A logic for concepts and similarity

M. Sheremet[1], D. Tishkovsky[2], F. Wolter[2] and M. Zakharyaschev[1]

[1]School of Computer Science and Information Systems, Birkbeck College
Malet Street, London WC1E 7HX, U.K.
{mikhail,michael}@dcs.bbk.ac.uk

[2]Department of Computer Science
University of Liverpool
Liverpool L69 3BX, U.K.
{dmitry,frank}@csc.liv.ac.uk

Abstract

Categorisation of objects into classes is currently supported by (at least) two 'orthogonal' methods. In logic-based approaches, classifications are defined through ontologies or knowledge bases which describe the existing relationships among terms. Description logic (DL) has become one of the most successful formalisms for representing such knowledge bases, in particular because theoretically well-founded and efficient reasoning tools have been readily available.

In numerical approaches, classifications are obtained by first computing similarity (or proximity) measures between objects and then categorising them into classes by means of Voronoi tessellations, clustering algorithms, nearest neighbour computations, etc.

In many areas such as bioinformatics, computational linguistics or medical informatics, these two methods have been used independently of each other: although both of them are often applied to the same domain (and even by the same researcher), up to now no formal interaction mechanism has been developed.

In this paper, we propose a DL-based integration of the two classification methods. Our formalism, called $\mathcal{SL} + \mathcal{ALCQIO}$, extends the expressive DL $\mathcal{ALCQIO}$ by means of the constructors of the similarity logic $\mathcal{SL}$ which allow definitions of concepts in terms of both comparative and absolute similarity. In the combined knowledge base the user should declare the similarity spaces where the new operators are interpreted.

Of course, $\mathcal{SL} + \mathcal{ALCQIO}$ can only be useful if classifications with this logic are supported by automated reasoning tools. We lay theoretical foundations for the development of such tools by showing that reasoning problems for $\mathcal{SL} + \mathcal{ALCQIO}$ can be decomposed into the corresponding problems for its DL-part $\mathcal{ALCQIO}$ and similarity part $\mathcal{SL}$. Then we investigate reasoning in $\mathcal{SL}$ and prove that consistency and many other reasoning problems are ExpTime-complete for this logic. Using this result and a recent complexity result of Pratt-Hartmann for $\mathcal{ALCQIO}$, we prove that reasoning in $\mathcal{SL} + \mathcal{ALCQIO}$ is NExpTime-complete.

As the 'closer' operator of $\mathcal{SL}$ has the same expressive power as the standard implication $>$ of conditional logic, these results may have interesting consequences for conditional logic as well.

Keywords: similarity logic, description logic, ontology, complexity, conditional logic.

1 Introduction

Classification — "the actual or ideal arrangement together of those which are like, and the separation of those which are unlike" (see [28, 23] and references therein) — is said to be one of the main objectives of science, the Linnaean taxonomy and the Mendeleev periodic table being probably the best known witnesses. Modern classification and conceptualisation problems in bioinformatics, linguistics or the World Wide Web are characterised by huge numbers of objects to be categorised and/or their very complex structures. Building adequate 'ontologies' for such areas and developing powerful tools for dealing with them have been recognised among the most important and urgent tasks, and can be regarded as a real challenge for the knowledge representation and reasoning (KR&R) community.

There are (at least) three general ways of categorising things into classes:

- by establishing logical relationships between them (e.g., 'a mother is a woman with a child'),
- by identifying prototypical (the most representative) instances of classes against which the surrounding 'poorer' instances are categorised (e.g., a colour is blueish if it is more similar to the prototypical colour blue than to the prototypical colours green, red, etc.),
- by means of clustering, that is, by partitioning a set of objects into subsets (clusters), so that the objects in each subset are more similar to each other than to the objects in other subsets.

From the KR point of view, the main difference between these approaches is that the first one is *logic-based* and usually involves building knowledge bases or ontologies in such formalisms as description logic or (fragments of) first-order logic, while the last two are 'quantitative' in that they are based on implicit or explicit similarity (or proximity) measures and use *numerical computations* (say, Voronoi tessellations, nearest neighbour or clustering algorithms). Successful examples of the logic-based approach are such well-known ontologies in medicine and bioinformatics as SNOMED (`www.snomed.org`) or GO (`www.geneontology.org`). On the other hand, in those (still huge) areas of bioinformatics, computational linguistics, etc. where there exist vast amounts of poorly investigated and unstructured data, the quantitative, similarity-based methods (say, using alignment algorithms and tools like BLAST [1]) often seem to be the only choice.

The apparent 'qualitative–quantitative orthogonality' of these approaches was probably the reason why they have not been formally integrated — despite the fact that researchers in bioinformatics and linguistics as well as 'ordinary people' in their everyday life use all these methods frequently and in parallel. We will illustrate this point by two simple examples from the 'World of Whisky.' The classification at `www.whiskymag.com` can easily be represented in a sufficiently expressive description logic. For instance, the definition '*vatted malt Speyside whisky is malt whisky blended from several distilleries located in the Speyside region of Scotland*' can be formalised as

$$\begin{aligned}&\mathsf{Vatted\_malt\_Speyside\_whisky} \equiv \\ &\qquad\qquad \mathsf{Malt\_whisky} \sqcap \geq 2\ \mathsf{made\_by.Distillery} \sqcap \forall \mathsf{made\_by}.\exists \mathsf{is\_located\_in}.\{\mathsf{Speyside}\}\end{aligned}$$

One would hardly like to replace this clear axiom with a quantitative similarity-based representation. In general, we do not see any evidence that ontologies could be built purely on the basis of, say, conceptual similarity spaces in the sense of [10].

On the other hand, it appears to be extremely difficult to define by means of logical relations the taste characteristics of whiskies in terms of natural factors such as the geographical region, the type of rock from which the water source originates, the amount of peat, etc.

For example, the classification of malt whiskies according to their taste in [36] is based on computing a proximity matrix and does not correspond to more traditional regional (and other) classifications ('the regional classification is really only helpful today if you are trying to visit a distillery'). Again, there appears to be no evidence that logic-based formalisms could replace such similarity-based classifications.

Given that both approaches are applied for classifying objects from the same domain (and, moreover, can potentially correct errors of each other[1]), it seems natural to try integrate them into a single knowledge representation formalism and support it with suitable reasoning and classification tools.

In this paper we propose an integration of logic-based and similarity-based approaches of the following form. A knowledge base consists of three ingredients: a set of axioms fixing the terminology (usually called a TBox), a set of assertions about individuals (an ABox), and a declaration of similarity spaces that are used in the TBox and ABox. The TBox defines logical relations between concepts as it is by now standard in description logics (DLs); see the example above. However, in contrast to standard DLs, we do not only consider definitions within abstract spaces of concepts, but also those based on similarity measures (these similarity measures should not be confused with *semantic similarity* considered, e.g., in [27]). First, we allow definitions by means of 'comparative similarity.' To be able to use comparative similarity in concept definitions, the user has to declare a similarity space by means of an atomic concept comprising the elements of the space and certain constraints on the space (saying that it is symmetric and/or satisfies the triangle inequality; for a discussion of these properties of similarity see, e.g., [34]). For example, in the World of Whisky, one could declare a similarity space for comparing 'tastes' by introducing a concept name Taste which comprises the set of all elements of the (possibly not completely specified) similarity space with a similarity measure d_{Taste} on it. Using the role $\mathsf{has\_taste}$ and its inverse $\mathsf{taste\_of}$, the ontology designer can first introduce the taste of a whisky (e.g., of Caol Ila) by the axiom

$$\mathsf{T\_Caol\_Ila} \;\equiv\; \mathsf{Taste} \sqcap \exists \mathsf{taste\_of}.\,\{\mathsf{Caol\_Ila}\}$$

and then define, for example,

$$\mathsf{Peaty\_malt} \;\equiv\; \mathsf{Single\_malt} \sqcap \exists \mathsf{has\_taste}.\big(\mathsf{T\_Caol\_Ila} \leftleftarrows_{\mathsf{Taste}} \mathsf{T\_Macallan} \sqcup \cdots \sqcup \mathsf{T\_Glenmorangie}\big).$$

Intuitively, this TBox axiom states that peaty malt whiskies are those single malt whiskies that are *more similar regarding their taste* to (the prototype) Caol Ila than to (the prototypes) Macallan, ..., Glenmorangie. Notice that, as Taste is a concept, we can further refine the notion of taste in the TBox by means of various flavour and taste descriptors: for example,

$$\mathsf{Taste} \;\sqsubseteq\; \exists \mathsf{has\_taste\_descriptor}.\,(\mathsf{Winey} \sqcup \mathsf{Woody} \sqcup \cdots \sqcup \mathsf{Peaty}).$$

Along with comparative similarity, the user has an option to define concepts in terms of 'absolute similarity' which can, for example, be found using numerical techniques in the proximity matrix:

$$\mathsf{T\_Caol\_Ila} \;\sqsubseteq\; \neg \exists^{\leq 0.67}_{\mathsf{Taste}} \exists \mathsf{taste\_of}.\,\mathsf{Medium\_sweet\_whisky}$$

says that the taste of Caol Ila cannot be too close (at most 0.67 close) to the taste of medium sweet whiskies.

[1]For example, the clustering algorithm in [19] places the Hungarian language into the same family as Turkish, which contradicts the existing ontologies.

Besides the TBox, the knowledge base contains an ABox which provides data about concrete objects. First, one can specify within description logic:

$$\mathsf{Lagavulin} : \mathsf{Dry\_whisky} \qquad\qquad \mathsf{Lagavulin\_Distillery}\ \mathsf{is\_located\_in}\ \mathsf{Islay}$$

where the former says that Lagavulin is a dry whisky. Second, one can include information about similarities between concrete objects by writing down (part of the) proximity matrix in assertions of the form

$$d_{\mathsf{Taste}}(\ell_1, \ell_2) \; = \; 0.15 \qquad\qquad d_{\mathsf{Taste}}(\ell_3, \ell_4) < 1$$

The resulting knowledge representation formalism, called $\mathcal{SL}+\mathcal{ALCQIO}$, is a combination of two logics. One of them is a suitable DL (see [2] for a survey); in this paper we consider the expressive DL $\mathcal{ALCQIO}$, but it could be replaced with any other well-behaved DL. The second one is the similarity logic $\mathcal{SL}$, which itself is a combination of two logics:

- the logic $\mathcal{MS}$ of 'absolute' similarity with concept constructors of the form $\exists^{<a}C$ and $\exists^{\leq a}C$, that is, 'in the a-neighbourhood of C' (excluding or including the boundary), where a is a positive rational number (this and other similar logics were first introduced and investigated in [17, 37, 22, 38]), and
- the logic $\mathcal{CSL}$ of 'comparative' similarity with the 'closer' operator $C \leftleftarrows D$ which is interpreted by the set of all points in the similarity space that are closer to the instances of C than to the instances of D. For example, it can be used to model statements like 'X resembles C more than D.' This logic was first introduced in [30]. $\mathcal{CSL}$ is closely related to conditional logics [18, 29]; we will briefly discuss this connection in Section 5.

The applications of $\mathcal{SL} + \mathcal{ALCQIO}$ we envisage are similar to the use of DL in the process of ontology formation and maintenance, which means that *reasoning* with the logic is of fundamental importance (to check whether the resulting classification is consistent, whether it reflects properly the available knowledge, etc.); see [15, 24, 14] for the pure DL-reasoners Fact, Pellet, and Racer.

The *main contributions of this paper* are as follows:

1. An analysis of the computational complexity of reasoning in the similarity logics $\mathcal{SL}$ and $\mathcal{CSL}$ interpreted in various distance spaces, in particular, metric spaces. We show that the satisfiability (or consistency) problem for either logic is ExpTime-complete over standard distance spaces, even under the binary coding of the parameters in absolute distance operators of the logic $\mathcal{SL}$. We also prove the finite model property of the logics under consideration. The ExpTime upper bound proofs depend heavily on whether the underlying distance spaces are assumed to be symmetric or not: in the non-symmetric case the upper bound is established by an exponential reduction to the emptiness problem for finite looping tree automata, whereas in the symmetric case it is proved by an exponential reduction to the emptiness problem for single complemented pair automata. (Note that the complexity of the logic $\mathcal{MS}$ of 'absolute' similarity has been determined in [38].)

2. A modular reasoning procedure for the combination $\mathcal{SL}+\mathcal{ALCQIO}$ of $\mathcal{SL}$ and $\mathcal{ALCQIO}$ is developed. Reasoning in this logic is shown to be NExpTime-complete.

2 Similarity description logic

We begin by introducing the syntax and semantics of our logic. The language can be based on any of the standard DLs with the nominal constructor, say, the very expressive DL $\mathcal{SHOIQ}$ [16] which is the core of the ontology language OWL-DL. For simplicity, in this paper we consider the DL $\mathcal{ALCQIO}$, that is, $\mathcal{SHOIQ}$ without transitive roles and role hierarchies [2].

Language. Let $\mathsf{N_C}$ be a countable set of *concept names*, $\mathsf{N_R}$ a countable set of *role names*, $\mathsf{N_O}$ a countable set of *object names*, and S a finite set of *names for similarity spaces*. The set R of *roles* is $\mathsf{N_R} \cup \{R^{-1} \mid R \in \mathsf{N_R}\}$.

The *concepts* of the '*similarity + description*' language $\mathcal{SL} + \mathcal{ALCQIO}$ are defined inductively as follows:

$$C, D \ ::= \ A \quad | \quad \{\ell\} \quad | \quad C \sqcap D \quad | \quad \neg C \quad | \quad \leq nR.C \quad | \quad \geq nR.C$$
$$C \leftleftarrows_s D \quad | \quad \exists_s^{<a} C \quad | \quad \exists_s^{\leq a} C \tag{1}$$

where A ranges over $\mathsf{N_C}$, R over R, ℓ over $\mathsf{N_O}$, n over the natural numbers $\mathbb{N}$, a over the positive rational numbers $\mathbb{Q}^{>0}$, and s over the names for similarity spaces from S. As usual, $\exists R.C$ stands for $\geq 1R.C$. We also use $\sqcup$, $\bot$ (for $\emptyset$), and $\top$ (for the whole space) as the standard Boolean abbreviations.

A *knowledge base* (*KB*, for short) in this language is a triple of the form $\Sigma = (\mathcal{T}^\Sigma, \mathcal{A}^\Sigma, \mathcal{D}^\Sigma)$, where

- $\mathcal{T}^\Sigma$ is a finite set (called a *TBox*) of *subsumption relations* $C \sqsubseteq D$ between concepts C and D of $\mathcal{SL} + \mathcal{ALCQIO}$,
- $\mathcal{A}^\Sigma$ is a finite set (called an *ABox*) of assertions of the form

  $$\ell_1 : C, \qquad \ell_1 R \ell_2, \qquad d_s(\ell_1, \ell_2) < a, \qquad d_s(\ell_1, \ell_2) = a,$$

  where C is a concept, R a role, $s \in \mathsf{S}$, the ℓ_i are object names, d_s is the distance (or similarity) measure on the distance (similarity) space s (see below), and $a \in \mathbb{Q}^{>0}$,
- $\mathcal{D}^\Sigma$ associates with each $s \in \mathsf{S}$ a pair $(\mathcal{D}_1(s), \mathcal{D}_2(s))$, where $\mathcal{D}_1(s)$ is a concept name from $\mathsf{N_C}$ and $\mathcal{D}_2(s) \subseteq \{\mathsf{sym}, \mathsf{tr}\}$ (to indicate whether the similarity space on $\mathcal{D}_1(s)$ is symmetric and/or satisfies the triangle inequality; see below for definitions).

Models. To introduce models interpreting such knowledge bases, we need a definition of a distance (or similarity) space. Strictly speaking, the notions of distance and similarity are *dual*, but it will be more convenient for us not to distinguish between them here.

A *distance space* is a structure of the form $\mathfrak{D} = (D, d)$, where D is a nonempty set and d, the *distance function* (or *similarity measure*), is a map from $D \times D$ to the set $\mathbb{R}^{\geq 0}$ of nonnegative real numbers such that, for all $x, y \in D$, we have $d(x, y) = 0$ iff $x = y$. If d satisfies two additional well-known properties

$$d(x, y) \ = \ d(y, x) \tag{\textbf{sym}}$$
$$d(x, z) \ \leq \ d(x, y) + d(y, z) \tag{\textbf{tr}}$$

then $\mathfrak{D}$ is a standard *metric space*. The *distance* $d(X, Y)$ between two nonempty sets X and Y of D is defined by

$$d(X, Y) \ = \ \inf\{d(x, y) \mid x \in X, \ y \in Y\}.$$

If at least one of X, Y is empty then $d(X,Y) = \infty$. Finally, if we actually have

$$d(X,Y) \;=\; \min\{d(x,y) \mid x \in X,\ y \in Y\} \qquad \textbf{(min)}$$

for any nonempty $X, Y \subseteq D$, then $\mathfrak{D}$ is called a *min-space*. Every finite distance space is obviously a min-space. From now on we will only work with min-spaces. This means, in particular, that the topology induced by the distance spaces under consideration is trivial: every set of the space is both closed and open. Such a restriction seems to be quite natural for the intended applications, since most similarity spaces are either finite or can be approximated by min-spaces (unlike those in spatial and temporal KR&R). It is of interest to observe that a similar restriction is required in the semantics of conditional logics (where it is called the *limit assumption*). The condition (min) and the relation between the logics considered here and conditional logics will be discussed in more detail in Section 5.

Now, a Σ-*type model* (or *interpretation*) $\mathcal{I}$ consists of a nonempty domain $\Delta^{\mathcal{I}}$ and interpretations

- $\ell^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, for each $\ell \in \mathsf{N_O}$,
- $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, for each $A \in \mathsf{N_C}$,
- $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, for each $R \in \mathsf{N_R}$,
- $(R^{-1})^{\mathcal{I}} = (R^{\mathcal{I}})^{-1}$, for each $R \in \mathsf{N_R}$,
- a min-space $s^{\mathcal{I}} = (A_s^{\mathcal{I}}, d_s^{\mathcal{I}})$ such that $\mathcal{D}_1(s) = A_s \in \mathsf{N_C}$, $A_s^{\mathcal{I}} \neq \emptyset$, and $s^{\mathcal{I}}$ satisfies the constraints in $\mathcal{D}_2(s)$, for each $s \in \mathsf{S}$.

Given a Σ-type model $\mathcal{I}$, we define inductively the *extension* $C^{\mathcal{I}}$ of a concept C:

$$\begin{aligned}
\{\ell\}^{\mathcal{I}} &= \{\ell^{\mathcal{I}}\}\\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}\\
(\leq nR.C)^{\mathcal{I}} &= \big\{x \in \Delta^{\mathcal{I}} \;\big|\; |\{y \in C^{\mathcal{I}} \mid xR^{\mathcal{I}}y\}| \leq n\big\}\\
(\geq nR.C)^{\mathcal{I}} &= \big\{x \in \Delta^{\mathcal{I}} \;\big|\; |\{y \in C^{\mathcal{I}} \mid xR^{\mathcal{I}}y\}| \geq n\big\}\\
(C_1 \leftleftarrows_s C_2)^{\mathcal{I}} &= \{x \in A_s^{\mathcal{I}} \mid d_s^{\mathcal{I}}(x, C_1^{\mathcal{I}} \cap A_s^{\mathcal{I}}) < d_s^{\mathcal{I}}(x, C_2^{\mathcal{I}} \cap A_s^{\mathcal{I}})\}\\
(\exists_s^{<a} C)^{\mathcal{I}} &= \{x \in A_s^{\mathcal{I}} \mid \exists y \in (A_s^{\mathcal{I}} \cap C^{\mathcal{I}})\ d_s^{\mathcal{I}}(x,y) < a\}\\
(\exists_s^{\leq a} C)^{\mathcal{I}} &= \{x \in A_s^{\mathcal{I}} \mid \exists y \in (A_s^{\mathcal{I}} \cap C^{\mathcal{I}})\ d_s^{\mathcal{I}}(x,y) \leq a\}
\end{aligned}$$

Finally, a Σ-type model $\mathcal{I}$ *satisfies* a concept inclusion $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. It satisfies $\ell : C$ iff $\ell^{\mathcal{I}} \in C^{\mathcal{I}}$, $\ell_1 R \ell_2$ iff $\ell_1^{\mathcal{I}} R^{\mathcal{I}} \ell_2^{\mathcal{I}}$, $d_s(\ell_1, \ell_2) < a$ iff $d_s^{\mathcal{I}}(\ell_1^{\mathcal{I}}, \ell_2^{\mathcal{I}}) < a$, and $d_s(\ell_1, \ell_2) = a$ iff $d_s^{\mathcal{I}}(\ell_1^{\mathcal{I}}, \ell_2^{\mathcal{I}}) = a$. A Σ-type model $\mathcal{I}$ *satisfies* Σ if is satisfies all concept inclusions and assertions in Σ.

Reasoning problems. The first important reasoning problem when dealing with a given KB Σ is its *consistency*, that is, the question:

- Is there a Σ-type model satisfying Σ?

Other standard reasoning tasks include:

- *Concept satisfiability* relative to a KB: given a concept C and a KB Σ, does there exist a Σ-type model $\mathcal{I}$ satisfying Σ and such that $C^{\mathcal{I}} \neq \emptyset$?

- *Subsumption* relative to a KB: given a concept inclusion $C \sqsubseteq D$ and a KB Σ, is it the case that every model $\mathcal{I}$ satisfying Σ also satisfies $C \sqsubseteq D$?

- *Instance checking* relative to a KB: given a concept C, an object name ℓ, and a KB Σ, is it the case that every model $\mathcal{I}$ satisfying Σ satisfies $\ell : C$?

All these tasks are reducible to each other. For example, there exists a model $\mathcal{I}$ satisfying Σ where $C^{\mathcal{I}} \neq \emptyset$ iff $\Sigma \cup \{\ell : C\}$ is consistent, for some fresh object name ℓ. Or, every model satisfying Σ satisfies $\ell : C$ as well iff $\Sigma \cup \{\ell : \neg C\}$ is not consistent.

Component logics. The logic $\mathcal{SL} + \mathcal{ALCQIO}$ consists of two components. The first line of (1) defines $\mathcal{ALCQIO}$ concepts. In every $\mathcal{ALCQIO}$ knowledge base $\Sigma = (\mathcal{T}^{\Sigma}, \mathcal{A}^{\Sigma}, \mathcal{D}^{\Sigma})$ the third component is empty and the first two are restricted to the language of $\mathcal{ALCQIO}$.

The other component logic is the new *similarity logic* $\mathcal{SL}$. Its concepts are constructed using nominals $\{\ell\}$, the Booleans $\sqcap$, $\neg$, and the similarity operators $\leftleftarrows$, $\exists^{<a}$, and $\exists^{\leq a}$. In fact, $\mathcal{SL} + \mathcal{ALCQIO}$ contains as many 'copies' of this logic as the cardinality of S (the number of similarity spaces) prescribes (we then use the subscript s to distinguish between them).

An $\mathcal{SL}$-*model* $\mathcal{I}$ is a distance space $(\Delta^{\mathcal{I}}, d^{\mathcal{I}})$ together with interpretations $\ell^{\mathcal{I}}$ and $A^{\mathcal{I}}$ of the object and concept names. The extension $C^{\mathcal{I}}$ of an $\mathcal{SL}$-concept C is computed as defined above. An $\mathcal{SL}$ knowledge base $\Sigma = (\mathcal{T}^{\Sigma}, \mathcal{A}^{\Sigma}, \mathcal{D}^{\Sigma})$ is restricted to the language of $\mathcal{SL}$ and $\mathcal{D}^{\Sigma}$ contains just a single similarity space.

3 Decidability and complexity of $\mathcal{SL} + \mathcal{ALCQIO}$

The first main result of this paper is the following

Theorem 1. *The consistency and concept satisfiability problems formulated above for $\mathcal{SL} + \mathcal{ALCQIO}$ KBs Σ are NExpTime-complete. The subsumption and instance checking problems (relative to knowledge bases) are coNExpTime-complete. These complexity results hold for the binary coding of the numerical parameters in both number restrictions and similarity operators.*

The NExpTime lower bound follows from the fact that already consistency of $\mathcal{ALCQIO}$ knowledge bases is NExpTime-hard [33]. In what follows we, therefore, concentrate on the corresponding upper bound.

An almost obvious idea of how to prove this theorem would be to try to decompose $\mathcal{SL} + \mathcal{ALCQIO}$ into the *fusion* [4] of its component logics. Indeed, one can polynomially reduce the reasoning problems for $\mathcal{SL}+\mathcal{ALCQIO}$ to the corresponding problems for the fusion of the description logic $\mathcal{ALCQIO}$ and the similarity logics $\mathcal{SL}_s$, $s \in \mathsf{S}$. But unfortunately such a reduction does not help to determine the computational complexity of the reasoning tasks. What is more important is that the reduction does not even provide us with a decidability proof. The reason is that all known transfer results for fusions, with the exception of [11], rely upon the assumption that the component languages *do not* contain nominals. In the language we consider here, however, the presence of nominals is crucial (to model, for example, prototypical reasoning). The transfer result of [11] does apply to logics with nominals. Yet, we cannot use it for our logic because it makes the strong assumption that there are no 'hidden' nominals in the components in the sense that they must not be able to express, using a TBox, that certain concepts C have to be interpreted as singleton sets unless there actually exists an object name ℓ such that $\{\ell\} = C$ follows from the TBox (this property is called *nominal-closedness* in [11]). Clearly, the logic $\mathcal{ALCQIO}$ does not have this property.

In spite of these difficulties, we show in the remaining part of this section that a NExpTime decision procedure for $\mathcal{SL} + \mathcal{ALCQIO}$ can be obtained as a modular combination of an existing decision procedure for $\mathcal{ALCQIO}$ (actually, we take a decision procedure for C^2, the two-variable fragment of first-order logic with counting [26]) and a decision procedure for $\mathcal{SL}$ developed in Section 4.

Before turning on to technical details, we observe that without loss of generality we can always assume that the ABox of a given KB is empty. Indeed, the assertion $\ell : C$ is equivalent to $\{\ell\} \sqsubseteq C$, $\ell_1 R \ell_2$ is equivalent to $\{\ell_1\} \sqsubseteq \exists R.\{\ell_2\}$, $d_s(\ell_1, \ell_2) < a$ is equivalent to $\{\ell_1\} \sqsubseteq \exists_s^{<a}\{\ell_2\}$, and $d_s(\ell_1, \ell_2) = a$ is equivalent to $\{\ell_1\} \sqsubseteq \exists_s^{\leq a}\{\ell_2\}$, $\{\ell_1\} \sqsubseteq \neg\exists_s^{<a}\{\ell_2\}$. In what follows we always assume that our knowledge bases have empty ABoxes.

The decision procedure for $\mathcal{SL}+\mathcal{ALCQIO}$. We now show how the consistency problem for an arbitrary KB Σ can be split into the consistency problems for $\mathcal{SL}$ and $\mathcal{ALCQIO}$ knowledge bases. Let $\mathrm{P}(\Sigma)$ be the set of all rational numbers (we will call them *parameters*) occurring in the similarity operators ($\exists_s^{<a}$ and $\exists_s^{\leq a}$) from Σ, and let

$$\mathrm{P}^+(\Sigma) \;=\; \{b = a_1 + \cdots + a_n \mid a_i \in \mathrm{P}(\Sigma) \;\&\; b < \max \mathrm{P}(\Sigma)\}.$$

Let $\exists_s D = D \leftleftarrows_s \bot$. Note that $\exists_s$ behaves like the *existential modality* on the spaces $A_s^{\mathcal{I}}$ [12], that is, $(\exists_s D)^{\mathcal{I}} = A_s^{\mathcal{I}}$ if $D^{\mathcal{I}} \cap A_s^{\mathcal{I}} \neq \emptyset$, and $(\exists_s D)^{\mathcal{I}} = \emptyset$ otherwise.

Denote by $\mathrm{cl}(\Sigma)$ the smallest set of concepts containing all concepts from Σ and closed under the following rules:

- $\mathrm{cl}(\Sigma)$ is closed under taking subconcepts and single negation,
- if $D_1 \leftleftarrows_s D_2 \in \mathrm{cl}(\Sigma)$ then $\exists_s D_1, \exists_s D_2 \in \mathrm{cl}(\Sigma)$,
- if $\exists_s^{<a} D \in \mathrm{cl}(\Sigma)$ or $\exists_s^{\leq a} D \in \mathrm{cl}(\Sigma)$ then $\exists_s D \in \mathrm{cl}(\Sigma)$,
- if $\exists_s D \in \mathrm{cl}(\Sigma)$ and $a \in \mathrm{P}^+(\Sigma)$ then $\exists_s^{<a} D, \exists_s^{\leq a} D \in \mathrm{cl}(\Sigma)$.

It should be clear that the size of $\mathrm{cl}(\Sigma)$ is at most exponential in the *length* $l(\Sigma)$ of Σ if the parameters are coded in *binary*.

A very important notion in our construction is that of a type over Σ. Namely, a *type over* Σ is a subset T of $\mathrm{cl}(\Sigma)$ such that the following conditions are satisfied:

- $D_1 \sqcap D_2 \in T$ iff $D_1, D_2 \in T$, for every $D_1 \sqcap D_2 \in \mathrm{cl}(\Sigma)$,
- $\neg D \in T$ iff $D \notin T$, for every $\neg D \in \mathrm{cl}(\Sigma)$,
- $\exists_s^{<a} D \in T$ implies $\exists_s^{<b} D, \exists_s^{\leq b} D, \exists_s D \in T$, for every $\exists_s D \in \mathrm{cl}(\Sigma)$ and all $b \in \mathrm{P}^+(\Sigma)$, $b \geq a$,
- $\exists_s^{\leq a} D \in T$ implies $\exists_s^{<b} D, \exists_s^{\leq b} D, \exists_s D \in T$, for every $\exists_s D \in \mathrm{cl}(\Sigma)$ and all $b \in \mathrm{P}^+(\Sigma)$, $b > a$,
- if $D_1 \leftleftarrows_s D_2 \in T$ and $\exists_s^{\leq a} D_2 \in T$, then $\exists_s^{<a} D_1 \in T$,
- $\exists_s D \in T$ for some D, then $A_s \in T$.

It is to be noted that although the size of $\mathrm{cl}(\Sigma)$ can be exponential in $l(\Sigma)$ (due to taking the closure under addition when defining $\mathrm{P}^+(\Sigma)$), we still have at most exponentially many distinct types over Σ because whenever $\exists_s^{<a} D \in T$, then $\exists_s^{<b} D \in T$ for *all* $b \geq a$, $b \in \mathrm{P}^+(\Sigma)$ (and similarly for $\exists_s^{\leq a} D$).

Given an interpretation $\mathcal{I}$ and an element u of its domain, define the *type* $T^{\mathcal{I}}(u)$ *of* u *in* $\mathcal{I}$ by taking

$$T^{\mathcal{I}}(u) \;=\; \{C \in \mathrm{cl}(\Sigma) \mid u \in C^{\mathcal{I}}\}.$$

It is easily seen that this is indeed a type over Σ. We remind the reader that types of a similar form are basic ingredients of various worst-case optimal algorithms in description, modal, and temporal logics [35, 25]. In the definition above, however, we had to be particularly careful in order to avoid creating double-exponentially many types.

To design a modular consistency checking algorithm for $\mathcal{SL} + \mathcal{ALCQIO}$ knowledge bases, we define 'abstractions' of concepts that will be used as inputs for the component algorithms. For each C of the form $\exists_s^{<a} D$, $\exists_s^{\leq a} D$, $D_1 \leftleftarrows_s D_2$, $\leq nR.D$ and $\geq nR.D$ we reserve a fresh concept name A^C, the *surrogate of* C. Now, given an arbitrary concept C, denote by $C^{\mathcal{DL}}$ the result of replacing in C all outermost occurrences of subconcepts of the form $\exists_s^{<a} D$, $\exists_s^{\leq a} D$, $D_1 \leftleftarrows_s D_2$ with their surrogates. Let $\mathrm{cl}(\Sigma)^{\mathcal{DL}}$ (or $T^{\mathcal{DL}}$) be the set of all $C^{\mathcal{DL}}$ such that $C \in \mathrm{cl}(\Sigma)$ (respectively, $C \in T$).

Similarly, for each $s \in \mathsf{S}$ and each concept C, denote by C^s the result of replacing in C all outermost occurrences of subconcepts of the form $\leq nR.D$, $\geq nR.D$, $\exists_{s'}^{<a} D$, $\exists_{s'}^{\leq a} D$, $D_1 \leftleftarrows_{s'} D_2$, for $s' \neq s$, with their surrogates.

Say that a set $\mathcal{T}$ of types over $\mathrm{cl}(\Sigma)^{\mathcal{DL}}$ is $\mathcal{ALCQIO}$*-realisable* if there exists an interpretation $\mathcal{I}$ such that

$$\{T^{\mathcal{I}}(u) \mid u \in \Delta^{\mathcal{I}}\} \;=\; \mathcal{T}. \tag{2}$$

Similarly, we say that a set $\mathcal{T}$ of types over $\mathrm{cl}(\Sigma)^s$ is $\mathcal{SL}$*-realisable* if there exists an $\mathcal{SL}$-model $\mathcal{I}$ with domain $A_s^{\mathcal{I}}$, *where possibly not all nominals are interpreted*, such that (2) holds.

A modular nondeterministic consistency checking procedure for knowledge bases over $\mathcal{SL} + \mathcal{ALCQIO}$ is sketched in Fig. 1.

It is easily seen that if there is no set $\mathcal{T}$ with the properties listed in Fig. 1, then Σ is inconsistent. Indeed, suppose that Σ is consistent. Take a Σ-type model $\mathcal{I}$ satisfying Σ. Define $\mathcal{T}$ by

$$\mathcal{T} \;=\; \{T^{\mathcal{I}}(u) \mid u \in \Delta^{\mathcal{I}}\}.$$

It is straightforward to see that for this set $\mathcal{T}$ the algorithm in Fig. 1 returns the answer YES.

The converse direction as well as the proof that this algorithm runs in NExpTime are based on two theorems. The first one shows that item 3 in Fig. 1 can be checked in NExpTime:

Theorem 2. *It is decidable in NExpTime (in the size of Σ) whether a set $\mathcal{T}$ of types over an $\mathcal{ALCQIO}$ knowledge base Σ is realisable. This holds for the binary coding of the numerical parameters in the qualified number restriction operators.*

Proof. It is well known that $\mathcal{ALCQIO}$ can be regarded as a fragment of C^2, the *two-variable fragment of first-order logic with counting* [2, page 152]. A recent result of Pratt-Hartmann [26] shows that the satisfiability problem for C^2-formulas is NExpTime-complete under the binary coding of number restrictions. We cannot apply this result directly to the problem stated in the theorem because the C^2-formula encoding the realisability of $\mathcal{T}$ is already exponential in the size $l(\Sigma)$ of Σ. However, Pratt-Hartmann's proof is type-based in the sense

Given a knowledge base Σ (without an ABox),

1. Guess a nonempty set $\mathcal{T}$ of types over Σ.
2. Check whether the following conditions hold:
   - if $C \sqsubseteq D \in \mathcal{T}^{\Sigma}$ then $\neg C \in T$ or $D \in T$, for all $T \in \mathcal{T}$,
   - if $\exists_s D \in T$ for some $T \in \mathcal{T}$, then $\exists_s D \in T'$ for all $T' \in \mathcal{T}$ such that $A_s \in T'$.
3. Check whether the set $\{T^{\mathcal{DL}} \mid T \in \mathcal{T}\}$ of types over $\mathrm{cl}(\Sigma)^{\mathcal{DL}}$ is $\mathcal{ALCQIO}$-realisable.
4. For each $s \in \mathsf{S}$, consider the set of types

$$\mathcal{T}^s \;=\; \{T^s \mid A_s \in T,\; T \in \mathcal{T}\}$$

   over $\mathrm{cl}(\Sigma)^s$ and check whether the set $\mathcal{T}^s$ is $\mathcal{SL}$-realisable in an interpretation satisfying the constraints in $\mathcal{D}_2(s)$.
5. If the answer is YES for at least one guessed set $\mathcal{T}$ of types over Σ, then Σ is consistent; otherwise it is inconsistent.

Figure 1: Decision procedure.

that in order to decide (in NExpTime) whether a C^2-formula φ is satisfiable, it guesses a set of types over the formula φ (together with some additional information about the types) and then checks whether this set of types is realisable (again together with a number of additional constraints). A close inspection of Pratt-Hartmann's procedure shows that it still runs in NExpTime if the input is not a formula but a set of types whose realisability is to be checked. $\square$

We now formulate the second main result of this paper which gives an ExpTime decision procedure for the similarity logic $\mathcal{SL}$. It also establishes in (ii) that item 4 in Fig. 1 can be checked in ExpTime. Under (iii) it lists the property required to show the correctness of our algorithm.

Theorem 3. (i) *The consistency problem for $\mathcal{SL}$ knowledge bases is ExpTime-complete for the binary coding of the numerical parameters in the similarity operators.*

(ii) *It is decidable in ExpTime (in the size of Σ) whether a set $\mathcal{T}$ of types over an $\mathcal{SL}$ knowledge base Σ is realisable in a model satisfying the given constraints from $\{\mathsf{sym}, \mathsf{tr}\}$. This holds for the binary coding of the numerical parameters in the similarity operators.*

(iii) *Moreover, let $T_1, \dots, T_n$ be an enumeration of all those types from $\mathcal{T}$ that do not contain nominals, and let $k_1, \dots, k_n$ be a sequence of some positive numbers from $\mathbb{N} \cup \{\omega\}$. If $\mathcal{T}$ is realisable then it is realisable in a model where the set of points realising T_i is of cardinality k_i, for $i = 1, \dots n$.*

A proof of this theorem will be given in Section 4.

Let us see now that the algorithm in Fig. 1 is correct. Suppose that the procedure returns 'YES' for some set $\mathcal{T}$ of types over Σ.

Take an interpretation $\mathcal{I}$ which realises $\{T^{\mathcal{DL}} \mid T \in \mathcal{T}\}$. As $\mathcal{ALCQIO}$ is a fragment of first-order logic, we may assume that $\Delta^{\mathcal{I}}$ is countable. By Theorem 3 (iii), for each $s \in \mathsf{S}$, there exists an interpretation $\mathcal{I}_s$ with $\Delta^{\mathcal{I}_s} = A_s^{\mathcal{I}_s}$ which realises $\mathcal{T}^s = \{T^s \mid A_s \in T,\; T \in \mathcal{T}\}$

in such a way that, for each $T \in \mathcal{T}$ with $A_s \in T$,

$$|\bigcap_{C \in T^s} C^{\mathcal{I}_s}| \;=\; |\bigcap_{C \in T^{\mathcal{DL}}} C^{\mathcal{I}}|.$$

For every pair (s, T) with $s \in \mathsf{S}$ and $T \in \mathcal{T}^s$, take a bijection $f_{s,T}$ from the set in the left-hand side of this equality to the set in the right-hand side. Let

$$f_s \;=\; \bigcup_{T \in \mathcal{T}^s} f_{s,T}.$$

Then f_s is a bijection from $\Delta^{\mathcal{I}_s}$ onto $A_s^{\mathcal{I}}$. Now define a new model $\mathcal{I}'$ with domain $\Delta^{\mathcal{I}}$ by taking the interpretation of atomic concepts and role names from $\mathcal{I}$ and defining, for all $s \in \mathsf{S}$ and $x, y \in A_s^{\mathcal{I}}$, $d_s^{\mathcal{I}'}(x, y) \;=\; d_s^{\mathcal{I}_s}(f_s^{-1}(x), f_s^{-1}(y))$. One can prove by induction that, for all $C \in cl(\Sigma)$ and $s \in \mathsf{S}$, we have

$$C^{\mathcal{I}'} \cap A_s^{\mathcal{I}'} \;=\; (C^{\mathcal{DL}})^{\mathcal{I}} \cap A_s^{\mathcal{I}} \;=\; \{f_s(x) \mid x \in (C^s)^{\mathcal{I}_s}\}.$$

First note that the equalities

$$(C^{\mathcal{DL}})^{\mathcal{I}} \cap A_s^{\mathcal{I}} \;=\; \{f_s(x) \mid x \in (C^s)^{\mathcal{I}_s}\}$$

follow (without any inductive argument) from the definition of f_s. For concept names the two equalities follow immediately from the definitions. For the proof by induction it is therefore sufficient to show that

$$C^{\mathcal{I}'} \cap A_s^{\mathcal{I}'} \;=\; (C^{\mathcal{DL}})^{\mathcal{I}} \cap A_s^{\mathcal{I}}$$

if C starts with a description logic operator, and

$$C^{\mathcal{I}'} \cap A_s^{\mathcal{I}'} \;=\; \{f_s(x) \mid x \in (C^s)^{\mathcal{I}_s}\}$$

if C starts with an $\mathcal{SL}$-operator. As an example of the inductive step we consider $C = \exists_{s'}^{<a} D$. Suppose that the equalities above hold for D and all elements of S. Let $s' \in \mathsf{S}$. First assume $s = s'$. Let $u \in (\exists_s^{<a} D)^{\mathcal{I}'} \cap A_s^{\mathcal{I}'}$. This is the case iff there exists $v \in A_s^{\mathcal{I}'}$ such that $d_s^{\mathcal{I}'}(u, v) < a$ and $v \in D^{\mathcal{I}'}$. By the definitions of f_s, $d_s^{\mathcal{I}'}$ and the induction hypothesis, this is the case iff there exists $v \in A_s^{\mathcal{I}'}$ such that $d_s^{\mathcal{I}_s}(f_s^{-1}(u), f_s^{-1}(v)) < a$ and $f_s^{-1}(v) \in (D^s)^{\mathcal{I}_s}$. But this is equivalent to $f_s^{-1}(u) \in (\exists_s^{<a} D^s)^{\mathcal{I}_s}$. For $s \neq s'$ the equality follows immediately from the case $s = s'$.

Using these equalities, one can show that $\mathcal{I}'$ satisfies Σ: every subsumption $C \sqsubseteq D \in \mathcal{T}^{\Sigma}$ is true in $\mathcal{I}'$ because (by the first condition of item 2 in Fig. 1) $\neg C \in T$ or $D \in T$ for every $T \in \mathcal{T}$, and therefore $C^{\mathcal{I}'} \subseteq D^{\mathcal{I}'}$.

4 The similarity logic $\mathcal{SL}$

In this section we prove Theorem 3. Recall that in $\mathcal{SL}$ we do not have qualified number restrictions. Moreover, we deal with only one similarity space, and so can omit indexes from the distance operators and write simply $\exists^{<a}$, $\exists^{\leq a}$, $\leftleftarrows$.

An $\mathcal{SL}$-model $\mathcal{I}$ was defined as a min-space $(\Delta^{\mathcal{I}}, d^{\mathcal{I}})$ together with interpretations $\ell^{\mathcal{I}}$ and $A^{\mathcal{I}}$ of the object and concept names. We are going to prove Theorem 3 for two types of $\mathcal{SL}$ knowledge bases Σ:

- the *non-symmetric case*: $\mathcal{D}^\Sigma$ states that the min-spaces to be considered satisfy the triangle inequality only,
- the *symmetric case*: $\mathcal{D}^\Sigma$ states that the min-spaces to be considered are actually metric spaces (satisfying both (tr) and (sym)).

The remaining two cases are much simpler in the sense that one can easily obtain proofs for them by a straightforward modification of the proofs for the cases above. We leave this to the reader.

For all possible types of constraints, the ExpTime lower bound required for Theorem 3 (i) was proved in [30] even for the language without nominals and operators of the form $\exists^{<a}$ and $\exists^{\leq a}$. The proof was by reduction of the global consequence relation for the modal logic **K** which is known to be ExpTime-hard [31]. Another encoding of the global consequence relation for **K** using the operator $\exists^{\leq 1}$ and the Booleans can be found in [38].

In order to ease presentation, in the proof of the ExpTime upper bound we will consider the *satisfiability problem for $\mathcal{SL}$-concepts* instead of the consistency problem for $\mathcal{SL}$-knowledge bases. We call an $\mathcal{SL}$-concept C *satisfiable* in a class $\mathcal{M}$ of $\mathcal{SL}$-models if there exists $\mathcal{I} \in \mathcal{M}$ such that $C^{\mathcal{I}} \neq \emptyset$. To see that consistency of $\mathcal{SL}$-knowledge bases is reducible to satisfiability of $\mathcal{SL}$-concepts, recall first that we may assume that the ABox of Σ is empty. Also recall the definitions of the 'universal modalities:' $\exists D = D \Leftarrow \bot$ and $\forall D = \neg\exists\neg D$. Now, Σ is consistent iff $\forall C$ is satisfiable in a model from the class $\mathcal{M}$ of all $\mathcal{SL}$ models meeting the constraints in $\mathcal{D}^\Sigma$, where

$$C \;=\; \bigsqcap_{D_1 \sqsubseteq D_2 \in \mathcal{T}^\Sigma} (\neg D_1 \sqcup D_2).$$

The statements (ii) and (iii) of Theorem 3 will also be proved for sets of types over a concept C instead of sets of types over a knowledge base Σ. It will become clear that this does not make any essential difference.

The structure of this section is now as follows. We start with a proof of the (bounded) finite model property of concept satisfiability for both non-symmetric and symmetric cases. This result will first be used to prove Theorem 3 (iii). And then it will be employed as a basic ingredient for the proof of the upper bound of (i) for the non-symmetric and symmetric cases. The crucial idea in this proof is that satisfiable $\mathcal{SL}$-concepts can be satisfied in 'abstract tree distance spaces,' and that satisfiability in these tree distance spaces can be reduced (exponentially) to the emptiness problem for tree automata with *one* complemented pair, which is known to be decidable in polynomial time [8]. Claim (ii) will then be a simple corollary of the proof of (i).

4.1 The finite model property

Since here our concern is *$\mathcal{SL}$-concept satisfiability*, we relativise the definitions of the closure, type, etc. to concepts of the language $\mathcal{SL}$.

So suppose an $\mathcal{SL}$-concept C is given. Then $\mathrm{P}(C)$ is the set of all rational numbers (parameters) occurring in C, and

$$\mathrm{P}^+(C) \;=\; \{b = a_1 + \cdots + a_n \mid a_i \in \mathrm{P}(C),\; b < \max \mathrm{P}(C)\}.$$

Denote by $\mathrm{sub}(C)$ the set of all subconcepts of C, and by $\mathrm{cl}(C)$ the smallest set of concepts closed under the following rules:

- $\mathrm{cl}(C)$ contains $\mathrm{sub}(C)$ and is closed under taking single negation,
- if $D_1 \leftleftarrows D_2 \in \mathrm{cl}(C)$ then $\exists D_1, \exists D_2 \in \mathrm{cl}(C)$,
- if $\exists^{<a} D \in \mathrm{cl}(C)$ or $\exists^{\leq a} D \in \mathrm{cl}(C)$ then $\exists D \in \mathrm{cl}(C)$,
- if $\exists D \in \mathrm{cl}(C)$ and $a \in \mathrm{P}^+(C)$ then $\exists^{<a} D, \exists^{\leq a} D \in \mathrm{cl}(C)$.

As we have already observed, the size of $\mathrm{cl}(C)$ is at most exponential in the *length* $l(C)$ of C if the parameters are coded in binary.

If $\mathrm{P}^+(C) = \{a_0, \dots, a_{N-1}\}$ with $a_0 < \dots < a_{N-1}$, then we set

$$a_N \;=\; \min\{a + b \mid a, b \in \mathrm{P}^+(C),\ a + b > a_{N-1}\}, \quad \text{and} \quad \overline{\mathrm{P}}(C) \;=\; \{a_0, \dots, a_N\}. \qquad (3)$$

Given an $\mathcal{SL}$-model $\mathcal{I}$ and its element u, define the *type* $T^{\mathcal{I}}(u)$ of u in $\mathcal{I}$ by taking

$$T^{\mathcal{I}}(u) \;=\; \{D \in \mathrm{cl}(C) \mid u \in D^{\mathcal{I}}\}.$$

Theorem 4. *Let $\mathcal{M}$ be the class of metric min-models or the class of min-models with (tr). If C is satisfiable in $\mathcal{M}$ then C is satisfiable in a finite model from $\mathcal{M}$ (whose size is at most exponential in the length of C).*

Proof. The proof uses a more or less standard filtration argument from modal logic (see, e.g., [5]). The main difficulty will be to define a proper distance function on the filtrated model.

We consider first the case of $\mathcal{M}$ containing arbitrary (not necessarily symmetric) min-models with (tr). In fact, the technique we use below works for a wider class of models where (min) is relaxed to the following condition:

$$d(x, Y) \;=\; \min\{d(x, y) \mid y \in Y\} \qquad (\mathbf{wmin})$$

for every point x and every nonempty set Y. Such models are called *weak min-models*.

Non-symmetric case. Suppose that C is satisfied in a weak min-model $\mathcal{I}$ with (tr). Our aim is to construct a *finite* (not necessarily symmetric) model $\overline{\mathcal{I}}$ with (tr) satisfying C.

Define an equivalence relation $\sim$ on $\Delta^{\mathcal{I}}$ by taking

$$u \sim v \quad \text{iff} \quad T^{\mathcal{I}}(u) = T^{\mathcal{I}}(v), \qquad (4)$$

and let $[u] = \{v \in \Delta^{\mathcal{I}} \mid u \sim v\}$, for every $u \in \Delta$. The model $\overline{\mathcal{I}}$ will be based on the set $\Delta^{\overline{\mathcal{I}}} = \{[u] \mid u \in \Delta^{\mathcal{I}}\}$. To define a suitable distance function on $\Delta^{\overline{\mathcal{I}}}$, some extra work has to be done.

For every $a \in \overline{\mathrm{P}}(C)$, we first introduce two binary relations $R^{<a}$ and $R^{\leq a}$ on $\Delta^{\overline{\mathcal{I}}}$. We have $[u]R^{<a}[v]$ iff the following conditions are satisfied for all $\exists D \in \mathrm{cl}(C)$ and all $b, a{+}b \in \mathrm{P}^+(C)$:

$$\begin{array}{rcl} v \in D^{\mathcal{I}} & \text{implies} & u \in (\exists^{<a} D)^{\mathcal{I}}, \\ v \in (\exists^{\leq b} D)^{\mathcal{I}} & \text{implies} & u \in (\exists^{<a+b} D)^{\mathcal{I}}. \end{array} \qquad (5)$$

Let $d^{<}([u],[v]) = \min\big\{a \in \overline{\mathrm{P}}(C) \,\big|\, [u]R^{<a}[v]\big\}$. It should be clear that these definitions do not depend on the choice of u and v from $[u]$ and $[v]$.

Similarly, $[u]R^{\leq a}[v]$ holds iff the following conditions are satisfied:

$$\begin{array}{rcl} v \in D^{\mathcal{I}} & \text{implies} & u \in (\exists^{\leq a} D)^{\mathcal{I}}, \\ v \in (\exists^{\leq b} D)^{\mathcal{I}} & \text{implies} & u \in (\exists^{\leq a+b} D)^{\mathcal{I}}, \\ v \in (\exists^{< b} D)^{\mathcal{I}} & \text{implies} & u \in (\exists^{< a+b} D)^{\mathcal{I}}, \end{array} \tag{6}$$

and we let $d^{\leq}([u],[v]) = \min\big\{a \in \overline{\mathrm{P}}(C) \,\big|\, [u]R^{\leq a}[v]\big\}$.

Recall from (3) that a_N denotes the maximal element of $\overline{\mathrm{P}}(C)$ and $a_N \notin \mathrm{P}^+(C)$. It follows from the definition above that $R^{<a_N} = R^{\leq a_N} = \Delta^{\overline{\mathcal{I}}} \times \Delta^{\overline{\mathcal{I}}}$.

Let $a \oplus b = \min\{a_N, a+b\}$ for $a, b \in \overline{\mathrm{P}}(C)$. It is straightforward to check the following properties:

Lemma 5. *For all $a, b \in \overline{\mathrm{P}}(C)$ and $u, v \in \Delta^{\mathcal{I}}$, we have:*

(1) $R^{<a} \circ R^{\leq b} \subseteq R^{<a\oplus b}$, $R^{\leq a} \circ R^{<b} \subseteq R^{<a\oplus b}$, $R^{\leq a} \circ R^{\leq b} \subseteq R^{\leq a\oplus b}$,

(2) *if* $d(u,v) < a$ *then* $[u]R^{<a}[v]$ *and* $d^{<}([u],[v]) \leq a$,
if $d(u,v) \leq a$ *then* $[u]R^{\leq a}[v]$ *and* $d^{\leq}([u],[v]) \leq a$,
$d^{\leq}([u],[v]) \leq d^{<}([u],[v])$.

Corollary 6. *The functions $d^{<}$ and $d^{\leq}$ satisfy the triangle inequality. Moreover, for all $u, v, w \in \Delta^{\mathcal{I}}$, we have:*

$$\begin{array}{rcl} d^{<}([w],[u]) & \leq & d^{<}([w],[v]) + d^{\leq}([v],[u]), \\ d^{<}([w],[u]) & \leq & d^{\leq}([w],[v]) + d^{<}([v],[u]). \end{array}$$

We will use $d^{\leq}$ as a first approximation for the distance function on $\Delta^{\overline{\mathcal{I}}}$ we need. Let $\overline{\mathrm{P}}(C) = \{a_0, \dots, a_N\}$, $a_0 < \cdots < a_N$, be as in (3). For every $a_s \in \overline{\mathrm{P}}(C)$, we introduce a set E_s of 'sufficiently small' positive numbers that will serve as 'possible adjusting values.' To ensure that the triangle inequality still holds for the modified distances, we need to fulfil the following requirements:

$$\begin{array}{l} a_r < a_s + a_t \;\text{ implies }\; a_r < a_s - e_s + a_t - e_t, \\ a_r = a_s + a_t \;\text{ implies }\; a_r - e_r \leq a_s - e_s + a_t - e_t, \end{array}$$

for all $r, s, t \leq N$ and $e_i \in E_i$. This can be achieved as follows. Define δ and $\varepsilon_0, \dots, \varepsilon_N$ by taking

$$\begin{array}{l} \delta \;=\; \min\{a+b-c \mid a,b,c \in \overline{\mathrm{P}}(C) \cup \{0\},\; a+b > c\} \\ \delta/4 \;=\; \varepsilon_N, \quad \varepsilon_N/4 \;=\; \varepsilon_{N-1}, \quad \dots, \quad \varepsilon_1/4 \;=\; \varepsilon_0. \end{array} \tag{7}$$

Now, if $K = |\Delta^{\overline{\mathcal{I}}}|$ then we let E_s be any (fixed) subset of the open interval $(\varepsilon_s, 2\varepsilon_s)$ such that $|E_s| = K^2$.

For all $s = 0, \dots, N$, define

$$J_s \;=\; \Big\{([u],[v]) \in \Delta^{\overline{\mathcal{I}}} \times \Delta^{\overline{\mathcal{I}}} \mid d^{\leq}([u],[v]) = d^{<}([u],[v]) = a_s \text{ and } [u] \neq [v]\Big\}. \tag{8}$$

Note that $|J_s| \leq K^2$.

For each $[u] \in \Delta^{\overline{\mathcal{I}}}$, fix some representative $\widehat{u} \in [u]$, and, for each $([w],[v]) \in J_s$, choose a number $\varepsilon([w],[v]) \in E_s$ in such a way that the condition

$$\varepsilon([w],[v]) > \varepsilon([w],[u]) \quad \text{iff} \quad d(\widehat{w},[v]) < d(\widehat{w},[u])$$

holds for all $([w],[v])$ and $([w],[u])$ in J_s.

We are now in a position to define a distance function $d^{\overline{\mathcal{I}}}$ on $\Delta^{\overline{\mathcal{I}}}$. Namely, for $[u],[v] \in \Delta^{\overline{\mathcal{I}}}$, let

- $d^{\overline{\mathcal{I}}}([u],[u]) \;=\; 0,$
- $d^{\overline{\mathcal{I}}}([u],[v]) \;=\; d^{\leq}([u],[v]),$ if $[u] \neq [v]$ and $d^{\leq}([u],[v]) < d^{<}([u],[v])$,
- $d^{\overline{\mathcal{I}}}([u],[v]) \;=\; d^{\leq}([u],[v]) - \varepsilon([u],[v]),$ if $[u] \neq [v]$ and $d^{\leq}([u],[v]) = d^{<}([u],[v])$.

Finally, we set $D^{\overline{\mathcal{I}}} = \{[u] \mid u \in D^{\mathcal{I}}\}$ and $\ell^{\overline{\mathcal{I}}} = \{\ell\}^{\mathcal{I}}$, for every concept name D and every object name ℓ.

Lemma 7. *$\overline{\mathcal{I}}$ is an $\mathcal{SL}$-model satisfying the triangle inequality.*

Proof. We need to show that $d^{\overline{\mathcal{I}}}$ is a distance function satisfying (tr). We have $d^{\overline{\mathcal{I}}}([u],[v]) = 0$ by definition. And if $[u] \neq [v]$ and $d^{\leq}([u],[v]) = a_s$ then $d^{\overline{\mathcal{I}}}([u],[v]) \geq a_s - 2\varepsilon_s > \delta - \delta/2 > 0$.

Now consider $[w],[v],[u] \in \Delta^{\overline{\mathcal{I}}}$. We may assume that $[w],[v],[u]$ are all different, since otherwise the triangle inequality trivially holds. Let $d^{\leq}([w],[v]) = a_s$, $d^{\leq}([v],[u]) = a_t$ and $d^{\leq}([w],[u]) = a_r$. By Corollary 6, $a_r \leq a_s + a_t$. If $a_r < a_s + a_t$ then $a_s + a_t - a_r \geq \delta$, $\varepsilon_s, \varepsilon_t \leq \delta/4$, and therefore

$$d^{\overline{\mathcal{I}}}([w],[v]) + d^{\overline{\mathcal{I}}}([v],[u]) - d^{\overline{\mathcal{I}}}([w],[u]) \;>\; a_s + a_t - a_r - 2\varepsilon_s - 2\varepsilon_t \;\geq\; 0.$$

Suppose now that $a_r = a_s + a_t$. Then $a_s, a_t < a_r$, and hence $\varepsilon_s, \varepsilon_t \leq \varepsilon_r/4$. Therefore, if $d^{<}([w],[u]) = a_r$, then

$$d^{\overline{\mathcal{I}}}([w],[v]) + d^{\overline{\mathcal{I}}}([v],[u]) - d^{\overline{\mathcal{I}}}([w],[u]) \;>\; -2\varepsilon_s - 2\varepsilon_t + \varepsilon_r \;\geq\; 0.$$

Thus, it remains to consider the case $d^{<}([w],[u]) > a_r$. But then, by Corollary 6, we have

$$\begin{aligned}
d^{<}([w],[v]) &\geq d^{<}([w],[u]) - d^{\leq}([v],[u]) > a_r - a_t = a_s,\\
d^{<}([v],[u]) &\geq d^{<}([w],[u]) - d^{\leq}([w],[v]) > a_r - a_s = a_t.
\end{aligned}$$

Therefore, $d^{\overline{\mathcal{I}}}([w],[u]) = a_r$, $d^{\overline{\mathcal{I}}}([w],[v]) = a_s$, and $d^{\overline{\mathcal{I}}}([v],[u]) = a_t$, i.e., the triangle inequality holds. ∎

Thus, to complete the proof of Theorem 4 for the non-symmetric case, it is enough to show the following:

Lemma 8. *For all $w \in \Delta^{\mathcal{I}}$ and $D \in \mathrm{cl}(C)$, we have $[w] \in D^{\overline{\mathcal{I}}}$ iff $w \in D^{\mathcal{I}}$.*

Proof. The proof is by induction on the construction of D. The basis of induction and the Boolean cases are trivial.

Case $D = D_0 \leftleftarrows D_1$. We can assume that $D_0^{\mathcal{I}}, D_1^{\mathcal{I}} \neq \emptyset$, since otherwise the claim follows easily from the induction hypothesis. Let $w \in (D_0 \leftleftarrows D_1)^{\mathcal{I}}$. Then $d^{\mathcal{I}}(\widehat{w}, D_0^{\mathcal{I}}) < d^{\mathcal{I}}(\widehat{w}, D_1^{\mathcal{I}})$ where $\widehat{w}$ is the fixed representative of $[w]$. There exist $v_0 \in D_0^{\mathcal{I}}$ and $[v_1] \in D_1^{\overline{\mathcal{I}}}$ such that $d^{\mathcal{I}}(\widehat{w}, v_0) = d^{\mathcal{I}}(\widehat{w}, D_0^{\mathcal{I}})$ and $d^{\overline{\mathcal{I}}}([w],[v_1]) = d^{\overline{\mathcal{I}}}([w], D_1^{\overline{\mathcal{I}}})$ (note that by the induction hypothesis we also have $[v_0] \in D_0^{\overline{\mathcal{I}}}$ and $v_1 \in D_1^{\mathcal{I}}$). We thus need to show that $d^{\overline{\mathcal{I}}}([w],[v_0]) < d^{\overline{\mathcal{I}}}([w],[v_1])$. First, note that $d^{\mathcal{I}}(\widehat{w},[v_0]) = d^{\mathcal{I}}(\widehat{w}, D_0^{\mathcal{I}}) < d^{\mathcal{I}}(\widehat{w}, D_1^{\mathcal{I}}) \leq d^{\mathcal{I}}(\widehat{w},[v_1])$. Next, we prove that $d^{<}([w],[v_0]) \leq d^{\leq}([w],[v_1])$. Let $d^{\leq}([w],[v_1]) = a$, so we may assume that $a < a_N$, because otherwise the statement is trivial. We have $[w] \in R^{\leq a}[v_1]$ and $\exists^{<a} D_0, \exists^{\leq a} D_1 \in \mathrm{cl}(C)$, since $D_0 \leftleftarrows D_1 \in \mathrm{cl}(C)$. Therefore $v_1 \in D_1^{\mathcal{I}}$ implies $\widehat{w} \in (\exists^{\leq a} D_1)^{\mathcal{I}}$, whence $\widehat{w} \in (D_0 \leftleftarrows D_1)^{\mathcal{I}}$ implies $\widehat{w} \in (\exists^{<a} D_0)^{\mathcal{I}}$. Thus $d^{\mathcal{I}}(\widehat{w}, v_0) \leq d^{\mathcal{I}}(\widehat{w}, D_0^{\mathcal{I}}) < a$, and, by Lemma 5 (2), $d^{<}([w],[v_0]) \leq a$, as required.

We now have three subcases. If $d^{\leq}([w],[v_1]) < d^{<}([w],[v_1])$, then by definition we have $d^{\overline{\mathcal{I}}}([w],[v_0]) < d^{<}([w],[v_0]) < d^{\leq}([w],[v_1]) = d^{\overline{\mathcal{I}}}([w],[v_1])$. If $d^{\leq}([w],[v_0]) < d^{<}([w],[v_0])$ then $d^{\leq}([w],[v_1]) - d^{\leq}([w],[v_0]) \geq \delta$, $\varepsilon([w],[v_1]) < \delta/2$, and therefore

$$d^{\overline{\mathcal{I}}}([w],[v_1]) - d^{\overline{\mathcal{I}}}([w],[v_0]) \;\geq\; d^{\leq}([w],[v_1]) - \varepsilon([w],[v_1]) - d^{\leq}([w],[v_0]) \;>\; 0.$$

Finally, if $d^{\leq}([w],[v_0]) = d^{<}([w],[v_0])$ and $d^{\leq}([w],[v]) = d^{<}([w],[v])$ then

$$d^{\overline{\mathcal{I}}}([w],[v_1]) - d^{\overline{\mathcal{I}}}([w],[v_0]) \;=\; -\,\varepsilon([w],[v_1]) + \varepsilon([w],[v_0]) \;>\; 0,$$

since $d^{\mathcal{I}}(\widehat{w},[v_0]) < d^{\mathcal{I}}(\widehat{w},[v_1])$.

In the same way one can show that $w \notin (D_0 \leftleftarrows D_1)^{\mathcal{I}}$ implies $d^{\overline{\mathcal{I}}}([w], D_1^{\overline{\mathcal{I}}}) \leq d^{\overline{\mathcal{I}}}([w], D_0^{\overline{\mathcal{I}}})$.

Case $D = \exists^{<a} D_0$. Let $w \in (\exists^{<a} D_0)^{\mathcal{I}}$, i.e., there is $v \in D_0^{\mathcal{I}}$ with $d^{\mathcal{I}}(w,v) < a$. Then $d^{\overline{\mathcal{I}}}([w],[v]) < d^{<}([w],[v]) \leq a$ in view of Lemma 5 (2), and $[v] \in D_0^{\overline{\mathcal{I}}}$ by the induction hypothesis.

Conversely, suppose that $[v] \in D_0^{\overline{\mathcal{I}}}$ for some $[v]$ with $d^{\overline{\mathcal{I}}}([w],[v]) < a$. Then $d^{\leq}([w],[v]) \leq a$, and $v \in D_0^{\mathcal{I}}$ by the induction hypothesis. If $d^{\leq}([w],[v]) = d^{<}([w],[v])$, then $d^{<}([w],[v]) \leq a$, $wR^{<a}v$ and, therefore, $w \in (\exists^{<a} D_0)^{\mathcal{I}}$, since $v \in D_0^{\mathcal{I}}$. And if $d^{\leq}([w],[v]) < d^{<}([w],[v])$, then $d^{\leq}([w],[v]) = d^{\overline{\mathcal{I}}}([w],[v]) < a$. Let $d^{\leq}([w],[v]) = b$. Then $[w]R^{\leq b}[v]$ and, since $v \in D_0^{\mathcal{I}}$, we have $w \in (\exists^{\leq b} D_0)^{\mathcal{I}} \subseteq (\exists^{<a} D_0)^{\mathcal{I}}$.

Case $D = \exists^{\leq a} D_0$. Let $w \in (\exists^{\leq a} D_0)^{\mathcal{I}}$, i.e., there is $v \in D_0^{\mathcal{I}}$ with $d^{\mathcal{I}}(w,v) \leq a$. Then $d^{\overline{\mathcal{I}}}([w],[v]) \leq d^{\leq}([w],[v]) \leq a$ in view of Lemma 5 (2); and $[v] \in D_0^{\overline{\mathcal{I}}}$ by the induction hypothesis. Conversely, suppose $[v] \in D_0^{\overline{\mathcal{I}}}$ for some $[v]$ with $d^{\overline{\mathcal{I}}}([w],[v]) \leq a$. Then $d^{\leq}([w],[v]) \leq a$, i.e., $[w]R^{\leq a}[v]$, and $v \in D_0^{\mathcal{I}}$ by the induction hypothesis. Therefore, $w \in (\exists^{\leq a} D_0)^{\mathcal{I}}$. ☐

Symmetric case. Now let $\mathcal{I}$ be a metric min-model. We first note one important property that follows from condition (min). For every $A, B, C \subseteq \Delta^{\mathcal{I}}$, $A \neq \emptyset$, we have

$$\text{if} \quad \forall a \in A \; d^{\mathcal{I}}(a,B) < d^{\mathcal{I}}(a,C) \quad \text{then} \quad d^{\mathcal{I}}(A,B) < d^{\mathcal{I}}(A,C). \tag{9}$$

Indeed, let $d^{\mathcal{I}}(A,B) = d^{\mathcal{I}}(a_0,B)$ and $d^{\mathcal{I}}(A,C) = d^{\mathcal{I}}(a_1,C)$ for some $a_0, a_1 \in A$. Then $d^{\mathcal{I}}(a_0,B) \leq d^{\mathcal{I}}(a_1,B) < d^{\mathcal{I}}(a_1,C)$.

As in the previous case, to construct a finite model $\overline{\mathcal{I}}$ from $\mathcal{I}$ we first introduce the equivalence relation $\sim$ on $\Delta^{\mathcal{I}}$ by (4), denote by $[u]$ the equivalence class of $u \in \Delta^{\mathcal{I}}$, and set

$\Delta^{\overline{\mathcal{I}}} = \{[u] \mid u \in \Delta^{\mathcal{I}}\}$. For every $a \in \overline{\mathrm{P}}(C)$, define relations $R^{<a}$ and $R^{\leq a}$ by (5) and (6). Let $S^{<a} = R^{<a} \cap (R^{<a})^{-1}$ and $S^{\leq a} = R^{\leq a} \cap (R^{\leq a})^{-1}$. Now we set

$$\begin{aligned} d^{<}([u],[v]) &= \min\{a \in \overline{\mathrm{P}}(C) \mid [u]S^{<a}[v]\}, \\ d^{\leq}([u],[v]) &= \min\{a \in \overline{\mathrm{P}}(C) \mid [u]S^{\leq a}[v]\}. \end{aligned}$$

As an easy corollary of Lemma 5 we obtain

Lemma 9. *For all $a, b \in \overline{\mathrm{P}}(C)$ and all $u, v \in \Delta^{\mathcal{I}}$, we have*

(1) $S^{<a} \circ S^{\leq b} \subseteq S^{<a\oplus b}$, $S^{\leq a} \circ S^{<b} \subseteq S^{<a\oplus b}$, $S^{\leq a} \circ S^{\leq b} \subseteq S^{\leq a\oplus b}$,

(2) *if* $d(u,v) < a$ *then* $[u]S^{<a}[v]$ *and* $d^{<}([u],[v]) \leq a$,
if $d(u,v) \leq a$ *then* $[u]S^{\leq a}[v]$ *and* $d^{\leq}([u],[v]) \leq a$,
$d^{\leq}([u],[v]) \leq d^{<}([u],[v])$.

Corollary 10. *The functions $d^{<}$ and $d^{\leq}$ are symmetric and satisfy the triangle inequality.*

As in the previous case, we introduce δ, ε_s, E_s, and J_s; see (7)–(8). Now, for every $s \leq N$ and $([w],[v]) \in J_s$, we choose $\varepsilon([w],[v])$ in E_s in such a way that the condition

$$\varepsilon([w],[v]) < \varepsilon([w'],[v']) \quad \text{iff} \quad d([w],[v]) > d([w'],[v']) \tag{10}$$

holds for all $([w],[v])$ and $([w'],[v'])$ in J_s. Again, this is possible because $|J_s| \leq |E_s|$. Note that (10) guarantees that ε is a symmetric function.

We are now in a position to define the metric function $d^{\overline{\mathcal{I}}}$: for $[u],[v] \in \Delta^{\overline{\mathcal{I}}}$,

- $d^{\overline{\mathcal{I}}}([u],[u]) = 0$,
- $d^{\overline{\mathcal{I}}}([u],[v]) = d^{\leq}([u],[v])$, if $[u] \neq [v]$ and $d^{\leq}([u],[v]) < d^{<}([u],[v])$,
- $d^{\overline{\mathcal{I}}}([u],[v]) = d^{\leq}([u],[v]) - \varepsilon([u],[v])$, if $[u] \neq [v]$ and $d^{\leq}([u],[v]) = d^{<}([u],[v])$.

Finally, we set $D^{\overline{\mathcal{I}}} = \{[u] \mid u \in D^{\mathcal{I}}\}$ and $\ell^{\overline{\mathcal{I}}} = \{\ell\}^{\mathcal{I}}$ for every concept name D and every object name ℓ.

Lemma 11. *$\overline{\mathcal{I}}$ is a metric model.*

Proof. The proof is almost the same as the proof of Lemma 7. The only difference is the symmetry condition, where it is enough to observe that the functions $d^{<}$, $d^{\leq}$ and d are symmetric, whence ε is symmetric, and so is $d^{\overline{\mathcal{I}}}$. □

Lemma 12. *For all $w \in \Delta^{\mathcal{I}}$ and $D \in \mathrm{cl}(C)$, we have $[w] \in D^{\overline{\mathcal{I}}}$ iff $w \in D^{\mathcal{I}}$.*

Proof. The proof (by induction on the construction of D) is similar to that of Lemma 8. The only essential difference is that here we use property (9) instead of picking up representatives in each class $[w]$.

Case $D = D_0 \leftleftarrows D_1$. We assume that $D_0^{\mathcal{I}} \neq \emptyset$ and $D_1^{\mathcal{I}} \neq \emptyset$; otherwise the lemma follows easily from the induction hypothesis. Let $w \in (D_0 \leftleftarrows D_1)^{\mathcal{I}}$. Then $d^{\mathcal{I}}(w', D_0^{\mathcal{I}}) < d^{\mathcal{I}}(w', D_1^{\mathcal{I}})$ for every $w' \in [w]$, and so $d^{\mathcal{I}}([w], D_0^{\mathcal{I}}) < d^{\mathcal{I}}([w], D_1^{\mathcal{I}})$ in view of (9). There exist $w_0 \in [w]$, $v_0 \in D_0^{\mathcal{I}}$ and $[v_1] \in D_1^{\overline{\mathcal{I}}}$ such that $d^{\mathcal{I}}(w_0, v_0) = d^{\mathcal{I}}([w], D_0^{\mathcal{I}})$ and $d^{\overline{\mathcal{I}}}([w],[v_1]) = d^{\overline{\mathcal{I}}}([w], D_1^{\overline{\mathcal{I}}})$.

By the induction hypothesis, $[v_0] \in D_0^{\overline{\mathcal{I}}}$ and $v_1 \in D_1^{\mathcal{I}}$. So we need to show that $d^{\overline{\mathcal{I}}}([w],[v_0]) < d^{\overline{\mathcal{I}}}([w],[v_1])$. Notice first that $d^{\mathcal{I}}([w],[v_0]) = d^{\mathcal{I}}([w], D_0^{\mathcal{I}}) < d^{\mathcal{I}}([w], D_1^{\mathcal{I}}) \le d^{\mathcal{I}}([w],[v_1])$. Next we prove $d^{<}([w],[v_0]) \le d^{\le}([w],[v_1])$. Let $d^{\le}([w],[v_1]) = a$. We may assume that $a < a_N$ because otherwise the statement is trivial. We have $[w_0]S^{\le a}[v_1]$ and $\exists^{<a}D_0, \exists^{\le a}D_1 \in \mathrm{cl}(C)$, since $D_0 \leftleftarrows D_1 \in \mathrm{cl}(C)$. Then $v_1 \in D_1^{\mathcal{I}}$ implies $w_0 \in (\exists^{\le a}D_1)^{\mathcal{I}}$, and so $w_0 \in (D_0 \leftleftarrows D_1)^{\mathcal{I}}$ implies $w_0 \in (\exists^{<a}D_0)^{\mathcal{I}}$. Thus $d^{\mathcal{I}}(w_0, v_0) \le d^{\mathcal{I}}(w_0, D_0^{\mathcal{I}}) < a$, and, by Lemma 9 (2), $d^{<}([w],[v_0]) \le a$, as required.

Consider three subcases. If $d^{\le}([w],[v_1]) < d^{<}([w],[v_1])$, then by definition we have $d^{\overline{\mathcal{I}}}([w],[v_0]) < d^{<}([w],[v_0]) \le d^{\le}([w],[v_1]) = d^{\overline{\mathcal{I}}}([w],[v_1])$.

If $d^{\le}([w],[v_0]) < d^{<}([w],[v_0])$ then $d^{\le}([w],[v_1]) - d^{\le}([w],[v_0]) \ge \delta$, $\varepsilon([w],[v_1]) < \delta/2$ and, therefore,

$$d^{\overline{\mathcal{I}}}([w],[v_1]) - d^{\overline{\mathcal{I}}}([w],[v_0]) \ \ge \ d^{\le}([w],[v_1]) - \varepsilon([w],[v_1]) - d^{\le}([w],[v_0]) \ > \ 0.$$

Finally, if $d^{\le}([w],[v_0]) = d^{<}([w],[v_0])$ and $d^{\le}([w],[v]) = d^{<}([w],[v])$ then

$$d^{\overline{\mathcal{I}}}([w],[v_1]) - d^{\overline{\mathcal{I}}}([w],[v_0]) \ = \ -\varepsilon([w],[v_1]) + \varepsilon([w],[v_0]) \ > \ 0,$$

since $d^{\mathcal{I}}([w],[v_0]) < d^{\mathcal{I}}([w],[v_1])$. Similarly, starting from $w \notin (D_0 \leftleftarrows D_1)^{\mathcal{I}}$, we obtain $d^{\overline{\mathcal{I}}}([w], D_1^{\overline{\mathcal{I}}}) \le d^{\overline{\mathcal{I}}}([w], D_0^{\overline{\mathcal{I}}})$.

The remaining part of the proof is similar to that of Lemma 8 and left to the reader. □

This completes the proof of Theorem 4. □

Remark 13. Theorem 4 and its proof give a nondeterministic exponential time algorithm for checking satisfiability of a given concept C. Indeed, as we have already observed, the size of $\overline{\mathrm{P}}(C)$ and $\mathrm{cl}(C)$ is exponential in $l(C)$. Moreover, the number of different types is also exponential in $l(C)$, since every type T is determined by the following:

- for every subterm D of C, whether D belongs to T,
- for every subterm of C of the form $\exists D$, what is a minimal value b (if any) such that $\exists^{\le b}D$ belongs to T, and whether $\exists^{<b}D$ belongs to T.

Therefore, if C is satisfied in a class of models $\mathcal{M}$ under consideration, then C is satisfied in a finite model in $\mathcal{M}$ of exponential in $l(C)$ size. Moreover, the distance function of this model can possess the values from a prescribed set of exponential in $l(C)$ size. So the existence of such a model can be detected nondeterministically in exponential in $l(C)$ time.

4.2 Proof of Theorem 3 (iii)

Fix an $\mathcal{SL}$-concept C. First we relativise the general definition of a type over a KB Σ to the case of a single concept. Namely, we say that a subset T of $\mathrm{cl}(C)$ is a *type for C* if it satisfies the following conditions:

- $D_1 \sqcap D_2 \in T$ iff $D_1, D_2 \in T$, for every $D_1 \sqcap D_2 \in \mathrm{cl}(C)$,
- $\neg D \in T$ iff $D \notin T$, for every $\neg D \in \mathrm{cl}(C)$,
- $\exists^{<a}D \in T$ implies $\exists^{<b}D, \exists^{\le b}D, \exists D \in T$, for every $\exists D \in \mathrm{cl}(C)$ and all $b \in \mathrm{P}^{+}(C)$, $a \le b$,

- $\exists^{\leq a} D \in T$ implies $\exists^{<b} D, \exists^{\leq b} D, \exists D \in T$, for every $\exists D \in \mathrm{cl}(C)$ and all $b \in \mathrm{P}^+(C)$, $a < b$,

- $D_1 \leftleftarrows D_2, \exists^{\leq a} D_2 \in T$ implies $\exists^{<a} D_1 \in T$, for all $D_1, D_2 \in \mathrm{cl}(C)$ and $a \in \mathrm{P}^+(C)$.

(The last condition of the definition of types over Σ is irrelevant in this case because all models now are based on a single similarity space.) As we observed above, the number of types for C is exponential in $l(C)$.

Let $\mathcal{T}$ be a realisable set of types, let $T_1, \dots, T_n$ be an enumeration of those of them that do not contain nominals, and let $T_{n+1}, \dots, T_m$ be the remaining types in $\mathcal{T}$. As we saw in the proof of Lemma 4, if $\mathcal{T}$ is realised in some model, then it is also realised in a (finite) model $\mathcal{I}$ such that $T^{\mathcal{I}}(u) \neq T^{\mathcal{I}}(v)$ whenever $u \neq v$. In other words, every type from $\mathcal{T}$ is realised by precisely one point from $\mathcal{I}$. Denote by u_i the point realising T_i, $i = 1, \dots, m$.

Not let $k_1, \dots, k_n$ with $k_i \in \mathbb{N}^{>0} \cup \{\omega\}$ be given, and let $k_j = 1$ for $j = n+1, \dots, m$. Our aim is to construct, starting from $\mathcal{I}$, a model $\mathcal{J}$ such that, for every $i = 1, \dots, m$,

$$\{u \in \Delta^{\mathcal{J}} \mid T^{\mathcal{J}}(u) = T_i\} \;=\; k_i.$$

Let U_i be a set of cardinality $k_i - 1$. We assume that all of the U_i and $\Delta^{\mathcal{I}}$ are pairwise disjoint. Let $U_i^+ = U_i \cup \{u_i\}$ and let

$$d \;=\; \min\{d^{\mathcal{I}}(u, v) \mid u, v \in \Delta^{\mathcal{I}},\ u \neq v\}.$$

Construct a new model $\mathcal{J}$ by taking

- $\Delta^{\mathcal{J}} \;=\; U_1^+ \cup \cdots \cup U_m^+$,

- $d^{\mathcal{J}}(u, v) = d^{\mathcal{I}}(u_i, u_j)$ for $u \in U_i^+$, $v \in U_j^+$, $i \neq j$,

- $d^{\mathcal{J}}(u, v) = d$ for $u, v \in U_i$, $u \neq v$,

- $\ell^{\mathcal{J}} \;=\; \ell^{\mathcal{I}}$, for every object name ℓ,

- $u \in A^{\mathcal{J}}$ iff $u \in U_i^+$ and $u_i \in A^{\mathcal{I}}$, for every concept name A.

The reader can readily check by induction that the model $\mathcal{J}$ is as required.

4.3 Proof of Theorem 3 (i)

Now we show that the concept satisfiability problem for $\mathcal{SL}$ is decidable in ExpTime for the binary coding of parameters. Fix again an arbitrary $\mathcal{SL}$-concept C.

For each $a \in \overline{\mathrm{P}}(C)$, we introduce a fresh symbol a^- and set

$$\mathrm{Q}(C) \;=\; \mathrm{P}^+(C) \cup \{a^- \mid a \in \overline{\mathrm{P}}(C)\} \cup \{0\}$$

(note that $\mathrm{Q}(C)$ contains a_N^-, but not a_N). Define a linear order $<$ on $\mathrm{Q}(C)$ in such a way that 0 is its minimal element, and $b < a^- < a < a_N^-$ whenever $b < a$ in $\mathrm{P}^+(C)$.

We are now in a position to introduce the 'building material' for constructing models from types.

A *bouquet for C* is a triple $\mathfrak{B} = (\mathcal{T}_{\mathfrak{B}}, \leq_{\mathfrak{B}}, d_{\mathfrak{B}})$ where

- $\mathcal{T}_{\mathfrak{B}}$ is a set of types for C such that $2 \leq |\mathcal{T}_{\mathfrak{B}}| \leq l(C)$,

- $\leq_{\mathfrak{B}}$ is a linear quasi-order on $\mathcal{T}_{\mathfrak{B}}$ with a unique minimal element $T_{\mathfrak{B}} \in \mathcal{T}_{\mathfrak{B}}$, and
- $d_{\mathfrak{B}}$ is a map from $\mathcal{T}_{\mathfrak{B}}$ to $\mathrm{Q}(C)$,

and the following conditions are satisfied, for all $T, T' \in \mathcal{T}_{\mathfrak{B}}$:

(B1) $d_{\mathfrak{B}}(T) = 0$ iff $T = T_{\mathfrak{B}}$,

(B2) $T \leq_{\mathfrak{B}} T'$ implies $d_{\mathfrak{B}}(T) \leq d_{\mathfrak{B}}(T')$,

(B3) $d_{\mathfrak{B}}(T) = d_{\mathfrak{B}}(T') \in \mathrm{P}^+(C)$ implies $T \leq_{\mathfrak{B}} T' \leq_{\mathfrak{B}} T$,

(B4) $D_1 \leftleftarrows D_2 \in T_{\mathfrak{B}}$ iff $D_1 \leftleftarrows D_2 \in \mathrm{cl}(C)$, there is $T \in \mathcal{T}_{\mathfrak{B}}$ for which $D_1 \in T$, and $D_2 \notin T'$ whenever $T' \leq_{\mathfrak{B}} T$,

(B5) $\exists^{<a} D \in T_{\mathfrak{B}}$ iff $\exists^{<a} D \in \mathrm{cl}(C)$ and there is $T \in \mathcal{T}_{\mathfrak{B}}$ such that $d_{\mathfrak{B}}(T) \leq a^-$ and $D \in T$,

(B6) $\exists^{\leq a} D \in T_{\mathfrak{B}}$ iff $\exists^{\leq a} D \in \mathrm{cl}(C)$ and there is $T \in \mathcal{T}_{\mathfrak{B}}$ such that $d_{\mathfrak{B}}(T) \leq a$ and $D \in T$,

(B7) $\exists D \in T_{\mathfrak{B}}$ iff $\exists D \in T$,

(B8) if $a, b, a{+}b \in \mathrm{P}^+(C)$ and $\exists D \in \mathrm{cl}(C)$, then
$\exists^{<a} D \in T$ and $d_{\mathfrak{B}}(T) \leq b$ imply $\exists^{<a+b} D \in T_{\mathfrak{B}}$,
$\exists^{\leq a} D \in T$ and $d_{\mathfrak{B}}(T) \leq b^-$ imply $\exists^{<a+b} D \in T_{\mathfrak{B}}$,
$\exists^{\leq a} D \in T$ and $d_{\mathfrak{B}}(T) \leq b$ imply $\exists^{\leq a+b} D \in T_{\mathfrak{B}}$.

We will write $T \simeq_{\mathfrak{B}} T'$ for $T \leq_{\mathfrak{B}} T' \leq_{\mathfrak{B}} T$, and $T <_{\mathfrak{B}} T'$ for $T \leq_{\mathfrak{B}} T' \not\leq_{\mathfrak{B}} T$.

We use the notion of a bouquet to encode the local requirements ensuring that the type $T_{\mathfrak{B}}$ can be realised. A 'typical' bouquet is constructed as follows.

Let $\mathcal{I}$ be a model and $w \in \Delta^{\mathcal{I}}$. For every $\exists D \in T^{\mathcal{I}}(w)$ with $D \notin T^{\mathcal{I}}(w)$, choose a point $v \in D^{\mathcal{I}}$ with a minimal distance $d^{\mathcal{I}}(w, v)$; let V denote the set of all selected points. We have $|V| < l(C)$, since $D \in \mathrm{sub}(C)$, for every $\exists D \in \mathrm{cl}(C)$; and we clearly can assume that $T^{\mathcal{I}}(v_1) \neq T^{\mathcal{I}}(v_2)$ for any distinct $v_1, v_2 \in V$. If $|V| \geq 1$, then we define the *bouquet* $(\mathcal{T}_V^{\mathcal{I}}(w), \leq_w, d_w)$ *induced by* w *and* V *in* $\mathcal{I}$ by taking:

$$
\begin{aligned}
\mathcal{T}_V^{\mathcal{I}}(w) &= \{T^{\mathcal{I}}(w)\} \cup \{T^{\mathcal{I}}(v) \mid v \in V\}, \\
T^{\mathcal{I}}(v_1) \leq_w T^{\mathcal{I}}(v_2) & \text{ iff } d^{\mathcal{I}}(w, v_1) \leq d^{\mathcal{I}}(w, v_2), \\
d_w(T^{\mathcal{I}}(v)) &= \begin{cases} a, & \text{if } d^{\mathcal{I}}(w,v) = a \in \mathrm{P}^+(C), \\ a^-, & \text{if } d^{\mathcal{I}}(w,v) < a = \min\{b \in \mathrm{P}^+(C) \mid d^{\mathcal{I}}(w,v) \leq b\}, \\ a_N^-, & \text{if } d^{\mathcal{I}}(w,v) > a_{N-1}. \end{cases}
\end{aligned}
$$

Note that if we require a certain type T realised in $\mathcal{I}$ to be a member of the bouquet then we can add to V a point v such that $d(w, v)$ is minimal with $T = T^{\mathcal{I}}(v)$ and form the bouquet induced by w and $V \cup \{v\}$. In particular, if $\mathcal{I}$ realises at least two distinct types, then we can always find a set V such that w and V induce a bouquet.

Note also that the number of all bouquets is exponential in $l(C)$.

In what follows we will only be working with models satisfying at least two distinct types, because checking satisfiability of a concept in a model satisfying only one type reduces to the case of pure Boolean concepts, which is decidable in NP.

Again we consider two cases.

Non-symmetric case. Let $\mathcal{N}$ be the set of nominals occurring in C. A set B of bouquets is said to be *nominal ready* if there is a set $\{T_\ell \mid \ell \in \mathcal{N}\}$ of types such that

$$B \;=\; \{\mathfrak{B} \mid \mathfrak{B} \text{ is a bouquet, } \{\ell\} \in T \in \mathcal{T}_{\mathfrak{B}} \text{ implies } T = T_\ell\}.$$

Thus, every nominal ready set is determined by a certain set $\{T_\ell \mid \ell \in \mathcal{N}\}$ of the size $< l(C)$, hence there exist only exponentially many in $l(C)$ nominal ready sets.

Let $k = l(C)$. As usual, $\{1, \dots, k\}^*$ denotes the set of all finite sequences of elements of $\{1, \dots, k\}$. By the *full k-ary tree over $\{1, \dots, k\}^*$* we mean the tree that contains the empty sequence ϵ as its root, and the immediate successors (children) of each node α are precisely the nodes αi, where $1 \leq i \leq k$. Given some set L (of labels), a function $K : \{1, \dots, k\}^* \to L$ will be called an *L-labelled tree over $\{1, \dots, k\}^*$*.

A *Hintikka tree satisfying C* is a B-labelled tree K over $\{1, \dots, k\}^*$, for some nominal ready set B of bouquets, such that the following conditions are satisfied (where, as before, $T_{K(\alpha)}$ denotes the unique $\leq_{K(\alpha)}$-minimal element of the set of types $\mathcal{T}_{K(\alpha)}$ in the bouquet $K(\alpha)$):

(H1) $C \in T_{K(\epsilon)}$,

(H2) for every nominal $\ell \in N$, there exists a type in $K(\epsilon)$ containing $\{\ell\}$,

(H3) for every $\alpha \in \{1, \dots, k\}^*$, $K(\alpha)$ is a bouquet such that

$$\mathcal{T}_{K(\alpha)} \setminus \{T_{K(\alpha)}\} \;=\; \{T_{K(\alpha i)} \mid 1 \leq i \leq k\},$$

and $T_{K(\alpha)} \in \mathcal{T}_{K(\alpha i)}$, for all $1 \leq i \leq k$.

Theorem 14. *For every concept C, the following conditions are equivalent:*

(a) *C is satisfiable in a min-model with (tr) and at least two distinct types,*

(b) *there exists a Hintikka tree over $\{1, \dots, k\}^*$ satisfying C, where $k = l(C)$.*

Proof. (a) $\Rightarrow$ (b) Suppose that $C^{\mathcal{I}} \neq \emptyset$ in some model $\mathcal{I}$ with the triangle inequality and at least two distinct types. We define a Hintikka tree K satisfying C by induction as follows. First take some $w \in C^{\mathcal{I}}$ and set

$$K(\epsilon) \;=\; (\mathcal{T}^{\mathcal{I}}_V(w), \leq_w, d_w),$$

where $(\mathcal{T}^{\mathcal{I}}_V(w), \leq_w, d_w)$ is a bouquet induced by w and a suitable set $V \subseteq W$ containing $\{\ell\}^{\mathcal{I}}$ for all ℓ that occur in C.

Suppose now that we have already defined $K(\alpha)$, for some $\alpha \in \{1, \dots, k\}^*$:

$$K(\alpha) \;=\; (\mathcal{T}^{\mathcal{I}}_{V_\alpha}(w_\alpha), \leq_{w_\alpha}, d_{w_\alpha}),$$

where $(\mathcal{T}^{\mathcal{I}}_{V_\alpha}(w_\alpha), \leq_{w_\alpha}, d_{w_\alpha})$ is induced by w_α and a suitable set V_α. For each $j \in \{1, \dots, k\}$, choose some $w_{\alpha j} \in V_\alpha$ so that $V_\alpha = \{w_{\alpha 1}, \dots, w_{\alpha k}\}$, and let

$$K(\alpha j) \;=\; (\mathcal{T}^{\mathcal{I}}_{V_{\alpha j}}(w_{\alpha j}), \leq_{w_{\alpha j}}, d_{w_{\alpha j}}),$$

where $(\mathcal{T}^{\mathcal{I}}_{V_{\alpha j}}(w_{\alpha j}), \leq_{w_{\alpha j}}, d_{w_{\alpha j}})$ is the bouquet induced by $w_{\alpha j}$ and a suitable set $V_{\alpha j}$ which contains some v with $T^{\mathcal{I}}(v) = T^{\mathcal{I}}(w_\alpha)$.

It is easy to see that the resulting K is a Hintikka tree satisfying C.

(b) $\Rightarrow$ (a) Let K be a Hintikka tree over $\{1,\dots,k\}^*$ satisfying C. First we define a distance space (Δ_0, d_0) with the domain $\{1,\dots,k\}^*$ in the following way.

To begin with, take an arbitrary node α and define d_0 for α and its successors $\alpha 1,\dots,\alpha k$. Let δ and $\varepsilon_0,\dots,\varepsilon_N$ be defined by (7). For every $s \le N$, let

$$I_{\alpha,s} \;=\; \{i \mid 1 \le i \le k,\; d_{K(\alpha)}(T_{K(\alpha i)}) = a_s^-\}.$$

Then we choose $\varepsilon(\alpha,\alpha i) \in (\varepsilon_s, 2\varepsilon_s)$, for all $i \in I_{\alpha,s}$, in such way that

$$\varepsilon(\alpha,\alpha i) \ge \varepsilon(\alpha,\alpha j) \quad \text{iff} \quad T_{K(\alpha i)} \le_{K(\alpha)} T_{K(\alpha j)},$$

for all $i, j \in I_{\alpha,s}$. Now define

$$d_0(\alpha,\alpha i) \;=\; \begin{cases} a, & \text{if } \; d_{K(\alpha)}(T_{K(\alpha j)}) = a \;\;\in \mathrm{P}^+(C),\\ a - \varepsilon(\alpha,\alpha j), & \text{if } \; d_{K(\alpha)}(T_{K(\alpha j)}) = a^- \;\in \mathrm{Q}(C) \setminus \mathrm{P}^+(C). \end{cases}$$

Now we can extend d_0 to the whole set Δ_0 as follows. Let $d_0(\alpha,\alpha) = 0$, $d_0(\alpha i,\alpha) = a_N$, and, for every $\alpha j_1 \cdots j_n$,

$$\begin{aligned} d_0(\alpha,\alpha j_1\cdots j_n) &= d_0(\alpha,\alpha j_1) + \cdots + d_0(\alpha j_1\cdots j_{n-1},\alpha j_1\cdots j_n),\\ d_0(\alpha j_1\cdots j_n,\alpha) &= d_0(\alpha j_1\cdots j_n,\alpha j_1\cdots j_{n-1}) + \cdots + d_0(\alpha j_1,\alpha). \end{aligned}$$

Finally, for every two nodes $\alpha j_1\cdots j_n$ and $\alpha j_1'\cdots j_{n'}'$ such that $j_1 \ne j_1'$, we let

$$d_0(\alpha j_1\cdots j_n, \alpha j_1'\cdots j_{n'}') \;=\; d_0(\alpha j_1\cdots j_n,\alpha) + d_0(\alpha,\alpha j_1'\cdots j_{n'}').$$

One can easily see that

$$T_{K(\alpha i)} \le_{K(\alpha)} T_{K(\alpha j)} \quad \text{iff} \quad d_0(\alpha,\alpha i) \le d_0(\alpha,\alpha j) \tag{11}$$

for all $\alpha \in \{1,\dots,k\}^*$ and $1 \le i,j \le k$.

Finally, we define an interpretation $\mathcal{I}_0$ on the space (Δ_0, d_0) by taking, for every concept name $A \in \mathsf{N_C}$ and every object name $\ell \in \mathsf{N_O}$,

$$\begin{aligned} A^{\mathcal{I}_0} &= \{\alpha \in \{1,\dots,k\}^* \mid A \in T_{K(\alpha)}\},\\ \ell^{\mathcal{I}_0} &= \{\alpha \in \{1,\dots,k\}^* \mid \{\ell\} \in T_{K(\alpha)}\}. \end{aligned}$$

Observe that this interpretation is not 'real' because object names can be interpreted by non-singleton sets, but let us forget about this for the moment.

Lemma 15. *Let $\alpha,\beta \in \{1,\dots,k\}^*$ and $D \in T_{K(\beta)}$, for some $\exists D \in \mathrm{cl}(C)$. Then either $D \in T_{K(\alpha)}$, or $D \in T_{K(\alpha j)}$ and $d_0(\alpha,\alpha j) \le d_0(\alpha,\beta)$, for some $j \in \{1,\dots,k\}$.*

Proof. First, using condition (B7), it is not hard to show by induction on the length of $\alpha \in \{1,\dots,k\}^*$ that

$$\exists C' \in T_{K(\alpha)} \;\text{ iff }\; \exists C' \in T_{K(\epsilon)},$$

for all $\exists C' \in \mathrm{cl}(C)$. Therefore, $T_{K(\alpha)}$ contains $\exists D$.

If $D \in T_{K(\alpha)}$ then we are done. Suppose that $D \notin T_{K(\alpha)}$. Then $\alpha \neq \beta$ and we can choose $j \in \{1, \ldots, k\}$ such that $d_0(\alpha, \alpha j)$ is minimal with $D \in T_{K(\alpha j)}$.

If β is not a successor of α then, by the definition of d_0, $d_0(\alpha, \alpha i) < a_N \leq d_0(\alpha, \beta)$, as required. So, let $\beta = \alpha j_1 \cdots j_n$ and proceed by induction on n.

$n = 1$: This means that $\beta = \alpha j_1$, and we are done again.

$n = 2$: Let $a = d_0(\alpha, \alpha j)$, $b = d_0(\alpha, \alpha j_1)$, $c = d_0(\alpha j_1, \beta)$, and r, s, t be such that

$$d_{K(\alpha)}(T_{K(\alpha j)}) \in \{a_r^-, a_r\}, \quad d_{K(\alpha)}(T_{K(\alpha j_1)}) \in \{a_s^-, a_s\}, \quad d_{K(\alpha j_1)}(T_{K(\beta)}) \in \{a_t^-, a_t\}.$$

We need to show $a \leq b + c$. If $a_r < a_s + a_t$ then we have

$$b + c - a \;>\; (a_s + a_t - a_r) - 2\varepsilon_s - 2\varepsilon_t \;\geq\; \delta - \delta/2 - \delta/2 \;=\; 0.$$

Therefore, we may assume that $a_s + a_t \leq a_r$, in particular $s, t < r$ and $a_s + a_t \in \mathrm{P}^+(C)$. Four cases are possible now: $b < a_s$, $c < a_t$; $b < a_s$, $c = a_t$; $b = a_s$, $c < a_t$; and $b = a_s$, $c = a_t$. As all of them are similar, and we only consider the second one as most demonstrative.

Since $D \in T_{K(\beta)}$, we have $\exists^{\leq a_t} D \in T_{K(\alpha j_1)}$ by (B6), whence $\exists^{< a_s + a_t} D \in T_{K(\alpha)}$ by (B8), and $d_{K(\alpha)}(T_{K(\alpha j)}) \leq (a_s + a_t)^-$ in view of (B5) and by the choice of j. Thus, $a_r = a_s + a_t$, $a = a_r^-$, and

$$b + c - a \;>\; -2\varepsilon_s - 2\varepsilon_t + \varepsilon_r \;\geq\; 0,$$

since $\varepsilon_r, \varepsilon_s \leq \varepsilon_r/4$.

$n > 2$: Let $\alpha' = \alpha j_1$. By the induction hypothesis, there exists $j' \in \{1, \ldots, k\}$ such that $D \in T_{K(\alpha' j')}$ and $d_0(\alpha', \alpha' j') \leq d_0(\alpha', \beta)$. Again by the induction hypothesis, there exists $j \in \{1, \ldots, k\}$ such that $D \in T_{K(\alpha j)}$ and $d_0(\alpha, \alpha j) \leq d_0(\alpha, \alpha' j')$. Hence

$$d_0(\alpha, \alpha j) \leq d_0(\alpha, \alpha') + d_0(\alpha', \alpha' j') \leq d_0(\alpha, \alpha') + d_0(\alpha', \beta) = d_0(\alpha, \beta),$$

according to the definition of d_0. □

Lemma 16. *For every $D \in \mathrm{cl}(C)$, we have $\alpha \in D^{\mathcal{I}_0}$ iff $D \in T_{K(\alpha)}$.*

Proof. The proof is by induction on the construction of D. The basis of induction and the Boolean cases are trivial.

Case $D = D_0 \leftleftarrows D_1$. Let $\alpha \in (D_0 \leftleftarrows D_1)^{\mathcal{I}_0}$, i.e., $d_0(\alpha, D_0^{\mathcal{I}_0}) < d_0(\alpha, D_1^{\mathcal{I}_0})$. By Lemma 15,

$$d_0(\alpha, D_0^{\mathcal{I}_0}) = d_0(\alpha, D_0^{\mathcal{I}_0} \cap X), \quad \text{where } X = \{\alpha\} \cup \{\alpha j \mid 1 \leq j \leq k\}. \tag{12}$$

Hence, there exists $\beta \in D_0^{\mathcal{I}_0} \cap X$ such that $d_0(\alpha, \beta) < d_0(\alpha, \beta')$ for all $\beta' \in D_1^{\mathcal{I}_0} \cap X$. By the induction hypothesis and (11), this means that $D_0 \in T_{K(\beta)}$ and $T_{K(\beta)} <_{K(\alpha)} T$, for all $T \in \mathcal{T}_{K(\alpha)}$ with $D_1 \in T$. Thus $D_0 \leftleftarrows D_1 \in T_{K(\alpha)}$ by (B4).

For the converse, suppose $D_0 \leftleftarrows D_1 \in T_{K(\alpha)}$, i.e., by (B4), there exists $T \in \mathcal{T}_{K(\alpha)}$ such that $D_0 \in T$ and $T <_{K(\alpha)} T'$ for all $T' \in \mathcal{T}_{K(\alpha)}$ with $D_1 \in T'$. Let $T = T_{K(\alpha')}$ for some $\alpha' \in X$. Then $\alpha' \in D_0^{\mathcal{I}_0}$ by the induction hypothesis. If $D_1^{\mathcal{I}_0} = \emptyset$ then $\alpha \in (D_0 \leftleftarrows D_1)^{\mathcal{I}_0}$ holds trivially. Let $D_1^{\mathcal{I}_0} \neq \emptyset$. By Lemma 15, there is $\beta' \in D_1^{\mathcal{I}_0} \cap X$ such that $d_0(\alpha, D_1^{\mathcal{I}_0}) = d_0(\alpha, \beta')$. By the induction hypothesis, we have $D_1 \in T_{K(\beta')}$, and so $T_{K(\alpha')} <_{K(\alpha)} T_{K(\beta')}$, whence $d_0(\alpha, \alpha') < d_0(\alpha, \beta')$ by (11). But then $\alpha \in (D_0 \leftleftarrows D_1)^{\mathcal{I}_0}$ by the choice of α' and β'.

Case $D = \exists^{<a} D_0$. Let $\alpha \in (\exists^{<a} D_0)^{\mathcal{I}_0}$, i.e., $d_0(\alpha, D_0^{\mathcal{I}_0}) < a$. By Lemma 15, we have (12). Hence there exists $\beta \in D_0^{\mathcal{I}_0} \cap X$ such that $d_0(\alpha, \beta) < a$. By the induction hypothesis and the definition of d_0, we obtain that $D_0 \in T_{K(\beta)}$ and $d_{K(\alpha)}(T_{K(\beta)}) \leq a^-$. Thus $\exists^{<a} D_0 \in T_{K(\alpha)}$ by (B5).

For the converse, let $\exists^{<a} D_0 \in T_{K(\alpha)}$, i.e., by (B4), $d_{K(\alpha)}(T) \leq a^-$ and $D_0 \in T$ for some $T \in \mathcal{T}_{K(\alpha)}$. Let $T = T_{K(\alpha')}$ for some $\alpha' \in X$. Then $\alpha' \in D_0^{\mathcal{I}_0}$ by the induction hypothesis, and $d_0(\alpha, \alpha') < a$ by the definition of d_0. But then $\alpha \in (\exists^{<a} D_0)^{\mathcal{I}_0}$.

Case $D = \exists^{\leq a} D_0$ is similar to the previous one. ☐

Thus, it remains to fix the problem with nominals, which can be done as follows. We apply to the pseudo-interpretation $\mathcal{I}_0$ the 'filtration' procedure described in the proof of Theorem 4. Since, by definition, the Hintikka tree K is labelled by a nominal ready set of bouquets and since the filtration results in a model where each type can be realised only once, we then obtain an $\mathcal{SL}$-model satisfying C where each set of the form $\{\ell\}$ is interpreted by a singleton set. ☐

We are now in a position to prove the ExpTime upper bound by a reduction to the emptiness problem for finite looping tree automata; see [35, 32]. Recall that a *finite looping tree automaton $\mathcal{A}$ for infinite k-ary trees* is a quadruple (Σ, Q, Γ, Q_0), where

- Σ is a (nonempty) finite alphabet,
- Q is a (nonempty) finite set of states of the automaton,
- $\Gamma \subseteq \Sigma \times Q \times Q^k$ is a transition relation,
- $Q_0 \subseteq Q$ is a (nonempty) set of start states of the automaton.

Let T be a Σ-labelled tree over $\{1, \ldots, k\}^*$. A *run* of $\mathcal{A}$ on T is a function $R : \{1, \ldots, k\}^* \to Q$ such that

- $R(\epsilon) \in Q_0$, and
- $\big(T(\alpha), R(\alpha), (R(\alpha 1), \ldots, R(\alpha k))\big) \in \Gamma$ for all nodes α of T.

$\mathcal{A}$ *accepts* T if there exists a run R of $\mathcal{A}$ on T. The following *emptiness problem* for looping automata is decidable in polynomial time [32]: given a looping automaton for k-ary trees, decide whether the set of trees it accepts is empty.

To reduce the satisfiability problem for $\mathcal{SL}$-concepts C in $\mathcal{M}$, we associate with every concept C and every nominal ready set B of bouquets for C a finite looping automaton $\mathcal{A}_C^B = (\Sigma, Q, \Gamma, Q_0)$ which is defined as follows:

- Σ is the set of types occurring in bouquets from B,
- $Q = B$,
- $Q_0 = \{\mathfrak{B} \in B \mid C \in T_{\mathfrak{B}}, \mathfrak{B}$ contains a type containing ℓ, for every ℓ in $C\}$,
- $\big(T, \mathfrak{B}_0, (\mathfrak{B}_1, \ldots, \mathfrak{B}_k)\big) \in \Gamma$ iff $T_{\mathfrak{B}_0} = T$, $\mathcal{T}_{\mathfrak{B}_0} \setminus \{T_{\mathfrak{B}_0}\} = \{T_{\mathfrak{B}_1}, \ldots, T_{\mathfrak{B}_k}\}$, and $T_{\mathfrak{B}_0} \in \mathcal{T}_{\mathfrak{B}_i}$, for $1 \leq i \leq k$.

It follows immediately from Theorem 14 and the given definitions that the runs of $\mathcal{A}_C^B$ on Σ-labelled trees are exactly the B-labelled Hintikka trees satisfying C.

Lemma 17. *A concept C is satisfiable in a min-model (with at least two types) iff there exists a nominal ready set B such that $\mathcal{A}_C^B$ accepts at least one tree.*

As there are only exponentially many different nominal ready sets B and as $\mathcal{A}_C^B$ is only exponential in $l(C)$, the satisfiability problem in min- (and finite) models which satisfy the triangle inequality is decidable in ExpTime.

Symmetric case. The construction of the min-model in the proof of Theorem 14 cannot be directly extended to the symmetric case because we had there $d_0(\alpha i, \alpha) = a_N > d_0(\alpha, \alpha i)$. The straightforward idea of defining $d_0(\alpha i, \alpha) = d_0(\alpha, \alpha i)$ could lead to the situation where $d_0(\alpha, \alpha i) \geq d_0(\alpha i, \alpha i j) \geq d_0(\alpha i j, \alpha i j k) \geq \dots$ and infinitely many $\geq$ are actually $>$. The corresponding metric model would not be then a min-model. Our first task is to guarantee that such situations are impossible.

With each Hintikka tree K we associate a certain node-colouring c. The set of colours is: inc (for increasing), const (for constant), and dec (for decreasing), and c is a map from $\{1, \dots, k\}^*$ to $\{\mathsf{inc}, \mathsf{const}, \mathsf{dec}\}$ defined by the following rules. The root ϵ and its immediate successors $1, \dots, k$ are coloured in the same way, say, $c(\epsilon) = c(1) = \dots = c(k) = \mathsf{inc}$. And for every $\alpha i j \in \{1, \dots, k\}^*$ we set

$$c(\alpha i j) = \begin{cases} \mathsf{dec}, & \text{if } T_{K(\alpha i j)} <_{K(\alpha i)} T_{K(\alpha)} \\ \mathsf{const}, & \text{if } T_{K(\alpha i j)} \simeq_{K(\alpha i)} T_{K(\alpha)} \\ \mathsf{inc}, & \text{if } T_{K(\alpha)} <_{K(\alpha i)} T_{K(\alpha i j)}. \end{cases}$$

We call K a *min-tree* if the following conditions hold:

(H4) $d_{K(\alpha i)}(T_{K(\alpha)}) \leq d_{K(\alpha)}(T_{K(\alpha i)})$, for all $\alpha \in \{1, \dots, k\}^*$ and $1 \leq i \leq k$,

(H5) every branch with infinitely many dec nodes also contains infinitely many inc nodes.

Note first that we can 'prune' Hintikka trees with 'loops' in the following sense:

Lemma 18. *Let K be a Hintikka tree satisfying C, and let $\alpha_0, \alpha_1, \dots, \alpha_n, \alpha_{n+1}$ be a path in K such that $(K(\alpha_0), K(\alpha_1)) = (K(\alpha_n), K(\alpha_{n+1}))$. For $\beta \in \{1, \dots, k\}^*$, let*

$$K^\sharp(\beta) = K(\beta^\sharp), \quad \textit{where} \quad \beta^\sharp = \begin{cases} \alpha_{n+1}\beta_1, & \textit{if } \beta = \alpha_1\beta_1 \textit{ for some } \beta_1 \neq \epsilon, \\ \beta, & \textit{otherwise.} \end{cases}$$

Then $K^\sharp$ is also a Hintikka tree satisfying C, and $c^\sharp(\beta) = c(\beta^\sharp)$, for all $\beta \in \{1, \dots, k\}^*$, where $c^\sharp$ and c are the colourings of $K^\sharp$ and K, respectively.

Proof. The straightforward proof is left to the reader. $\square$

Let K be a Hintikka tree and $\overline{\alpha} = (\alpha_1, \dots, \alpha_n)$ be a path in K. We say that $\overline{\alpha}$ is a *descending path of degree m* if $c(\alpha_1), \dots, c(\alpha_n) \in \{\mathsf{dec}, \mathsf{const}\}$ and $\overline{\alpha}$ contains exactly m nodes coloured with dec.

Lemma 19. *Let K be a min-tree over $\{1, \dots, k\}^*$ satisfying C. Let $h = b_C^2$, where b_C is the number of all bouquets for C. Then there exists a min-tree H over $\{1, \dots, k\}^*$ satisfying C and such that every descending path in H is of degree $< h$.*

Proof. Call a descending path $\alpha_0,\dots,\alpha_{n+1}$ in K *special* if $n>1$, $c(\alpha_1)=c(\alpha_{n+1})=\mathsf{dec}$, and $(K(\alpha_0),K(\alpha_1))=(K(\alpha_n),K(\alpha_{n+1}))$. We say that $\alpha_0,\dots,\alpha_{n+1}$ is *maximal* if no path of the form $\alpha_0,\dots,\alpha_r$ with $r>n+1$ is special in K. In view of (H5), every special path in K is contained in a maximal one.

We note also that every descending path $\beta_1,\dots,\beta_l$ of degree $\geq h$ gives rise to some special path. Indeed, let β_0 be the immediate predecessor of β_1. Since $K(\alpha)\neq K(\alpha i)$ for all $\alpha i\in\{1,\dots,k\}^*$, the number of possible combinations $(K(\beta_{m-1}),K(\beta_m))$ is less than h, i.e., less then the number of β_m with $c(\beta_m)=\mathsf{dec}$. Therefore $(K(\beta_{m-1}),K(\beta_m))=(K(\beta_{n-1}),K(\beta_n))$ for some $1\leq m<n\leq l$ with $c(\beta_m)=c(\beta_n)=\mathsf{dec}$, i.e., $\beta_{m-1},\dots,\beta_n$ is a special path.

Let α_0 be a node of a minimal length in K that starts some special path. Let $\alpha_0,\dots,\alpha_{n+1}$ be a maximal special path. By pruning the tree as described in Lemma 18, we obtain a min-tree $K^\sharp$ satisfying C and such that no special path in $K^\sharp$ can start with the pair α_0,α_1 or with a node shorter than α_0. Indeed, suppose that $\alpha_0,\dots,\alpha_{m+1}$ is a special path in $K^\sharp$. Then

$$\alpha_0,\alpha_1,\dots,\alpha_{n+1},\alpha_2^\sharp,\dots\alpha_m^\sharp,\alpha_{m+1}^\sharp$$

is a special path in K, contrary to the maximality assumption.

Suppose now that $\beta_0,\dots,\beta_{m+1}$ is a special path in $K^\sharp$ such that β_0 is shorter than α_0. If $\alpha_1=\beta_l$ for some $l\leq m$ then we arrive to a contradiction in the same way as before. Otherwise, we have $\beta_l^\sharp=\beta_l$ for all $l\leq m{+}1$. Hence $\beta_0,\dots,\beta_{m+1}$ is a special path in K itself, and β_0 is shorter than α_0, which is again a contradiction.

Thus we can construct a sequence $K_0,K_1,K_2,\dots$ of min-trees satisfying C such that, for all $m<n$ in $\mathbb{N}$ and $\alpha\in\{1,\dots,k\}^m$,

$$K_0=K,\quad K_n(\alpha)=K_m(\alpha),\quad K_n \text{ contains no special path starting with } \alpha.$$

Define a labelled tree H by taking

$$H(\alpha)\;=\;K_n(\alpha),\qquad \text{whenever } \alpha\in\{1,\dots,k\}^n,$$

for all $n\in\mathbb{N}$. Then H is a min-tree satisfying C and containing no special paths, in particular, no descending paths of degree $\geq h$. □

We are now in a position to prove the following:

Theorem 20. *For every concept C, the following conditions are equivalent:*

(a) *C is satisfiable in some metric min-model with at least two distinct types,*

(b) *there exists a min-tree satisfying C over $\{1,\dots,k\}^*$, where $k=l(C)$.*

Proof. (a) ⇒ (b) Suppose that $C^{\mathcal{I}}\neq\emptyset$ in some metric model $\mathcal{I}$ with at least two distinct types. Let K be the Hintikka tree K constructed in the same way as in the proof of Theorem 14, (a) ⇒ (b). We only need to show that (H4) and (H5) hold for K.

(H4) follows from the requirement $T_{K(\alpha)}\in\mathcal{T}_{K(\alpha i)}$ for all $\alpha i\in\{1,\dots,k\}^*$ [see (H3)] and the construction of bouquets.

Now suppose that (H5) does not hold, i.e., there exists a branch $(\alpha_m\mid m\in\mathbb{N})$ in K such that $c(\alpha_m)\in\{\mathsf{dec},\mathsf{const}\}$ for all m, and $c(\alpha_m)=\mathsf{dec}$, for infinitely many m. As the set $\{T^{\mathcal{I}}(\alpha_m)\mid m\in\mathbb{N}\}$ is finite and $T_{K(\alpha_m)}\neq T_{K(\alpha_{m+1})}$ for every m, we must have two distinct types T,T' such that the set of pairs

$$X\;=\;\big\{(\alpha_m,\alpha_{m+1})\;\big|\;(T_{K(\alpha_m)},T_{K(\alpha_{m+1})})=(T,T')\big\}$$

is infinite. For each $m \in \mathbb{N}$, take the point $w_{\alpha_m} \in \Delta^{\mathcal{I}}$ which induces the node α_m. As $\mathcal{I}$ is symmetric and by the construction of bouquets, we have $d(w_{\alpha_m}, w_{\alpha_{m+1}}) \geq d(w_{\alpha_{m+1}}, w_{\alpha_{m+2}})$ whenever $c(\alpha_{m+2}) = \mathsf{const}$, and $d(w_{\alpha_m}, w_{\alpha_{m+1}}) > d(w_{\alpha_{m+1}}, w_{\alpha_{m+2}})$ whenever $c(\alpha_{m+2}) = \mathsf{dec}$. Now let $Z \subseteq \Delta^{\mathcal{I}}$ and $Z' \subseteq \Delta^{\mathcal{I}}$ be defined by

$$Z \;=\; \{w_{\alpha_m} \mid (\alpha_m, \alpha_{m+1}) \in X\} \quad \text{and} \quad Z' \;=\; \{w_{\alpha_{m+1}} \mid (\alpha_m, \alpha_{m+1}) \in X\}.$$

As $\mathcal{I}$ is a min-model, there are points $u \in Z$ and $u' \in Z'$ such that $d^{\mathcal{I}}(u, u') = d^{\mathcal{I}}(Z, Z')$. By the definition of bouquets induced by points of $\mathcal{I}$, we may assume that (u, u') gives rise to a $(\alpha_r, \alpha_{r+1}) \in X$. Now take some $(\alpha_s, \alpha_{s+1}) \in X$ such that $s > r$ and a dec node occurs between α_r and α_s. Let v and v' be the points from $\mathcal{I}$ inducing α_s and α_{s+1}. Since all α_m, $m \in \mathbb{N}$, are dec or const nodes we come to the conclusion that $d^{\mathcal{I}}(v, v') < d^{\mathcal{I}}(u, u') = d^{\mathcal{I}}(Z, Z')$, which is a contradiction.

(b) $\Rightarrow$ (a) Let K be a min-tree satisfying C. In view of Lemma 19, we can assume that every descending path in K is of degree $< h = b_C^2$.

Define a metric space (Δ_1, d_1) over $\{1, \dots, k\}^*$ in the following way. First, we introduce d_1 for every node and its successors by induction. Let δ and $\varepsilon_0, \dots, \varepsilon_N$ be defined by (7). For every $s \leq N$, let E_s be an arbitrary fixed subset of $(\varepsilon_s, 2\varepsilon_s)$ of cardinality $h \cdot k$, and let F_s be a countable subset of $(\varepsilon_s, \min E_s)$ with the order type of the negative integers.

For every $\alpha \in \{1, \dots, k\}^*$, set

$$I_{\alpha,s} \;=\; \{i \mid 1 \leq i \leq k,\ d_{K(\alpha)}(T_{K(\alpha i)}) = a_s^-\}.$$

For every $s \leq N$, choose $\varepsilon(\epsilon, i) \in F_s$, for $i \in I_{\epsilon,s}$, in such a way that

$$\varepsilon(\epsilon, i) \geq \varepsilon(\epsilon, j) \quad \text{iff} \quad T_{K(i)} \leq_{K(\epsilon)} T_{K(j)},$$

for all $i, j \in I_{\epsilon,s}$. Then we define

$$d_1(\epsilon, i) \;=\; \begin{cases} a, & \text{if } \ d_{K(\epsilon)}(T_{K(j)}) = a \ \ \in \mathrm{P}^+(C), \\ a - \varepsilon(\epsilon, j), & \text{if } \ d_{K(\epsilon)}(T_{K(j)}) = a^- \in \mathrm{Q}(C) \setminus \mathrm{P}^+(C). \end{cases}$$

Now suppose that $d_1(\alpha', \alpha)$ is already defined, where α' is the immediate predecessor of α. For all $s \leq N$, choose $\varepsilon(\alpha, \alpha i) \in E_s \cup F_s$, for $i \in I_{\alpha,s}$, satisfying the following conditions:

(D1) if $c(\alpha i) = \mathsf{const}$ and $d_1(\alpha', \alpha) \geq a_s$, then $\varepsilon(\alpha, \alpha i) \in F_s$,

(D2) if $c(\alpha i) = \mathsf{const}$ and $d_1(\alpha', \alpha) < a_s$, then $\varepsilon(\alpha, \alpha i) = \varepsilon(\alpha', \alpha) = a_s - d_1(\alpha', \alpha)$,

(D3) if $c(\alpha i) = \mathsf{inc}$, then $\varepsilon(\alpha, \alpha i) \in F_s$,

(D4) if $c(\alpha i) = \mathsf{dec}$, then

$$\varepsilon(\alpha, \alpha i) \;=\; \min\{x \in E_s \mid x > \varepsilon(\alpha, \alpha j),\ \text{for all } j \in I_{\alpha,s} \text{ with } T_{K(\alpha i)} <_{K(\alpha)} T_{K(\alpha j)}\},$$

(D5) $\varepsilon(\alpha, \alpha i) \geq \varepsilon(\alpha, \alpha j)$ iff $T_{K(\alpha i)} \leq_{K(\alpha)} T_{K(\alpha j)}$, for all $i, j \in I_{\alpha,s}$.

The next lemma shows that the sets E_s is large enough to satisfy these conditions.

Lemma 21. *Let $\varepsilon(\alpha, \alpha i) \in E_s$, for some $\alpha i \in \{1, \dots, k\}$ and $s \leq N$. Then αi is the end of some descending path of positive degree. Let m be degree of the longest such path, and $E(\alpha i)$ denote the set $\{x \in E_s \mid x \leq \varepsilon(\alpha, \alpha i)\}$. Then $|E(\alpha i)| \leq m \cdot k$.*

Proof. Note first that α cannot be empty, since $\varepsilon(\alpha, \alpha i) \notin F_s$. Let α' be the immediate predecessor of α.

CLAIM 1. *Let $\alpha_1, \ldots, \alpha_{n+1}$ be a path such that $\alpha_{n+1} = \alpha i$ and $c(\alpha_1) \neq \mathsf{const} = c(\alpha_m)$, for all $2 \leq m \leq n{+}1$. Then $c(\alpha_1) = \mathsf{dec}$ and $\varepsilon(\alpha_m, \alpha_{m+1}) = \varepsilon(\alpha, \alpha i)$, for all $m < n$.*

The proof is by induction on n.

Case $n = 0$: then $c(\alpha i) \neq \mathsf{const}$. But $c(\alpha i) = \mathsf{inc}$ implies $\varepsilon(\alpha, \alpha i) \in F_s$, contrary to $\varepsilon(\alpha, \alpha i) \in E_s$. Hence $c(\alpha i) = \mathsf{dec}$, as required.

Case $n > 0$: then $c(\alpha i) = \mathsf{const}$, i.e., $T_{K(\alpha i)} \simeq_{K(\alpha)} T_{K(\alpha')}$. By applying (B2) and (H4) we obtain $a_s^- = d_{K(\alpha)}(T_{K(\alpha i)}) = d_{K(\alpha)}(T_{K(\alpha')}) \leq d_{K(\alpha')}(T_{K(\alpha)})$. So we have either $d_1(\alpha', \alpha) \geq a_s$, or $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$. In the former case we obtain $\varepsilon(\alpha, \alpha i) \in F_s$ by (D1), contrary to $\varepsilon(\alpha, \alpha i) \in E_s$. Therefore $\varepsilon(\alpha, \alpha i) = \varepsilon(\alpha', \alpha)$ by (D2), i.e., $\varepsilon(\alpha', \alpha) \in E_s$, and we are done by the induction hypothesis (note that $\alpha' = \alpha_{n-1}$).

CLAIM 2. *Let $E'(\alpha i) = \{x \in E_s \mid a_s - d_1(\alpha', \alpha) < x \leq \varepsilon(\alpha, \alpha i)\}$. Then $|E'(\alpha i)| \leq k$.*

In view of Claim 1, we have $c(\alpha i) \in \{\mathsf{dec}, \mathsf{const}\}$. Suppose that $c(\alpha i) = \mathsf{const}$. Then, again by Claim 1, we obtain that $a_s - d_1(\alpha, \alpha') = \varepsilon(\alpha', \alpha) = \varepsilon(\alpha, \alpha i)$, i.e., $|E'(\alpha i)| = 0$.

We assume therefore that $c(\alpha i) = \mathsf{dec}$. Then $T_{K(\alpha')} <_{K(\alpha)} T_{K(\alpha i)}$ and by applying (B2) and (H4) we obtain $a_s^- = d_{K(\alpha)}(T_{K(\alpha i)}) \leq d_{K(\alpha)}(T_{K(\alpha')}) \leq d_{K(\alpha')}(T_{K(\alpha)})$. So we have either $d_1(\alpha', \alpha) \geq a_s$ or $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$.

Consider first the case $d_1(\alpha', \alpha) \geq a_s$. Then, for all $j \in I_{\alpha,s}$ such that $c(\alpha j) \in \{\mathsf{const}, \mathsf{inc}\}$, we obtain $\varepsilon(\alpha, \alpha j) \in F_s$ by $(D1)$ or $(D3)$. Let

$$\{j \in I_{\alpha,s} \mid c(\alpha j) = \mathsf{dec}\} = \{j_1, \ldots, j_l\}, \quad \text{where } T_{K(\alpha j_r)} \leq_{K(\alpha)} T_{K(\alpha j_s)} \text{ for } r \leq s. \tag{13}$$

Then, in view of (D4), we obtain

$$\begin{aligned} \varepsilon(\alpha, \alpha j_1) &= \min E_s, \\ \varepsilon(\alpha, \alpha j_{r+1}) &= \varepsilon(\alpha, \alpha j_r) \text{ or } \min\{x \in E_s \mid x > \varepsilon(\alpha, \alpha j_r)\}, \quad \text{for all } r < l. \end{aligned}$$

Therefore, $|E'(\alpha i)| \leq |I_{\alpha,s}| \leq k$.

Suppose now that $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$. Then, for all $j \in I_{\alpha,s}$ with $c(\alpha j) \in \{\mathsf{const}, \mathsf{inc}\}$, we have $T_{K(\alpha i)} \leq_{K(\alpha)} T_{K(\alpha')}$, and so $\varepsilon(\alpha', \alpha) \geq \varepsilon(\alpha, \alpha j)$ by (D5). Consider again (13). Then, in view of (D4), we obtain

$$\begin{aligned} \varepsilon(\alpha, \alpha j_1) &= \min\{x \in E_s \mid x > \varepsilon(\alpha', \alpha)\}, \\ \varepsilon(\alpha, \alpha j_{r+1}) &= \varepsilon(\alpha, \alpha j_r) \text{ or } \min\{x \in E_s \mid x > \varepsilon(\alpha, \alpha j_r)\}, \quad \text{for all } r < l. \end{aligned}$$

Therefore, $|E'(\alpha i)| \leq |I_{\alpha,s}| \leq k$.

Let us now return to the proof of Lemma 21. In view of Claim 1, we can consider the longest descending path $\alpha_0, \ldots, \alpha_{n+1}$ with $\alpha_{n+1} = \alpha i$. We prove $|E(\alpha i)| \leq m \cdot k$ by induction on n.

Case $n = 0$: then $c(\alpha) = \mathsf{inc}$ and, in view of Claim 1, $c(\alpha i) = \mathsf{dec}$, i.e., $m = 1$. As before, we have either $d_1(\alpha', \alpha) \geq a_s$ or $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$. And since $c(\alpha) = \mathsf{inc}$, we have $\varepsilon(\alpha', \alpha) \in F_s$ in the latter case by (D5). Therefore, in any case, $a_s - d_1(\alpha', \alpha) < x$ for all $x \in E_s$. We obtain $|E(\alpha i)| \leq k \leq m \cdot k$ by Claim 2.

Case $n > 0$: then $c(\alpha i) \in \{\mathsf{dec}, \mathsf{const}\}$ and $m \geq 1$ by Claim 1. Suppose first that $c(\alpha i) = \mathsf{const}$. Then, in view of Claim 1, $E(\alpha i) = E(\alpha)$ and the degree of the longest descending path which ends at α is m as well. We obtain $|E(\alpha i)| \leq m \cdot k$ by the induction hypothesis.

Suppose now that $c(\alpha i) = \mathsf{dec}$. Then, as before, we have either $d_1(\alpha', \alpha) \geq a_s$ or $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$. If $d_1(\alpha', \alpha) \geq a_s$ or $\varepsilon(\alpha', \alpha) \in F_s$, then $E(\alpha i) = E'(\alpha i)$, and so $|E(\alpha i)| \leq k (\leq m \cdot k)$ by Claim 2. Assume therefore that $d_1(\alpha', \alpha) = a_s - \varepsilon(\alpha', \alpha)$, where $\varepsilon(\alpha', \alpha) \in E_s$. Then the longest descending path with the end α is of degree $m - 1$. Hence $|E(\alpha)| \leq (m-1)k$ by the induction hypothesis, and $|E'(\alpha i)| \leq k$ by Claim 2. Finally, we obtain $|E(\alpha i)| = |E(\alpha) \cup E'(\alpha i)| \leq m \cdot k$. $\square$

Thus, we can set

$$d_1(\alpha, \alpha i) \;=\; \begin{cases} a, & \text{if } \; d_{K(\alpha)}(T_{K(\alpha j)}) = a \;\; \in \mathrm{P}^+(\Sigma), \\ a - \varepsilon(\alpha, \alpha j), & \text{if } \; d_{K(\alpha)}(T_{K(\alpha j)}) = a^- \;\; \in \mathrm{Q}(C) \setminus \mathrm{P}^+(\Sigma). \end{cases}$$

It is not hard to see that

$$T_{K(\alpha i)} \leq_{K(\alpha)} T_{K(\alpha j)} \quad \text{iff} \quad d_1(\alpha, \alpha i) \leq d_1(\alpha, \alpha j) \tag{14}$$

for all $\alpha \in \{1, \ldots, k\}^*$ and $1 \leq i, j \leq k$.

We extend d_1 to the whole tree K as follows. Let $d_1(\alpha, \alpha) = 0$ and, for every node $\alpha j_1 \cdots j_n$,

$$d_1(\alpha, \alpha j_1 \cdots j_n) \;=\; d_1(\alpha, \alpha j_1) + \cdots + d_1(\alpha j_1 \cdots j_{n-1}, \alpha j_1 \cdots j_n).$$

Finally, for every two nodes $\alpha j_1 \cdots j_n$ and $\alpha j'_1 \cdots j'_{n'}$ such that $j_1 \neq j'_1$ we let

$$d_1(\alpha j_1 \cdots j_n, \alpha j'_1 \cdots j'_{n'}) \;=\; d_1(\alpha, \alpha j_1 \cdots j_n) + d_1(\alpha, \alpha j'_1 \cdots j'_{n'}).$$

All the remaining distances are defined by symmetry.

Now we can we define an interpretation $\mathcal{I}_1$ on the space (Δ_1, d_1) by taking, for every concept name $A \in \mathsf{N_C}$ and every object name $\ell \in \mathsf{N_O}$,

$$\begin{aligned} A^{\mathcal{I}_1} &= \{\alpha \in \{1, \ldots, k\}^* \mid A \in T_{K(\alpha)}\}, \\ \ell^{\mathcal{I}_1} &= \{\alpha \in \{1, \ldots, k\}^* \mid \{\ell\} \in T_{K(\alpha)}\}. \end{aligned}$$

Observe that, as before, this interpretation is not 'real' because object names can be interpreted by non-singleton sets, but let us forget about this for the moment.

Lemma 22. *Let $\alpha, \beta \in \{1, \ldots, k\}^*$ and $D \in T_{K(\beta)}$, for some $\exists D \in \mathrm{cl}(C)$. Then either $D \in T_{K(\alpha)}$, or $D \in T_{K(\alpha j)}$ and $d_1(\alpha, \alpha j) \leq d_1(\alpha, \beta)$, for some $j \in \{1, \ldots, k\}$.*

Proof. First, using condition (B7), it is not hard to show by induction on the length of $\alpha \in \{1, \ldots, k\}^*$ that

$$\exists C' \in T_{K(\alpha)} \quad \text{iff} \quad \exists C' \in T_{K(\epsilon)},$$

for all $\exists C' \in \mathrm{cl}(C)$. Therefore $T_{K(\alpha)}$ contains $\exists D$.

If $D \in T_{K(\alpha)}$ then we are done. So suppose that $D \notin T_{K(\alpha)}$. Then $\alpha \neq \beta$ and there is $j \in \{1, \ldots, k\}$ such that $D \in T_{K(\alpha j)}$.

Let $\alpha = \gamma i_1 \dots i_m$ and $\beta = \gamma j_1 \dots j_n$. We proceed by the induction on $m+n$.

$m = 0$, $n = 1$: this means that $\beta = \alpha j_1$, and so we are done again.

$m = 1$, $n = 0$: this means that $\alpha = \beta i_1$, and so $T_{K(\beta)} \in \mathcal{T}_{K(\alpha)}$ by (H3). Take $j \in \{1, \dots, k\}$ such that $T_{K(\alpha j)} = T_{K(\beta)}$. Then $c(\alpha j) = \mathsf{const}$, and hence $d_1(\alpha, \alpha j) \leq d_1(\alpha, \beta)$ by (D1) or (D2), as required.

$m = 1$, $n = 1$: choose $j \in \{1, \dots, k\}$ such that $d_1(\alpha, \alpha j)$ is minimal with $D \in T_{K(\alpha j)}$. Let $a = d_0(\alpha, \alpha j)$, $b = d_0(\gamma, \alpha)$, $c = d_0(\gamma, \beta)$, and r, s, t be such that

$$d_{K(\alpha)}(T_{K(\alpha j)}) \in \{a_r^-, a_r\}, \quad d_{K(\gamma)}(T_{K(\alpha)}) \in \{a_s^-, a_s\}, \quad d_{K(\gamma)}(T_{K(\beta)}) \in \{a_t^-, a_t\}.$$

We need to show that $a \leq b + c$. If $a_r < a_s + a_t$ then we have

$$b + c - a \; > \; (a_s + a_t - a_r) - 2\varepsilon_s - 2\varepsilon_t \; \geq \; \delta - \delta/2 - \delta/2 \; = \; 0.$$

Therefore, we may assume that $a_s + a_t \leq a_r$, in particular $s, t < r$ and $a_s + a_t \in \mathrm{P}^+(C)$. Four cases are possible now: $b < a_s$, $c < a_t$; $b < a_s$, $c = a_t$; $b = a_s$, $c < a_t$; and $b = a_s$, $c = a_t$. As all of them are similar, and we only consider the second one, which is the most demonstrative.

Note first that $d_{K(\alpha)}(T_{K(\gamma)}) \leq a_s^-$ by (H4). Now, since $D \in T_{K(\beta)}$, we have $\exists^{\leq a_t} D \in T_{K(\gamma)}$ by (B6), whence $\exists^{< a_s + a_t} D \in T_{K(\alpha)}$ by (B8) and the observation above. Therefore, $d_{K(\alpha)}(T_{K(\alpha j)}) \leq (a_s + a_t)^-$ by (B5) and the choice of j. Thus, $a_r = a_s + a_t$, $a = a_r^-$, and

$$b + c - a \; > \; -2\varepsilon_s - 2\varepsilon_t + \varepsilon_r \; \geq \; 0,$$

since $\varepsilon_r, \varepsilon_s \leq \varepsilon_r/4$.

The case $m = 2$, $n = 0$ is considered similarly to the previous one, while the case $m = 0$, $n = 2$ remains the same as in the non-symmetric case.

$m > 2$: Let $\alpha' = \gamma i_1 \dots i_{m-1}$. By the induction hypothesis, there exists $j' \in \{1, \dots, k\}$ such that $D \in T_{K(\alpha' j')}$ and $d_1(\alpha', \alpha' j') \leq d_1(\alpha', \beta)$. Again by the induction hypothesis, there exists $j \in \{1, \dots, k\}$ such that $D \in T_{K(\alpha j)}$ and $d_1(\alpha, \alpha j) \leq d_1(\alpha, \alpha' j')$. Hence

$$d_1(\alpha, \alpha j) \leq d_1(\alpha, \alpha') + d_1(\alpha', \alpha' j') \leq d_1(\alpha, \alpha') + d_1(\alpha', \beta) = d_1(\alpha, \beta),$$

by the definition of d_1.

The case $n > 2$ is similar. □

Lemma 23. *For every $D \in \mathrm{cl}(C)$, we have $\alpha \in D^{\mathcal{I}_1}$ iff $D \in T_{K(\alpha)}$*

Proof. The proof is similar to the proof of Lemma 16, and we leave it to the reader. □

Thus, it remains to fix the problem with nominals, which can be done as follows. We apply to the pseudo-interpretation $\mathcal{I}_1$ the 'filtration' procedure described in the proof of Theorem 4. Since, by definition, the min-tree K is labelled by a nominal ready set of bouquets and since the filtration results in a model where each type can be realised only once, we then obtain an $\mathcal{SL}$-model satisfying C where each set of the form $\{\ell\}$ is interpreted by a singleton set. □

A *single complemented pair automaton* $\mathcal{A}$ on infinite k-ary trees is a tuple $(\Sigma, Q, \Gamma, Q_0, F)$, where

- (Σ, Q, Γ, Q_0) is a looping tree automaton as defined above,
- F is a pair of disjoint sets of states from Q; it will be convenient for us to assume that $F = (\mathsf{dec}, \mathsf{inc})$ and $\mathsf{dec}, \mathsf{inc} \subseteq Q$.

$\mathcal{A}$ *accepts* a Σ-labelled tree T over $\{1, \dots, k\}^*$ iff there exists a run R of $\mathcal{A}$ on T such that, for every path $i_0 i_1 \dots$ in T, if $R(i_0 i_1 \dots i_j) \in \mathsf{dec}$ for infinitely many j, then $R(i_0 i_1 \dots i_j) \in \mathsf{inc}$ for infinitely many j as well.

As was shown in [8], the emptiness problem for single complemented pair automata is decidable in polynomial time. We show now how to reduce the satisfiability problem for $\mathcal{SL}$-concepts in symmetric models to the emptiness problem for these automata.

A *coloured C-bouquet* is a pair $(\mathfrak{B}, c)$ where $\mathfrak{B} = (\mathcal{T}_\mathfrak{B}, \leq_\mathfrak{B}, d_\mathfrak{B})$ is a bouquet for C and c is a function from $\mathcal{T}_\mathfrak{B}$ to $\{\mathsf{dec}, \mathsf{inc}, \mathsf{const}\}$.

With every concept C and every nominal ready set B of coloured C-bouquets we associate a single complemented pair automaton $\mathcal{A}_C^B = (\Sigma, Q, \Delta, Q_0, F)$ by taking

- Σ to be the set of types occurring in coloured bouquets of B,
- $Q = B$,
- $Q_0 = \{(\mathfrak{B}, c) \in B \mid C \in T_\mathfrak{B},\ \mathfrak{B} \text{ contains a type with } \ell \text{ for every } \ell \text{ in } \tau\}$,
- $\mathsf{dec} = \{(\mathfrak{B}, c) \in B \mid c(T_\mathfrak{B}) = \mathsf{dec}\}$,
- $\mathsf{inc} = \{(\mathfrak{B}, c) \in B \mid c(T_\mathfrak{B}) = \mathsf{inc}\}$,
- $\big(T, (\mathfrak{B}_0, c_0), (\mathfrak{B}_1, c_1), \dots, (\mathfrak{B}_k, c_k)\big) \in \Gamma$ iff $T_{\mathfrak{B}_0} = T$, $\mathcal{T}_{\mathfrak{B}_0} \setminus \{T_{\mathfrak{B}_0}\} = \{T_{\mathfrak{B}_1}, \dots, T_{\mathfrak{B}_k}\}$, $T_{\mathfrak{B}_0} \in \mathcal{T}_{\mathfrak{B}_i}$, $c_i(T_{\mathfrak{B}_i}) = c_0(T_{\mathfrak{B}_i})$, and $d_{\mathfrak{B}_i}(T_{\mathfrak{B}_0}) \leq d_{\mathfrak{B}_0}(T_{\mathfrak{B}_i})$ for $1 \leq i \leq k$, and for all $T' \in \mathcal{T}_{\mathfrak{B}_i} \setminus \{T_{\mathfrak{B}_i}\}$,
  - $c_i(T') = \mathsf{inc}$ iff $T <_{\mathfrak{B}_i} T'$,
  - $c_i(T') = \mathsf{const}$ iff $T' \sim_{\mathfrak{B}_i} T$,
  - $c_i(T') = \mathsf{dec}$ iff $T' <_{\mathfrak{B}_i} T$.

It follows immediately from Theorem 20 and the given definitions that the runs of $\mathcal{A}_C^B$ on Σ-labelled trees are exactly the B-labelled Hintikka-trees satisfying C.

Lemma 24. *A concept C is satisfiable in a symmetric min-model (with at least two distinct types) iff there exists a nominal ready set B of coloured C-bouquets such that $\mathcal{A}_C^B$ accepts at least one tree.*

As there are only exponentially many different nominal ready sets B of coloured C-bouquets and as $\mathcal{A}_C^B$ is only exponential in $l(C)$, the satisfiability problem in symmetric min-models is decidable in ExpTime.

4.4 Proof of Theorem 3 (ii)

As in the proof of Theorem 3 (iii), we show this result for types over an $\mathcal{SL}$-concept C rather than types over a knowledge base Σ. Suppose that a set $\mathcal{T}$ of types over a concept C is given. We first consider the problem of checking realisability in a min-space satisfying the triangle inequality (but not necessarily symmetric).

Denote by B the set of bouquets for C with types from $\mathcal{T}$. For every $T_0 \in \mathcal{T}$ construct a finite looping automaton $\mathcal{A}_{T_0} = (\Sigma, Q, \Gamma, Q_0)$ as follows:

- Σ is the set $\mathcal{T}$,
- $Q = B$,
- $Q_0 = \{\mathfrak{B} \in B \mid T_0 \in T_{\mathfrak{B}}\}$,
- $\big(T, \mathfrak{B}_0, (\mathfrak{B}_1, \dots, \mathfrak{B}_k)\big) \in \Gamma$ iff $T_{\mathfrak{B}_0} = T$, $\mathcal{T}_{\mathfrak{B}_0} \setminus \{T_{\mathfrak{B}_0}\} = \{T_{\mathfrak{B}_1}, \dots, T_{\mathfrak{B}_k}\}$, and $T_{\mathfrak{B}_0} \in \mathcal{T}_{\mathfrak{B}_i}$, for $1 \le i \le k$.

Now the decision procedure runs as follows:

- Check whether $\mathcal{T}$ is nominal ready in the sense that for each ℓ in C there exists only one $T \in \mathcal{T}$ such that $\{\ell\} \in T$. If this is not case, then $\mathcal{T}$ is not realisable. Otherwise we go to the next step.
- Check whether for each $\exists D \in \mathrm{cl}(C)$, if there exists $T \in \mathcal{T}$ such that $\exists D \in T$, then $\exists D \in T'$ for all $T' \in \mathcal{T}$. If this is not the case, then $\mathcal{T}$ is not realisable. Otherwise we go to the next step.
- Check, for each $T \in \mathcal{T}$, whether (i) T is satisfied in a min-model with one point or (ii) $\mathcal{A}_T$ accepts at least one tree. If (i) or (ii) hold for all $T \in \mathcal{T}$, then $\mathcal{T}$ is realisable. Otherwise it is not realisable.

It follows immediately from our previous results that if this procedure states that $\mathcal{T}$ is not realisable, then $\mathcal{T}$ is not realisable indeed. Moreover, the procedure clearly terminates after exponentially many steps. Now suppose that the procedure claims that $\mathcal{T}$ is realisable. By the definition of $\mathcal{A}_T$, for each $T \in \mathcal{T}$ there exists a min-model $\mathcal{I}_T$ which realises a subset of $\mathcal{T}$ containing T. In view of the finite model property we can assume all these models to be finite.

We first construct a 'disjoint union' $\mathcal{I}$ of the models $\mathcal{I}_T$, $T \in \mathcal{T}$, as follows. Take

$$d \;=\; \max\{d^{\mathcal{I}_T}(u,v) \mid u, v \in \Delta^{\mathcal{I}_T},\ T \in \mathcal{T}\},$$

and then define

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \bigcup \{\Delta^{\mathcal{I}_T} \times \{T\} \mid T \in \mathcal{T}\},\\
d^{\mathcal{I}}\big((u,T),(v,T')\big) &= \begin{cases} d^{\mathcal{I}_T}(u,v), & \text{if } T = T',\\ d+1, & \text{if } T \neq T',\end{cases}\\
\ell^{\mathcal{I}} &= \big\{(\ell^{\mathcal{I}_T}, T) \mid T \in \mathcal{T}\big\}, \quad \text{for every object name } \ell,\\
A^{\mathcal{I}} &= \bigcup \{A^{\mathcal{I}_T} \times \{T\} \mid T \in \mathcal{T}\}, \quad \text{for every concept name } A.
\end{aligned}$$

It is not hard to check that $\mathcal{I}$ is a min-model with (tr) which satisfies exactly the types from $\mathcal{T}$. However, the object names may be interpreted by non-singleton sets in $\mathcal{I}$. Therefore we apply to $\mathcal{I}$ the 'filtration' procedure above to construct the required model $\overline{\overline{\mathcal{I}}}$.

Let us now consider the problem of realisability in metric min-spaces. Denote by B' the set of coloured bouquets with types from $\mathcal{T}$. For every $T_0 \in \mathcal{T}$ construct a single complemented pair automaton $\mathcal{A}'_{T_0} = (\Sigma, Q, \Gamma, Q_0, F)$ as follows:

- Σ is the set $\mathcal{T}$,

- $Q = B'$,
- $Q_0 = \{(\mathfrak{B}, c) \in B' \mid T_0 \in T_{\mathfrak{B}}\}$,
- $\mathsf{dec} = \{(\mathfrak{B}, c) \in B \mid c(T_{\mathfrak{B}}) = \mathsf{dec}\}$,
- $\mathsf{inc} = \{(\mathfrak{B}, c) \in B \mid c(T_{\mathfrak{B}}) = \mathsf{inc}\}$,
- $\big(T, (\mathfrak{B}_0, c_0), (\mathfrak{B}_1, c_1), \ldots, (\mathfrak{B}_k, c_k)\big) \in \Gamma$ iff $T_{\mathfrak{B}_0} = T$, $\mathcal{T}_{\mathfrak{B}_0} \setminus \{T_{\mathfrak{B}_0}\} = \{T_{\mathfrak{B}_1}, \ldots, T_{\mathfrak{B}_k}\}$, $T_{\mathfrak{B}_0} \in \mathcal{T}_{\mathfrak{B}_i}$, $c_i(T_{\mathfrak{B}_i}) = c_0(T_{\mathfrak{B}_i})$, and $d_{\mathfrak{B}_i}(T_{\mathfrak{B}_0}) \leq d_{\mathfrak{B}_0}(T_{\mathfrak{B}_i})$ for $1 \leq i \leq k$, and for all $T' \in \mathcal{T}_{\mathfrak{B}_i} \setminus \{T_{\mathfrak{B}_i}\}$,
  - $c_i(T') = \mathsf{inc}$ iff $T <_{\mathfrak{B}_i} T'$,
  - $c_i(T') = \mathsf{const}$ iff $T' \sim_{\mathfrak{B}_i} T$,
  - $c_i(T') = \mathsf{dec}$ iff $T' <_{\mathfrak{B}_i} T$.

Then the decision procedure is the same as above, but with $\mathcal{A}_T$ replaced by $\mathcal{A}'_T$.

5 Related logics

The operator $\leftleftarrows$ of the fragment $\mathcal{CSL}$ of $\mathcal{SL}$ without the absolute distance operators $\exists^{<a}$ and $\exists^{\leq a}$ is closely related to the 'implication' $>$ of conditional logics. According to Lewis' [18] semantics for conditionals, propositions are interpreted in a set W of possible worlds which come together with orderings $\preceq_w \subseteq W \times W$, for $w \in W$, which can be understood as follows: $w' \preceq_w w''$ if w' is more similar or closer to w than w''. A formula $C > D$ is true at w iff, for every $\preceq_w$-minimal v with $v \models C$, we have $v \models D$. Various authors (see, for example, [7, 29]) have considered the case where the relations $\preceq_w$ are induced by min-spaces (Δ, d) (in conditional logic, the requirement (min) is often called the *limit assumption*) by setting

$$w' \preceq_w w'' \quad \text{iff} \quad d(w, w') \leq d(w, w'').$$

Under this interpretation the operators $\leftleftarrows$ and $>$ have exactly the same expressive power: for every min-model $\mathcal{I} = (\Delta^{\mathcal{I}}, d^{\mathcal{I}}, A_1^{\mathcal{I}}, A_2^{\mathcal{I}}, \ldots)$ we have

$$(A_1 > A_2)^{\mathcal{I}} \;=\; ((A_1 \leftleftarrows (A_1 \sqcap \neg A_2)) \sqcup \forall \neg A_1)^{\mathcal{I}}$$

(recall that the universal modality $\forall$ is definable using $\leftleftarrows$) and, conversely,

$$(A_1 \leftleftarrows A_2)^{\mathcal{I}} \;=\; (((A_1 \sqcup A_2) > A_1) \sqcap (A_1 > \neg A_2) \sqcap \neg(A_1 > \bot))^{\mathcal{I}}.$$

Relations $\preceq_w$ induced by *symmetric* distance spaces have not been considered in the conditional logic literature. According to the classification of [9], our (nominal-free) logic of arbitrary min-spaces corresponds to the conditional logic of frames satisfying the normality, reflexivity, strict centering, uniformity and connectedness condition. In particular, the ExpTime upper bound for the nominal-free fragment of $\mathcal{CSL}$ over arbitrary min-models, was obtained before in [9] in the framework of conditional logic.

The requirement (min) on distance spaces is crucial for the complexity results obtained for $\mathcal{SL}$ and $\mathcal{CSL}$. Observe that in arbitrary metric models the operator $\Box$ defined by taking $\Box C = (\top \leftleftarrows \neg C)$ actually corresponds to the *interior operator* of the induced topology.

Thus, $\mathcal{CSL}$ contains the full logic $\mathbf{S4}_u$ of topological spaces, and so can be used for spatial representation and reasoning (see, e.g., [6]). The topological aspects of $\mathcal{CSL}$ will be considered elsewhere. It will also be shown that even the nominal-free fragment of $\mathcal{CSL}$ is undecidable when interpreted over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$ or their finite subspaces. The proof is by reduction of the decision problem for Diophantine equations (Hilbert's 10th problem).

The integration of description logic and similarity logic presented in this paper is closely related to the well-established construction of adding *concrete domains* to description logics [3, 20, 21, 13]. From the expressiveness point of view, the main difference between our approach and the concrete domain approach is that in our integration the domains of the similarity spaces are *part of* the 'abstract' domain of the description logic, whereas concrete domains are separated from the abstract domain interpreting the description logic concepts and roles. Thus, in contrast to concrete domains we can, for example, attach attributes and properties to elements of the similarity spaces (in the example from the introduction, we introduced the role has_taste_descriptor). The computational behaviour of DLs with concrete domains is in general much worse compared to the results above: even rather weak languages for the concrete domain can result in undecidable logics. Needless to say, however, that the concrete domain approach is considerably more generic and applies to various formalisms which we cannot cover.

Acknowledgements: The work on this paper was partially supported by the U.K. EPSRC research grants GR/S61966/01 and GR/S61973/01.

References

[1] http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook. Theory, Implementation and Applications.* Cambridge University Press, 2003.

[3] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI-91*, pages 452–457, Sydney (Australia), 1991.

[4] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[5] A. Chagrov and M. Zakharyaschev. *Modal Logic.* Oxford University Press, Oxford, 1997.

[6] A. Cohn and S. Hazarika. Qualitative spatial representation and reasoning: an overview. *Fundamenta Informaticae*, 43:2–32, 2001.

[7] J. P. Delgrande. Preliminary considerations on the modelling of belief change operators by metric spaces. In *NMR*, pages 118–125, 2004.

[8] E. Emerson and C. Jutla. The complexity of tree automata and logics of programs. *Siam Journal of Computing*, 29:132–158, 1999.

[9] N. Friedman and J. Halpern. On the complexity of conditional logics. In *Proceedings of KR'94*, pages 202–213, 1994.

[10] P. Gärdenfors. How to make the Semantic Web more semantic. In A.C. Varzi and L. Vieu, editors, *Formal Ontology in Information Systems*, pages 19–36. IOS Press, 2004.

[11] S. Ghilardi and L. Santocanale. Algebraic and model theoretic techniques for fusion decidability in modal logics. In *Proceedings of LPAR 2003*, volume 2850 of *LNAI*, pages 152–166. Springer, 2003.

[12] V. Goranko and S. Passy. Using the universal modality. *Journal of Logic and Computation*, 2:203–233, 1992.

[13] V. Haarslev, C. Lutz, and R. Möller. Foundations of spatioterminological reasoning with description logics. In G. Cohn, L.K. Schubert, and S.C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*, pages 112–123. Morgan-Kaufmann, 1998.

[14] V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR'01)*, number 2083 in Lecture Notes in Artifical Intelligence, pages 701–705. Springer-Verlag, 2001.

[15] I. Horrocks. The Fact++ system. `http://owl.man.ac.uk/factplusplus`.

[16] I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453, 2005.

[17] O. Kutz, H. Sturm, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev. Logics of metric spaces. *ACM Transactions on Computational Logic*, 4:260–294, 2003.

[18] D. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.

[19] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 863–872. Society for Industrial and Applied Mathematics, 2003.

[20] C. Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen, 2002.

[21] C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*, pages 265–296. King's College Publications, 2003.

[22] C. Lutz, F. Wolter, and M. Zakharyaschev. A tableau algorithm for reasoning about concepts and similarity. In M. C. Mayer and F. Pirri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 2796 of *LNCS*, pages 134–149. Springer, 2003.

[23] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.

[24] B. Parsia and E. Sirin. Pellet: An OWL DL-reasoner. In *Proceedings of the International Workshop on Description Logics*, 2004.

[25] V. Pratt. Models of program logic. In *Proceedings of the 20th Annual Symposion on the Foundations of Computer Science*, pages 115–122, 1979.

[26] I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14:369–395, 2005.

[27] Ph. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, 11:95–130, 1999.

[28] W. Sayers. *An Introduction to Library Classification.* Grafton & Co, London, 1955.

[29] K. Schlechta. *Coherent Systems.* Elsevier, 2004.

[30] M. Sheremet, D. Tishkovsky, F. Wolter, and M. Zakharyaschev. Comparative similarity, tree automata, and Diophantine equations. In *Proceedings of LPAR 2005*, volume 3835 of *LNAI*, pages 651–665. Springer, 2005.

[31] E. Spaan. *Complexity of Modal Logics.* PhD thesis, University of Amsterdam, 1993.

[32] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, pages 133–191. Elsevier, 1990.

[33] S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research*, 12:199–217, May 2000.

[34] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[35] M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.

[36] D. Wishart. *Whisky Classified. Choosing single malts by flavour.* Pavilion Books, London, 2002.

[37] F. Wolter and M. Zakharyaschev. Reasoning about distances. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 1275–1280. Morgan Kaufmann, 2003.

[38] F. Wolter and M. Zakharyaschev. A logic for metric and topology. *Journal of Symbolic Logic*, 70:795–828, 2005.