‘Closer’ representation and reasoning

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

[1]Department of Computer Science
King’s College London
Strand, London WC2R 2LS, U.K.
{mikhail,mz}@dcs.kcl.ac.uk

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

Abstract

We argue that orthodox tools for defining concepts in the framework of description logic should often be augmented with constructors that could allow definitions in terms of similarity (or closeness). We present a corresponding logical formalism with the binary operator ‘more similar or closer to X than to Y’ and investigate its computational behaviour in different distance (or similarity) spaces. The concept satisfiability problem turns out to be ExpTime-complete for many classes of distances spaces no matter whether they are required to be symmetric and/or satisfy the triangle inequality. Moreover, the complexity remains the same if we extend the language with the operators ‘somewhere in the neighbourhood of radius a’ where a is a non-negative rational number. However, for various natural subspaces of the real line $\mathbb{R}$ (and Euclidean spaces of higher dimensions) even the similarity logic with the sole ‘closer’ operator turns out to be undecidable. This quite unexpected result is proved by reduction of the solvability problem for Diophantine equations (Hilbert’s 10th problem).

> “There is nothing more basic to thought and language than our sense of similarity; our sorting of things into kinds.”
> (Quine 1969)

1 Introduction

How do we define concepts?

In description logic, we do this by establishing relationships between concepts, for example,

$$\mathsf{Mother} \quad \equiv \quad \mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Person}$$

The main tool for analysing and using such definitions is *reasoning*.

In areas such as image processing, data mining, case-based reasoning (and our everyday life as well) we often define concepts using various (explicit or

implicit) similarity measures: for example,

$$\mathsf{Reddish} \quad \equiv \quad \{\mathit{Red}\} \leftleftarrows \{\mathit{Green}, \ldots, \mathit{Black}\}$$

which reads: 'a colour is reddish iff it is more similar (with respect to the RGB, HSL or some other explicit or implicit colour model) to the prototypical colour *Red* than to the prototypical colours *Green*, ..., *Black*.' The main tools for dealing with concepts of this sort are *numerical computations* (say, with the help of Voronoi tessellations, nearest neighbour or clustering algorithms).

As bioinformatics, linguistics and similar areas use both ways of concept formation, we are facing the problem of integrating these two types of representation. Although there is a lot of research concerned with the derivation of a similarity measure from an ontology [9, 7], an integration of similarity- and DL-based formalisms to define concepts (or formulate constraints on their relations) and reason about them has so far been proposed only in [8].

The main goal of our ongoing research project is to develop, investigate and implement uniform logic-based formalisms capable of representing and reasoning with both terminological and similarity-based knowledge in their interaction.

In [5, 14, 8, 15] we presented and investigated rudimentary DL-like formalisms for reasoning about concepts and similarity with concept constructors of the form $\exists^{<a}C$, that is, 'in the a-neighbourhood of C,' where $a \in \mathbb{Q}^+$. In particular, we showed that reasoning with such formalisms can be organised with the help of tableaux, and that the typical worst case complexity is ExpTime.

The apparent limitation of these languages is that they can only operate with *concrete* degrees of similarity $a \in \mathbb{Q}^+$. Although qualitative similarity measures such as 'A is very close to B' and 'B is far from C, and medium close to D' can be encoded with concrete numbers, in many cases similarities can only be defined implicitly using statements like 'X resembles Y more than Z.'

The logic we propose in this paper extends any 'reasonably well-behaved' description logic with finitely many binary 'closer' operators $\leftleftarrows_i$ and unary distance operators $\exists_i^{<a}, \exists_i^{\leq a}$, for $a \in \mathbb{Q}^+$ (to capture different similarity measures).

The applications of the logic we envisage are similar to the use of description logic 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.). At the moment we do not assume any strong interaction axioms between the DL constructors and the similarity operators. That is why our primary goal in this paper is to investigate the computational properties of the pure *similarity logic* $\mathcal{SL}$ which contains concept names, nominals, the Booleans, and the similarity operators mentioned above. We interpret $\mathcal{SL}$ in models based on (finite) metric spaces. However, as similarity measures are not always symmetric and do not necessarily satisfy the triangle inequality (see, e.g., [12]), we also consider 'distance spaces' without one or both of these properties.

Our first main result is that the concept satisfiability problem (relative to arbitrary knowledge bases) over such classes of finite models is decidable in ExpTime, even if the numerical parameters are coded in binary. This upper bound is obtained using a reduction to the emptiness problem for tree automata with one complemented pair [2]. We show that for all these classes of models concept satisfiability is ExpTime-hard even for the 'qualitative' language $\mathcal{SL}^{\mathsf{q}}$ with the Booleans and the sole operator $\leftleftarrows$.

Similarity measures based on physical measurements (e.g., weight, length or colour) often form subspaces of the Euclidean $\mathbb{R}^n$ for some $n > 0$. It was a great surprise for us to discover that the satisfiability problem for $\mathcal{SL}^{\mathsf{q}}$ in finite subspaces of $\mathbb{R}$ (or $\mathbb{N}$) turns out to be undecidable. It is also undecidable in finite subspaces of $\mathbb{R}^n$ for each $n > 1$. The proof is by reduction of the solvability problem for Diophantine equations (Hilbert's 10th problem).

All proofs in this paper are only sketched; for a detailed exposition the reader is referred to [10].

2 Syntax and semantics

Language. We extend the language of the description logic we are interested in, say, $\mathcal{ALCQO}$ (containing object names $\ell_1, \dots$, concept names $A_1, \dots$, and role names $R_1, \dots$) with the following concept formation rules, where $a \in \mathbb{Q}^+$, $i = 1, \dots, n$, and $D_1, \dots, D_n$ are certain concepts (which are supposed to be modelled by sets of objects with similarity measures):

$$C \quad ::= \quad C_1 \leftleftarrows_i C_2 \quad | \quad \exists_i^{<a} C \quad | \quad \exists_i^{\leq a} C$$

Intended models of this language are structures of the form

$$\mathfrak{I} \;=\; \big\langle \Delta^{\mathfrak{I}}, \ell_1^{\mathfrak{I}}, \dots, A_1^{\mathfrak{I}}, \dots, R_1^{\mathfrak{I}}, \dots, (D_1^{\mathfrak{I}}, d_1^{\mathfrak{I}}), \dots, (D_n^{\mathfrak{I}}, d_n^{\mathfrak{I}}) \big\rangle \tag{1}$$

where the $D_i^{\mathfrak{I}} \subseteq \Delta^{\mathfrak{I}}$ are interpretations of the D_i above and the $d_i^{\mathfrak{I}}$ are *similarity measures* on $D_i^{\mathfrak{I}}$, that is, the $(D_i^{\mathfrak{I}}, d_i^{\mathfrak{I}})$ with $d_i^{\mathfrak{I}} : D_i^{\mathfrak{I}} \times D_i^{\mathfrak{I}} \to \mathbb{R}^+$ are 'distance spaces' of the appropriate type (a precise definition will be given below). The interpretation of $\mathcal{ALCQO}$ in $\mathfrak{I}$ is standard, while the 'similarity' part of the language is interpreted as follows. For $x \in D_i^{\mathfrak{I}}$ and nonempty $Y \subseteq D_i^{\mathfrak{I}}$, let $d_i^{\mathfrak{I}}(x, Y) = \inf\{d_i^{\mathfrak{I}}(x, y) \mid y \in Y\}$. As usual, $d_i^{\mathfrak{I}}(x, \emptyset) = \infty$ and $a < \infty$ for all $a \in \mathbb{R}$. Given a concept C, let $C_i^{\mathfrak{I}} = C^{\mathfrak{I}} \cap D_i^{\mathfrak{I}}$, for $i = 1, \dots, n$. Now we set

$$(C \leftleftarrows_i E)^{\mathfrak{I}} \;=\; \{x \in D_i^{\mathfrak{I}} \mid d_i^{\mathfrak{I}}(x, C_i^{\mathfrak{I}}) < d_i^{\mathfrak{I}}(x, E_i^{\mathfrak{I}})\} \tag{2}$$

$$(\exists_i^{<a} C)^{\mathfrak{I}} \;=\; \{x \in D_i^{\mathfrak{I}} \mid \exists y \in C_i^{\mathfrak{I}}\ d_i^{\mathfrak{I}}(x, y) < a\} \tag{3}$$

$$(\exists_i^{\leq a} C)^{\mathfrak{I}} \;=\; \{x \in D_i^{\mathfrak{I}} \mid \exists y \in C_i^{\mathfrak{I}}\ d_i^{\mathfrak{I}}(x, y) \leq a\} \tag{4}$$

In other words, $C \leftleftarrows_i E$ is the concept containing those objects of D_i that are more similar to C than to E. $\exists_i^{<a} C$ is the (open) a-neighbourhood of C in D_i.

A *knowledge base* consists of (finitely many) concept and role assertions $C(\ell)$ and $\ell R_j \ell'$, and terminological axioms $C \sqsubseteq D$. Notice that we can express assertions like 'the distance between ℓ_1 and ℓ_2 is a' by $\{\ell_1\} \sqsubseteq \neg\exists^{<a}\{\ell_2\} \sqcap \exists^{\leq a}\{\ell_2\}$. Our main reasoning task is *concept satisfiability* with respect to knowledge bases, or, more precisely, given a knowledge base Σ and a concept C, determine whether there exists a model $\mathfrak{J}$ such that $\mathfrak{J} \models \Sigma$ and $C^{\mathfrak{J}} \neq \emptyset$.

Example. We illustrate possible applications of the resulting logic by outlining some (raw) ideas of how a (suitable) DL extended with the similarity operators above can be used to help building ontologies for historical linguistics. Historical (or comparative) linguistics studies languages and their change over time with the aim of establishing and classifying genetic relationships between the world's languages and reconstructing their historic development. For example, Latin of the Roman Empire gave rise to the family of Romance languages which includes French, Spanish, Portuguese, Italian, and Romanian. *Genetically related* languages are defined to be the languages with a common ancestor.

Comparative linguistics uses a number of methods to establish genetic relationships between languages and to speculate on how the genetic linguistic tree (or forest) can look like. The methods differ in their conclusive power and accuracy, in how much human or computational efforts they require, in the amount and availability of the prerequisite linguistic data, etc. For the purpose of our presentation, we divide these methods into three groups:

1. The comparative method defines strict criteria for languages to be genetically related (regular correspondences between languages that ideally lead to the reconstruction of the ancestor language); it is usually regarded to be the most conclusive. The shared innovations method generally supplies valuable additional information that might help to make the genetic tree more accurate (see, e.g., [13]).
2. Methods based on 'genetically relevant' similarity measures such as lexicostatistics, etymostatistics, etc.
3. Less reliable statistical and alignment techniques (the method of resemblances and mass comparison, etc.) can provide useful conjectures or approximations to be justified or rejected by other methods (see, e.g., [6]).

To give some hint of the scale of the problem, we remind the reader that linguistics deals with about 6000 (living and dead) languages which are divided in about 300 families. This complexity as well as the limited knowledge of certain languages (not only ancient but also the existing ones) prevent linguists from agreeing on a single *canonical* genetic tree of languages. New data and methods cause changes, doubts and debates. For example, according to [13] the number of families may be reduced to as low as 200.

It seems that the situation in historical linguistics is similar to that in bioinformatics, where the research community has recognised the need to develop

bioinformatics ontologies in order to exploit vast amount of biological information. Moreover, the idea of using logical formalisms such as description logic (or Prolog) as the underlying ontology languages to cope with difficulties in consistency and maintenance is becoming more and more popular [11].

To build a formal ontology for historical linguistics, one can start with the data provided by 1. and represent the resulting genetic forest of languages, their properties and relations in some suitable DL. For example, we may have (in a 'slang' DL)

- *Latin* parent *French* says that Latin is an immediate predecessor of French;
- $\mathsf{IE} = \exists \mathsf{parent}^{-*}.\{\mathit{ProtoIndoEuropean}\}$ says that the family of Indo-European languages is the subtree with Proto-Indo-European language as the root;
- $L_1 \in C \wedge L_2 \in C \to L_1(\mathsf{parent} \cup \mathsf{parent}^{-})^* L_2$ claims that languages sharing some 'good' property C are connected in the genetic forest.

Methods 2. and 3. provide us with knowledge based on some similarity measures, say, a genetically relevant g from 3. and a less accurate d from 4. In this case we may need to represent both pure metric data (e.g., $d(L_1, L_2) = 0.7$) and our conjectures connecting the metric with properties of languages. E.g.,

- $(C_1 \sqsubseteq \neg \exists_g^{<\varepsilon} C_2) \to (C_1 \sqsubseteq \neg \exists (\mathsf{parent} \cup \mathsf{parent}^{-})^* C_2)$ means that languages from distant classes C_1 and C_2 cannot be genetically related (they belong to disjoint trees in the forest);
- $C \sqsubseteq \exists_g^{<a} L$ or $\exists_g^{<b} C \sqsubseteq D$ say that the whole class C (containing L) should not be too far from L, and, on the other hand, all languages sufficiently close to C must be in some family D;
- $C \sqsubseteq \{L_1, \ldots, L_n\} \leftleftarrows_d D$ represents the conjecture that a family C is closer to prototypical languages $L_1, \ldots, L_n$ then to some family D.

Combining description and similarity logics. We can easily reduce reasoning in the language above to reasoning in the *fusion* of the underlying description logic (say $\mathcal{ALCQO}$) and the similarity logics induced by the distance spaces [1, 4]. Therefore, as long as we have either no nominals in any of the components or nominals in each of the components of the fusion, decidability of the full language follows from the decidability of its components. Moreover, we can even obtain the ExpTime upper bounds for the full logic defined above by extending the proof sketched below for the pure similarity language. In the remaining part of the paper we only focus on this similarity part of the language.

The logic of similarity $\mathcal{SL}$ is defined by taking

$$C \quad ::= \quad \{\ell\} \quad | \quad A \quad | \quad \neg C \quad | \quad C_1 \sqcap C_2 \quad | \quad C_1 \leftleftarrows C_2 \quad | \quad \exists^{<a} C \quad | \quad \exists^{\leq a} C$$

where the ℓ are *object names* and $a \in \mathbb{Q}^+$. $\mathcal{SL}$ is interpreted in the appropriate reducts of the models (1), namely, in the structures of the form

$$\mathfrak{I} \;=\; \left\langle \Delta^{\mathfrak{I}}, d^{\mathfrak{I}}, \ell_1^{\mathfrak{I}}, \ldots, A_1^{\mathfrak{I}}, \ldots \right\rangle,$$

where $\mathfrak{D} = \left\langle \Delta^{\mathfrak{I}}, d^{\mathfrak{I}} \right\rangle$ is a *distance space*, i.e., $d^{\mathfrak{I}}$ is a map from $\Delta^{\mathfrak{I}} \times \Delta^{\mathfrak{I}}$ to $\mathbb{R}^+$ such that, for all $x, y \in \Delta^{\mathfrak{I}}$, $d^{\mathfrak{I}}(x, y) = 0$ iff $x = y$. Such structures will be called $\mathcal{SL}$*-models*. If $d^{\mathfrak{I}}$ satisfies two additional properties

$$d^{\mathfrak{I}}(x, y) \;=\; d^{\mathfrak{I}}(y, x) \qquad (sym)$$

$$d^{\mathfrak{I}}(x, z) \;\leq\; d^{\mathfrak{I}}(x, y) + d^{\mathfrak{I}}(y, z) \qquad (tr)$$

then $\mathfrak{D}$ is called a *metric space*. The interpretation of the Boolean operators in $\mathfrak{I}$ is as usual (we will use $\sqcup$ as a standard abbreviation), $\{\ell\}^{\mathfrak{I}} = \{\ell^{\mathfrak{I}}\}$, and the similarity operators $\leftrightarrows$, $\exists^{<a}$, $\exists^{\leq a}$ are interpreted in the same way as in (2)–(4).

Having in mind applications mentioned in the introduction, it would be natural to consider the $\mathcal{SL}$-concept satisfiability problem in various classes of *finite* $\mathcal{SL}$-models. On the other hand, the DL component of the combined language does not necessarily have the finite model property. That is why our intended models in this paper are based on natural generalisations of finite distance spaces which can be defined as follows. Let $\mathfrak{D} = (\Delta, d)$ be a distance space. The *distance* $d(X, Y)$ between two nonempty $X, Y \subseteq \Delta$ is defined by taking

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

We call $\mathfrak{D}$ a *min-space* if, for all nonempty $X, Y \subseteq \Delta$,

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

We will see in the next section that actually the similarity logic $\mathcal{SL}$ *does not* feel the difference between finite and min-distance spaces. Note that this is not the case for the class of all metric spaces where $\mathcal{SL}$ can express the interior and closure operators induced by the metric: $\mathsf{Int}C ::= \top \leftrightarrows \neg C$ (where $\top$ is the whole space), $\mathsf{Cl}C ::= \neg(C \leftrightarrows \top) \sqcap \neg(\top \leftrightarrows C)$. Reasoning in the class of arbitrary metric spaces requires a completely different approach and will be considered elsewhere. (It is worth noting that in the literature on conditional logic requirement (min) above is often called the *limit assumption*.)

3 Concept satisfiability

We investigate the $\mathcal{SL}$-concept satisfiability problem in various classes of $\mathcal{SL}$-models based on min-spaces. To begin with, note that the concept satisfiability problem with respect to knowledge bases is easily reducible to pure concept

satisfiability (with empty knowledge base). Indeed, let $\forall C = \neg\exists\neg C$, where $\exists D = D \leftleftarrows \bot$ (that is, $\forall$ and $\exists$ are the *universal modalities* over the space). Then a concept C is satisfiable relative to a knowledge base Σ iff the concept $\forall\bigsqcap\{\neg C_1 \sqcup C_2 \mid C_1 \sqsubseteq C_2 \in \Sigma\} \sqcap C$ is satisfiable. In what follows we confine ourselves to investigating concept satisfiability with empty knowledge base.

Is easy to show that satisfiability of $\mathcal{SL}$-concepts depends on whether we assume (sym) and/or (tr). For the 'purely qualitative' fragment $\mathcal{SL}^{\mathsf{q}}$ of $\mathcal{SL}$ which does not contain numerical operators $\exists^{<a}$ and $\exists^{\leq a}$ the situation is different:

Proposition 1. *$\mathcal{SL}^{\mathsf{q}}$-concepts cannot distinguish between $\mathcal{SL}^{\mathsf{q}}$-models with and without (tr).*

Proof. Suppose C is satisfied in a model $\mathfrak{J}$ without (tr). Take any strictly monotonic $f : \mathbb{R}^+ \to (9, 10)$, where $(9, 10)$ is the open interval from 9 to 10. Define a new distance $d^{\mathfrak{J}'}$ on $\Delta^{\mathfrak{J}}$ by $d^{\mathfrak{J}'}(x, y) = f(d^{\mathfrak{J}}(x, y))$, for $x \neq y$. The remaining components of $\mathfrak{J}'$ coincide with those of $\mathfrak{J}$. Then C is satisfied in $\mathfrak{J}'$ and $\mathfrak{J}'$ satisfies (tr). □

Note, however, that $\mathcal{SL}^{\mathsf{q}}$ can distinguish between models with and without (sym). Consider the knowledge base Σ which consists of the inclusions

$$A \sqsubseteq (B \leftleftarrows C), \quad B \sqsubseteq (C \leftleftarrows A), \quad C \sqsubseteq (A \leftleftarrows B). \tag{5}$$

Then, relative to Σ, A is satisfiable in a three-point model without (sym). However, it is not satisfiable in any model satisfying (sym).

Proposition 2 (finite model property). *Let $\mathcal{C}$ be the class of all min-models satisfying any combination of the properties (sym) and (tr), in particular, neither of them. Then an $\mathcal{SL}$-concept is satisfiable in $\mathcal{C}$ iff it is satisfiable in a finite model from $\mathcal{C}$.*

Proof. This result can be proved by a (rather involved) filtration argument. □

In view of this proposition, from now on we will be considering—unless otherwise stated—only *finite* $\mathcal{SL}$-models.

Proposition 3 (lower bound). *Let $\mathcal{C}$ be any class of models mentioned in Proposition 2. Then the satisfiability problem for (nominal-free) $\mathcal{SL}^{\mathsf{q}}$-concepts in $\mathcal{C}$ is ExpTime-hard.*

Proof. The proof is by reduction of the following ExpTime-complete problem: given $\mathcal{ALC}$-concepts C and D with a single role R, decide whether $D \equiv \top$ follows from the TBox $\{C \equiv \top\}$. Let $\kappa_0 = B_0$, $\kappa_1 = \neg B_0 \sqcap B_1$, $\kappa_2 = \neg B_0 \sqcap \neg B_1$, for some fresh B_i. Define inductively a translation $\cdot^{\#}$ from $\mathcal{ALC}$ to $\mathcal{SL}^{\mathsf{q}}$ by taking: $A_i^{\#} = A_i$, $(\neg C_1)^{\#} = \neg C_1^{\#}$, $(C_1 \sqcap C_2)^{\#} = C_1^{\#} \sqcap C_2^{\#}$, and

$$(\exists R.E)^{\#} \;=\; \bigsqcup_{i<3}\Big(\kappa_i \sqcap \exists\kappa_{i\oplus 1} \sqcap \big((\kappa_{i\oplus 1} \sqcap E^{\#}) \leftrightarrows \kappa_{i\oplus 1}\big)\Big),$$

where $\oplus$ is addition modulo 3 and $\leftrightarrows$ means 'at the same distance,' i.e.,

$$C_1 \leftrightarrows C_2 \quad = \quad \neg(C_1 \leftleftarrows C_2) \sqcap \neg(C_2 \leftleftarrows C_1).$$

One can show that $D \equiv \top$ follows from $\{C \equiv \top\}$ iff $\forall C^{\#} \to D^{\#}$ is valid in all finite $\mathcal{SL}$-models (with and without (sym) and/or (tr)). $\square$

Proposition 4 (upper bound). *The $\mathcal{SL}$-concept satisfiability in any class $\mathcal{C}$ of models from Proposition 2 is decidable in ExpTime, even if the numerical parameters are coded in binary.*

Proof. We only give a brief sketch of how the proof works for finite metric spaces. The crucial idea is that a concept C is satisfiable in a finite metric space iff it is satisfiable in an 'abstract tree metric space' (where nominals can be interpreted by non-singleton sets whose members satisfy the same concepts from a closure $cl(C)$ of the set of subconcepts of C under certain rules) satisfying certain properties. Namely, the abstract tree metric space has the domain $\Delta = \{1, \ldots, k\}^*$ (the set of finite words over $\{1, \ldots, k\}$, where k is a natural number which is polynomial in the size of C), its distances $d(\alpha, \alpha i)$, $\alpha \in \Delta$, $i = 1, \ldots, k$, are from a previously specified ordered set of abstract distances (of size exponential in C) which encode constraints on proper distances, and the abstract distance between non-successor nodes x and y in Δ is computed as the sum of the distances over the shortest path from x to y in Δ. (See [8] for that part of the encoding of distances which takes care of the operators with parameters. Additional constraints are required to deal with $\leftleftarrows$; consult [10] for details.) Moreover, the abstract tree metric space is not allowed to contain an infinite sequence $d(\alpha, \alpha i) \geq d(\alpha i, \alpha ij) \geq d(\alpha ij, \alpha ijh) \geq \cdots$ where infinitely often $\geq$ is actually $>$. This condition is required to ensure that the encoded metric space is a min-space. For example, the concept A from (5) would be satisfiable (relative to Σ) in a space where there is an infinite sequence as above with all $\geq$ replaced by $>$.

Now, one can prove by an unravelling argument that any concept satisfiable in a finite metric space is also satisfiable in such an abstract tree metric space. The converse direction can be proved by a (quite involved) filtration argument: one can show that a concept C satisfiable in a discrete tree metric space $\mathfrak{J}$ as described above is satisfiable in a finite metric spaces whose domain consists of $\{[\alpha] \mid \alpha \in \Delta\}$, where $[\alpha]$ denotes the equivalence class of α relative to the relation $\sim$ defined by $\alpha \sim \alpha'$ iff for all $D \in cl(C)$, $\alpha \in D^{\mathfrak{J}}$ iff $\alpha' \in D^{\mathfrak{J}}$. (Notice that after the filtration nominals are interpreted by singletons.) Thus, it remains to show that satisfiability in abstract tree metric spaces of the form above can be decided in exponential time.

This can be done by reducing satisfiability of C to the emptiness problem for a tree automaton $\mathcal{A}_C$ with an acceptance condition consisting of one complemented pair $(\mathsf{red}, \mathsf{green})$. (This condition means that in an accepting run every

path with infinitely many red states must have infinitely many green states.) This problem is decidable in polynomial time; see, e.g., [2]. As the automaton $\mathcal{A}_C$ is exponential in the size of C, we obtain an exponential upper bound for the satisfiability problem. The acceptance condition is used to ensure that accepted trees are represent min-spaces: roughly, a state is red if the distance is decreasing, and green if the distance is increasing. □

It turns out, however, that for $\mathcal{SL}$-models based on subspaces of $\mathbb{R}^n$, for each $n > 0$, in particular $n = 1$, the satisfiability problem becomes undecidable, even for the language without numerical operators:

Proposition 5 (undecidability). *For each $n > 0$, the satisfiability of $\mathcal{SL}^{\mathrm{q}}$-concepts is undecidable in the class of finite models and the class of min-models based on subspaces of $\mathbb{R}^n$, or only $\mathbb{Z}^n$.*

Proof. The proof proceeds by reduction of the solvability problem for Diophantine equations. Here is a brief sketch; see [10] for details. Observe first that we can always deal with models based on *one-dimensional* spaces. Indeed, let $\mathfrak{I}$ be based on $\mathbb{R}^n$. Then, for nominals ℓ_0 and ℓ_1, the term $(\{\ell_0\} \leftrightarrows \{\ell_1\}) \sqcap \forall\neg(\{\ell_0\} \sqcap \{\ell_1\})$, if satisfiable, defines an affine subspace of dimension $n - 1$. By iterating this construction we can reduce dimension to 1.

Let us now focus on the class $\mathcal{R}$ of min-models based on subspaces of $\mathbb{R}$. The proof involves three main steps:

(1) ensure that our model is based on a space similar to $\mathbb{Z}$;

(2) define in this model sets of the form $\{lk + j \mid k \in \mathbb{Z}\}$—they can be used to encode the number l;

(3) encode addition and multiplication on such sets.

For (1) we take the concept

$$\mathsf{Base}(\mathbf{A}) \;=\; \forall \bigsqcap_{i<3} \big(A_i \;\to\; \neg A_{i\oplus 1} \sqcap (A_{i\ominus 1} \leftrightarrows A_{i\oplus 1})\big),$$

where $\oplus$ and $\ominus$ denote addition and subtraction modulo 3. Then a model $\mathfrak{I} \in \mathcal{R}$ satisfies $\mathsf{Base}(\mathbf{A})$ iff $\mathfrak{I}$ coincides (modulo an affine transformation) with a model $\mathfrak{Z}$ such that $\Delta^{\mathfrak{Z}} = \mathbb{Z}$ and $A_i^{\mathfrak{Z}} = \{3k + i \mid k \in \mathbb{Z}\}$, $i < 3$.

The following analogues of the 'next-time' operator and its inverse can simulate the functions '+1' and '−1':

$$\bigcirc C \;=\; \bigsqcup_{i<3} (A_i \sqcap (A_{i\oplus 1} \leftrightarrows A_{i\oplus 1} \sqcap C)), \quad \bigcirc^{-1} C \;=\; \bigsqcup_{i<3} (A_i \sqcap (A_{i\ominus 1} \leftrightarrows A_{i\ominus 1} \sqcap C)).$$

To fix an origin and an orientation for our model we take a fresh A and consider the term $\exists(A_2 \sqcap \neg A \sqcap \bigcirc A) \sqcap \forall(A \to \bigcirc A)$, which is satisfied in a model $\mathfrak{Z}$ of

the above form iff $A^{\mathfrak{Z}} = \{k, k+1, \dots\}$ for some $k \in \mathbb{Z}$, $k \equiv 0 (\mathrm{mod}\, 3)$. Assume further that $A^{\mathfrak{Z}} = \mathbb{N}$. Then $\mathsf{Zero} = A \sqcap \bigcirc^{-1} \neg A$ defines $\{0\}$. For (2), we define

$$\mathsf{Seq}(\mathbf{B}) \;=\; \forall \bigsqcap_{i<3} \big(B_i \to (B_{i\oplus 1} \leftrightarrows B_{i\ominus 1})\big) \;\sqcap\; \exists \big(B_0 \sqcap A \sqcap (B_2 \leftleftarrows B_2 \sqcap A)\big),$$

which is satisfied in $\mathfrak{Z}$ iff $B_i^{\mathfrak{Z}} = \{lk + j \mid k \equiv i (\mathrm{mod}\, 3)\}$, $i = 0, 1, 2$, for some $j < l$ in $\mathbb{N}$. Recall that a set of the form $\{lk + j \mid k \in \mathbb{Z}\}$ is used to encode the number l. Clearly, two sets of this form encode the same number iff they either coincide or are disjoint. Note that to encode a particular number n we can use the term $\mathsf{Seq}(\mathbf{B}) \sqcap \forall(\mathsf{Zero} \to (B_1 \leftrightarrows \bigcirc^{-n} \mathsf{Zero}))$.

For (3), let $U = \{ku \mid k \in \mathbb{Z}\}$, $V = \{kv \mid k \in \mathbb{Z}\}$ and $W = \{kw \mid k \in \mathbb{Z}\}$ represent the arguments and result of an operation. To encode addition, take an auxiliary set $V' = \{kv' + u' \mid k \in \mathbb{Z}\}$ and state that:

- the sets V and V' encode the same number, i.e., $v = v'$;
- the distances from $\{0\}$ to the sets $\{x \in U \mid x > 0\}$ and $\{x \in V' \mid x > 0\}$ coincide, i.e., $u = u'$;
- the distances from $\{0\}$ to the sets $\{x \in W \mid x > 0\}$ and $\{x \in V' \mid x > u'\}$ coincide, i.e., $u = u' + v' = u + v$.

To encode multiplication, we use the following fact

Fact 6. *Let $0 < u < v$ be integer numbers. Then*

(*i*) *$x = uv$ is the least solution of $x \equiv 0 (\mathrm{mod}\, b) \wedge x \equiv u (\mathrm{mod}\, (v-1)) \wedge x > 0$.*

(*ii*) *$x = v^2$ is the least solution of $x \equiv 0 (\mathrm{mod}\, v) \wedge x \equiv 1 (\mathrm{mod}\, (v-1)) \wedge x > v$.*

If $u < v$, we take sets $V' = \{kv' + u' \mid k \in \mathbb{Z}\}$, $V'' = \{kv'' \mid k \in \mathbb{Z}\}$ and state that:

- $u = u'$, $v = v'$, and and the distances from $\{0\}$ to the sets $\{x \in V'' \mid x > 0\}$ and $\{x - 1 \mid x \in V,\ x > 0\}$ coincide, i.e., $v'' = v' = v - 1$;
- the distances from $\{0\}$ to $\{x \in U \cap V'' \mid x > 0\}$ and $\{x \in W \mid x > 0\}$ coincide, i.e., $w = uv$, according to Fact 6 (*i*).

The case $u > v$ is dealt with by symmetry; and for the case $u = v$ we use a similar construction by applying Fact 6 (*ii*).

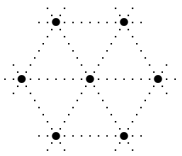
By iterating this constructions, we can encode any sequence of arithmetical operations, that is, a computation of any polynomial.

In the finite case we modify the terms $\mathsf{Base}(\mathbf{A})$ and $\mathsf{Seq}(\mathbf{B})$ to take care of the endpoints. And to ensure that two encoding sets represent the same number we require that they either coincide or strictly alternate and are sufficiently long. □

The most surprising part of the result above is the case $n = 1$. For $n = 2$ one can prove the result using a less 'heavy artillery' than Diophantine equations. For example, one can prove undecidability of satisfiability in min-subspaces of $\mathbb{R}^2$ even for nominal-free $\mathcal{SL}^{\mathfrak{q}}$-concepts by reduction of the undecidable $\mathbb{Z} \times \mathbb{Z}$ tiling problem. To simulate the $\mathbb{Z} \times \mathbb{Z}$ grid we use the formula

$$\exists A_0 \sqcap \exists A_1 \sqcap \forall \bigsqcap \big\{ A_i \to (A_j \leftrightarrows A_{i \oplus 1}) \mid i, j < 7, j \neq i, i \oplus 1 \big\},$$

where $\oplus$ is addition modulo 7. One can show that to satisfy it, a subspace of $\mathbb{R}^2$ must contain an infinite grid of the form:



4 Conclusion

Notice that the operator $\leftleftarrows$ is closely related to the implication $>$ of conditional logic [3]. In fact, one can show (by introducing the obvious semantics for $>$ on distance spaces) that $\mathcal{SL}^{\mathfrak{q}}$ without nominals has the same expressive power as conditional logic. To the best of our knowledge, conditional logic over finite metric spaces has not been investigated; however, $\mathcal{SL}^{\mathfrak{q}}$ over non-symmetric distance spaces corresponds to the conditional logic with properties (N), (R), (T), (U), (C) which is proved to be ExpTime-complete in [3]. We believe that the additional expressive power of the logic introduced in this paper could be useful for conditional logic as well.

Several problems remain open. We have provided an automata-based decision procedure, but we have not yet turned this into an optimised tableau-based decision procedure. Given the '**PDL**-like' acceptance condition required in the reduction, it is unclear whether such a tableau based algorithm can be efficient enough for applications. For this reason we are also exploring the algorithmic behaviour of fragments of the language introduced above.

Acknowledgements: The work on this paper was partially supported by the U.K. EPSRC research grants GR/S61966/01 and GR/S61973/01. We are grateful to Ivan Zakharyaschev for his generous help.

References

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] Ph.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Semantic similarity measures as tools for exploring the gene ontology. In *Pacific Symposium on Biocomputing*, pages 601–612, 2003.

[8] C. Lutz, F. Wolter, and M. Zakharyaschev. A tableau algorithm for reasoning about concepts and similarity. In *Proceedings of the Twelfth International Conference on Automated Reasoning with Analytic Tableaux and Related Methods TABLEAUX 2003*, volume 2796 of *LNCS*, 2003. Springer.

[9] Ph. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR)*, 11:95–130, 1999.

[10] M. Sheremet, D. Tishkowsky, F. Wolter, and M. Zakharyaschev. Comparative similarity, tree automata, and Diophantine equations. Manuscript, 2005 (available at `http://www.dcs.kcl.ac.uk/staff/mz`).

[11] R. Stevens, I. Horrocks, C. Goble, and S. Bechhofer Building a Reason-able Bioinformatics Ontology Using OIL. IJCAI'01 Workshop on Ontologies and Information Sharing, pp.81–90, 2001.

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

[13] T. Warnow. Mathematical approaches to comparative linguistics. *PNAS*, 94(13):6585–6590, 1997.

[14] 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.

[15] F. Wolter and M. Zakharyaschev. A logic for metric and topology. *Journal of Symbolic Logic*, 2005. (In print).