prior*, pages 169–213. Oxford University Press, 1996.
4. I. Horrocks and P. Patel-Schneider. The generation of DAML+OIL. In C. Goble, D. McGuinness, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the International Workshop in Description Logics 2001 (DL2001)*, number 49 in CEUR-WS (http://ceur-ws.org/), pages 30–35, 2001.
5. I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 199–204. Morgan Kaufmann, 2001.
6. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In D. MacAllester, editor, *Proc. of the 17th International Conference on Automated Deduction (CADE-17)*, number 1831 in LNCS. Springer, 2000.
7. I. Horrocks, P. Patel-Schneider, and F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, 2002.
8. M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *J. Symbolic Logic*, 56:1469–1485, 1991.
9. O. Kutz, H. Sturm, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev. Logics of metric spaces. *ACM Transactions on Computational Logic*, 2003. In print.
10. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.
11. E. Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.
12. F. Wolter. Fusions of modal logics revisited. In M. Kracht, M. De Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 1, pages 361–379. CSLI, Stanford, 1997.
13. F. Wolter and M. Zakharyaschev. Reasoning about distances. To appear in Proc. IJCAI 2003.