Comparative similarity, tree automata, and Diophantine equations

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 7ZF, U.K.
{dmitry,frank@csc.liv.ac.uk}

Abstract. The notion of comparative similarity 'X is more similar or closer to Y than to Z' has been investigated in both foundational and applied areas of knowledge representation and reasoning, e.g., in concept formation, similarity-based reasoning and areas of bioinformatics such as protein sequence alignment. In this paper we analyse the computational behaviour of the 'propositional' logic with the binary operator 'closer to a set τ_1 than to a set τ_2' and nominals interpreted over various classes of distance (or similarity) spaces. In particular, using a reduction to the emptiness problem for certain tree automata, we show that the satisfiability problem for this logic is ExpTime-complete for the classes of all finite symmetric and all finite (possibly non-symmetric) distance spaces. For finite subspaces of the real line (and higher dimensional Euclidean spaces) we prove the undecidability of satisfiability by a reduction of the solvability problem for Diophantine equations. As our 'closer' operator has the same expressive power as the standard operator $>$ of conditional logic, these results may have interesting implications for conditional logic as well.

1 Introduction

There are two main approaches to defining and classifying concepts in computer science and artificial intelligence. One of them is *logic based*. It uses formalisms like description logics to define concepts by establishing relationships between them, for example,

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

The main tool for analysing and using such definitions (e.g., to compute the concept hierarchy based on the subsumption relation) is *reasoning*.

Another approach is based on *similarity*.[1] Using various techniques (such as alignment algorithms) we compute suitable similarity measures on (part of) the application domain and then define concepts in terms of similarity, 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 established tools for dealing with concepts of this sort are *numerical computations* (say, with the help of Voronoi tessellations, nearest neighbour or clustering algorithms).

As more and more application areas—like bioinformatics and linguistics—use both of these ways of defining concepts, we are facing the problem of integrating them. In particular, we need formalisms that are capable of *reasoning* about concepts defined in terms of (explicit or implicit) similarity in the same way as this is done in description logic (DL).

In [6, 17, 8, 18] we presented and investigated rudimentary DL-like formalisms for reasoning about concepts and similarity with concept constructors of the form $\exists^{<a}\tau$, that is, 'in the a-neighbourhood of τ,' where $a \in \mathbb{Q}^{\geq 0}$. The apparent limitation of these languages is that they can only operate with *concrete* degrees of similarity $a \in \mathbb{Q}^{\geq 0}$, and so require substantial expert knowledge in order to define concepts.

In this paper we propose a purely *qualitative* logic $\mathcal{CSL}$ for knowledge representation and reasoning about *comparative similarity*. Its main ingredients are the binary *closer* operator $\leftleftarrows$ as in the example above and individual constants (nominals) for representing prototypical objects (we refer the reader to [7, 16] for a discussion of relations like 'X more similar to Y than to Z'). The logic is interpreted in various natural classes of distance (or similarity) spaces such as finite metric spaces, finite metric spaces without symmetry (see, e.g., [14] for an argumentation that similarity measures are not necessarily symmetric) as well as the finite subspaces of the Euclidean space $\mathbb{R}^n$, $n \geq 1$ (natural similarity measures for weight, length, etc.).

The computational behaviour of $\mathcal{CSL}$ over the class of finite metric spaces (with or without symmetry) turns out be similar to the behaviour of standard description logics: the satisfiability problem is ExpTime-complete which can be established by a reduction to the emptiness problem for certain tree automata. However, it was a great surprise for us to discover that over finite subspaces of the real line $\mathbb{R}$ (as well as any higher dimensional Euclidean space or any $\mathbb{Z}^n$) the logic turns out to be undecidable. This result is proved by a reduction of the (undecidable) solvability problem for Diophantine equations.

Because of the space limit some proofs in this paper are only sketched, some are omitted. For a detailed exposition the reader is referred to the full version [11].

[1] "There is nothing more basic to thought and language than our sense of similarity; our sorting of things into kinds." Quine (1969)

2 The logic of comparative similarity

The *logic $\mathcal{CSL}$ of comparative similarity* we consider in this paper is based on the following language:

$$\tau \quad ::= \quad p_i \quad | \quad \{\ell_i\} \quad | \quad \neg\tau \quad | \quad \tau_1 \sqcap \tau_2 \quad | \quad \tau_1 \leftleftarrows \tau_2$$

where the p_i are *atomic terms*, the ℓ_i are *object names*, and $\leftleftarrows$ is the *closer operator*. We call $\{\ell_i\}$ a *nominal* and τ a *$\mathcal{CSL}$-term* or simply a *term*.

The intended models for $\mathcal{CSL}$ are based on *distance* (or rather *similarity*) *spaces* $\mathfrak{D} = (\Delta, d)$, where Δ is a nonempty set and d is a map from $\Delta \times \Delta$ to the set $\mathbb{R}^{\geq 0}$ of nonnegative real numbers such that, for all $x, y \in \Delta$, we have $d(x, y) = 0$ iff $x = y$. If the *distance function* d satisfies two additional properties

$$d(x, y) \;=\; d(y, x) \qquad (\mathbf{sym})$$
$$d(x, z) \;\leq\; d(x, y) + d(y, z) \qquad (\mathbf{tr})$$

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

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

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

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

for any nonempty X and Y, then the distance space $\mathfrak{D}$ is called a *min-space*. Every finite distance space is clearly a min-space.

$\mathcal{CSL}$-models are structures of the form

$$\mathfrak{I} \;=\; \left(\Delta^{\mathfrak{I}}, d^{\mathfrak{I}}, \ell_1^{\mathfrak{I}}, \ell_2^{\mathfrak{I}}, \ldots, p_1^{\mathfrak{I}}, p_2^{\mathfrak{I}}, \ldots\right), \qquad (1)$$

where $\left(\Delta^{\mathfrak{I}}, d^{\mathfrak{I}}\right)$ is a distance space, the $p_i^{\mathfrak{I}}$ are subsets of $\Delta^{\mathfrak{I}}$, and $\ell_i^{\mathfrak{I}} \in \Delta^{\mathfrak{I}}$ for every i. We call such models *min-models*, *symmetric* or satisfying the *triangle inequality* if the underlying distance space satisfies (min), (sym) or (tr), respectively. If both (sym) and (tr) are satisfied then $\mathfrak{I}$ is called a *metric $\mathcal{CSL}$-model*.

The interpretation of the Boolean operators $\neg$ and $\sqcap$ in $\mathfrak{I}$ is as usual (we will use $\sqcup$, $\rightarrow$, $\bot$ (for $\emptyset$), and $\top$ (for the whole space) as standard abbreviations), $\{\ell\}^{\mathfrak{I}} = \{\ell^{\mathfrak{I}}\}$, and

$$(\tau_1 \leftleftarrows \tau_2)^{\mathfrak{I}} \quad = \quad \{x \in \Delta^{\mathfrak{I}} \mid d^{\mathfrak{I}}(x, \tau_1^{\mathfrak{I}}) < d^{\mathfrak{I}}(x, \tau_2^{\mathfrak{I}})\}. \qquad (2)$$

A term τ is *satisfied* in $\mathfrak{I}$ if $\tau^{\mathfrak{I}} \neq \emptyset$. More precisely, we say that $x \in \Delta^{\mathfrak{I}}$ *satisfies* τ whenever $x \in \tau^{\mathfrak{I}}$. τ is *satisfiable* in a class $\mathcal{C}$ of models if it is satisfied in some model from $\mathcal{C}$. Finally, τ is *valid* in $\mathfrak{I}$ if $\tau^{\mathfrak{I}} = \Delta^{\mathfrak{I}}$.

The seemingly simple logic $\mathcal{CSL}$ turns out to be quite expressive. First, the operator $\exists\tau = (\tau \leftleftarrows \bot)$ is interpreted by the *existential modality* (in the sense that $\exists\tau$ is the whole space iff τ is not empty); its dual, the *universal modality*, will

be denoted by $\forall$. Thus the term $\forall(\tau_1 \to \tau_2)$ expresses in $\mathcal{CSL}$ the *subsumption relation* $\tau_1 \sqsubseteq \tau_2$ which is usually used in description logic knowledge bases. Note also that the following definable operator $\leftrightarrows$ means 'at the same distance:'

$$\tau_1 \leftrightarrows \tau_2 \quad = \quad \neg(\tau_1 \leftleftarrows \tau_2) \sqcap \neg(\tau_2 \leftleftarrows \tau_1). \tag{3}$$

Second, in metric models the operator $\Box$ defined by taking $\Box\tau = (\top \leftleftarrows \neg\tau)$ is actually interpreted by 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., [1]). The topological aspects of $\mathcal{CSL}$ will be considered elsewhere.

Finally, it is to be noted that the operator $\leftleftarrows$ is closely related to the 'implication' $>$ of conditional logics. According to Lewis' [7] semantics for conditionals, propositions are interpreted in a set W of possible worlds that 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 $\varphi > \psi$ is true at w iff, for every $\preceq_w$-minimal v with $v \models \varphi$, we have $v \models \psi$. Various authors (see, for example, [3, 10]) 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 $\mathfrak{I} = \big(\Delta^{\mathfrak{I}}, d^{\mathfrak{I}}, p_1^{\mathfrak{I}}, p_2^{\mathfrak{I}}, \ldots\big)$ we have

$$(p_1 > p_2)^{\mathfrak{I}} \quad = \quad \big((p_1 \leftleftarrows (p_1 \sqcap \neg p_2)) \sqcup \forall\neg p_1\big)^{\mathfrak{I}}$$

and, conversely,

$$(p_1 \leftleftarrows p_2)^{\mathfrak{I}} \quad = \quad \big(((p_1 \sqcup p_2) > p_1) \sqcap (p_1 > \neg p_2) \sqcap \neg(p_1 > \bot)\big)^{\mathfrak{I}}.$$

Relations $\preceq_w$ induced by *symmetric* distance spaces have not been considered in the conditional logic literature. According to the classification of [5], our (nominal-free) logic of arbitrary min-spaces corresponds to the conditional logic of frames satisfying the normality, reflexivity, strict centering, uniformity and connectedness conditions.

3 Main results

In this paper, our main concern is the computational behaviour of $\mathcal{CSL}$ over natural classes of *min-models*, in particular, *finite* models.

Theorem 1. *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 the satisfiability problem for $\mathcal{CSL}$-terms in $\mathcal{C}$ is ExpTime-complete. Moreover, a term is satisfiable in $\mathcal{C}$ iff it is satisfiable in a finite model from $\mathcal{C}$.*

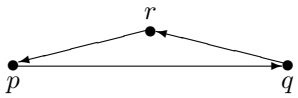
Remark 1. For the nominal-free fragment of $\mathcal{CSL}$ over arbitrary min-models, Theorem 1 was essentially proved in [5] in the framework of conditional logic. We provide a new proof here because it serves as a preparation for the more sophisticated proof for the class of *symmetric* min-models.

Remark 2. It is to be noted that in fact the language of $\mathcal{CSL}$ cannot distinguish between models with and without (tr). To see this, let us suppose that τ is satisfied in a model $\mathfrak{I}$ of the form (1) which does not satisfy (tr). Take any strictly monotonic function $f : \mathbb{R}^{\geq 0} \to (9, 10)$, where $(9, 10)$ is the open interval between 9 and 10. Define a new model $\mathfrak{I}'$ which differs from $\mathfrak{I}$ only in the distance function: $d^{\mathfrak{I}'}(x, y) = f(d^{\mathfrak{I}}(x, y))$ for all $x \neq y$ and $d^{\mathfrak{I}'}(x, x) = 0$ for all x. Clearly, $\mathfrak{I}'$ satisfies the triangle inequality. It is easily checked that τ is satisfied in $\mathfrak{I}'$.

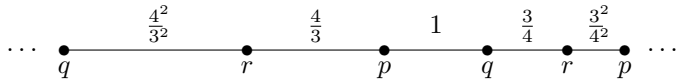
Remark 3. On the other hand, $\mathcal{CSL}$ can distinguish between models with and without (sym). Consider, for example, the term

$$p \sqcap \forall\big((p \to (q \leftleftarrows r)) \sqcap (q \to (r \leftleftarrows p)) \sqcap (r \to (p \leftleftarrows q))\big).$$

One can readily check that it is satisfiable in a three-point model without (sym), say, in the one depicted below where the distance from x to y is the length of the shortest directed path from x to y.



However, this term is not satisfiable in any symmetric min-model. On the other hand, it can be satisfied in the following subspace of $\mathbb{R}$ which is not a min-space:



Our second main result is quite surprising: $\mathcal{CSL}$ turns out to be undecidable when interpreted in finite subspaces of $\mathbb{R}$. More precisely, we are going to prove the following:

Theorem 2. *For each $n \geq 1$, the satisfiability problem for $\mathcal{CSL}$-terms is undecidable in the class of finite models and the class of min-models based on subspaces of $\mathbb{R}^n$, or only $\mathbb{Z}^n$.*

Theorem 1 will be proved in the next section: for the lower bound we use a reduction of the global consequence relation for the modal logic **K**, while the upper bound is established by reduction to the emptiness problem for tree automata. Theorem 2 is proved in Section 5 by reduction of the solvability problem for Diophantine equations (and, for $n \geq 2$, the $\mathbb{Z} \times \mathbb{Z}$ tiling problem).

4 Proof of Theorem 1

The ExpTime lower bound is proved by a reduction of the global consequence relation for the modal logic **K** which is known to be ExpTime-hard [12]. A detailed proof can be found in [11]. We now show how to establish the ExpTime upper bound and the finite model property with respect to the given class $\mathcal{C}$ of models.

Given a term τ, denote by $cl\,\tau$ the closure under single negation of the set consisting of all subterms of τ, the term $\bot$, and the term $\exists\rho = \rho \leftleftarrows \bot$ for every subterm ρ of τ. A *type* t for τ is a subset of $cl\,\tau$ such that $\bot \notin t$ and the following Boolean closure conditions are satisfied:

- $\tau_1 \sqcap \tau_2 \in t$ iff $\tau_1, \tau_2 \in t$, for every $\tau_1 \sqcap \tau_2 \in cl\,\tau$,
- $\neg\rho \in t$ iff $\rho \notin t$, for every $\neg\rho \in cl\,\tau$.

Clearly, $|cl\,\tau|$ is a linear function of the *length* $|\tau|$ (say, the number of subterms) of τ.

A 'typical' type is given by an element $w \in \Delta^{\mathfrak{I}}$ from a model $\mathfrak{I}$ of the form (1), namely,

$$t^{\mathfrak{I}}(w) \;=\; \{\rho \in cl\,\tau \mid w \in \rho^{\mathfrak{I}}\}.$$

A τ-*bouquet* is a pair $\mathfrak{B} = (T_{\mathfrak{B}}, \leq_{\mathfrak{B}})$, where $T_{\mathfrak{B}}$ is a set of types for τ such that $2 \leq |T_{\mathfrak{B}}| \leq |cl\,\tau|$, and $\leq_{\mathfrak{B}}$ is a transitive, reflexive, and connected relation on $T_{\mathfrak{B}}$ with a unique minimal element $t_{\mathfrak{B}} \in T_{\mathfrak{B}}$ for which the following conditions hold:

- $\tau_1 \leftleftarrows \tau_2 \in t_{\mathfrak{B}}$ iff there exists some $t \in T_{\mathfrak{B}}$ such that $\tau_1 \in t$ and $\tau_2 \notin t'$ for any $t' \leq_{\mathfrak{B}} t$,
- $\exists\rho \in t$ for some $t \in T_{\mathfrak{B}}$ iff $\exists\rho \in t$ for all $t \in T_{\mathfrak{B}}$.

We use the following notation:

$$\begin{array}{lll} t \sim_{\mathfrak{B}} t' & \text{iff} & t \leq_{\mathfrak{B}} t' \quad \text{and} \quad t' \leq_{\mathfrak{B}} t \\ t <_{\mathfrak{B}} t' & \text{iff} & t \leq_{\mathfrak{B}} t' \quad \text{and} \quad t \not\sim_{\mathfrak{B}} t'. \end{array}$$

The intended meaning of a τ-bouquet $\mathfrak{B}$ is to encode the local requirements in order to realise the type $t_{\mathfrak{B}}$. A 'typical' τ-bouquet can be obtained by taking a point w from $\mathfrak{I}$ above and then selecting, for every term $\tau_1 \leftleftarrows \tau_2$ from $t^{\mathfrak{I}}(w)$, a point w' such that $d^{\mathfrak{I}}(w, w')$ is minimal with $w' \in \tau_1^{\mathfrak{I}}$. Denote by V the set of all selected points. Clearly, $|V| < |cl\,\tau|$ and we can assume that $t^{\mathfrak{I}}(w_1) \neq t^{\mathfrak{I}}(w_2)$ for any two distinct w_1, w_2 from V. If $|V| \geq 1$, then we define the τ-*bouquet* $(T_V^{\mathfrak{I}}(w), \leq_w)$ *induced by* w *and* V *in* $\mathfrak{I}$ by taking

$$\begin{array}{l} T_V^{\mathfrak{I}}(w) \;=\; \{t^{\mathfrak{I}}(w)\} \cup \{t^{\mathfrak{I}}(w') \mid w' \in V\}, \\ t^{\mathfrak{I}}(w') \leq_w t^{\mathfrak{I}}(w'') \quad \text{iff} \quad d^{\mathfrak{I}}(w, w') \leq d^{\mathfrak{I}}(w, w''). \end{array}$$

Notice that if we require a certain type t satisfied in $\mathfrak{I}$ to be a member of the bouquet then we can add to V a point w' such that $d(w, w')$ is minimal with

$t = t^{\mathfrak{I}}(w')$ and form the bouquet induced by w and $V \cup \{w'\}$. In particular, if $\mathfrak{I}$ satisfies at least two distinct types, then we can always find a set V such that w and V induce a bouquet. In what follows we will only be working with models satisfying at least two distinct types. This is the interesting case because the problem of checking satisfiability in a model with only one type is clearly decidable in NP.

4.1 Non-symmetric case

First we establish the finite model property and the ExpTime upper bound for satisfiability in min-models that are not necessarily symmetric. Let N be the set of nominals occurring in τ. A set B τ-bouquets is said to be *nominal ready* if there is a set $\{t_\ell \mid \ell \in N\}$ of types for τ such that whenever $\{\ell\} \in t \in T_{\mathfrak{B}}$, for some $\mathfrak{B} \in B$, then $t = t_\ell$.

Let $k = |cl\tau|$. We remind the reader that the *full k-ary tree over* the set $\{1, \ldots, k\}^*$ (of finite sequences of elements of $\{1, \ldots, k\}$) 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 $K : \{1, \ldots, k\}^* \to L$ will be called an *L-labelled tree over* $\{1, \ldots, k\}^*$.

A *Hintikka tree satisfying* τ is a B-labelled tree K over $\{1, \ldots, k\}^*$, for some nominal ready set B of τ-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 $T_{K(\alpha)}$ in the bouquet $K(\alpha)$):

- $\tau \in t_{K(\epsilon)}$,
- for every nominal $\ell \in N$, there exists a type in $K(\epsilon)$ containing $\{\ell\}$,
- for every $\alpha \in \{1, \ldots, k\}^*$, $K(\alpha)$ is a bouquet such that

$$T_{K(\alpha)} \setminus \{t_{K(\alpha)}\} \;=\; \{t_{K(\alpha i)} \mid 1 \leq i \leq k\}$$

  and $t_{K(\alpha)} \in T_{K(\alpha i)}$, for $1 \leq i \leq k$.

Lemma 1. *For every term τ, the following conditions are equivalent:*

(a) *τ is satisfiable in some min-model (with at least two distinct types);*
(b) *there exists a Hintikka tree satisfying τ over $\{1, \ldots, k\}^*$, where $k = |cl\tau|$;*
(c) *τ is satisfiable in a finite model (with at least two distinct types).*

Proof. (a) $\Rightarrow$ (b) Suppose that $\tau^{\mathfrak{I}} \neq \emptyset$ in some model $\mathfrak{I} \in \mathcal{C}$ of the form (1) with at least two distinct types. We define a Hintikka tree K satisfying τ by induction as follows. First take some $w \in \tau^{\mathfrak{I}}$ and set

$$K(\epsilon) \;=\; (T^{\mathfrak{I}}_{V_\epsilon}(w), \leq_w),$$

where $(T^{\mathfrak{I}}_{V_\epsilon}(w), \leq_w)$ is a bouquet induced by w and a suitable set $V_\epsilon \subseteq W$ containing $\{\ell\}^{\mathfrak{I}}$ for all ℓ that occur in τ. Here and in what follows we assume that we construct the underlying sets of the bouquet as described above in the introduction of bouquets.

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

$$K(\alpha) \;=\; (T^{\mathfrak{J}}_{V_\alpha}(w_\alpha), \leq_{w_\alpha}),$$

where $(T^{\mathfrak{J}}_{V_\alpha}(w_\alpha), \leq_{w_\alpha})$ is induced by w_α and a suitable set V_α. Take some surjective map $s : \{1,\dots,k\} \to V_\alpha$. For each j, $1 \leq j \leq k$, let

$$K(\alpha j) \;=\; (T^{\mathfrak{J}}_{V_{\alpha j}}(s(j)), \leq_{s(j)})$$

where $(T^{\mathfrak{J}}_{V_{\alpha j}}(s(j)), \leq_{s(j)})$ is the bouquet induced by $s(j)$ and a suitable set $V_{\alpha j}$ which contains a w' such that $t^{\mathfrak{J}}(w') = t^{\mathfrak{J}}(w_\alpha)$.

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

(b) $\Rightarrow$ (c) Suppose that $K : \{1,\dots,k\}^* \to B$ with

$$K(\alpha) \;=\; (T_\alpha, \leq_\alpha)$$

is a Hintikka tree satisfying τ over a nominal ready set B of τ-bouquets. First we define a distance space (Δ_0, d_0) with the domain $\Delta_0 = \{1,\dots,k\}^*$ in the following way. Take a finite subset I of the interval $(0,1)$ and, for each $\alpha \in \Delta_0$, a map

$$f_\alpha : (T_{K(\alpha)} \setminus \{t_{K(\alpha)}\}) \to I$$

for which $t <_{K(\alpha)} t'$ iff $f_\alpha(t) < f_\alpha(t')$. Now set

- $d_0(\alpha, \alpha i) = f_\alpha(t_{K(\alpha i)})$ for all $\alpha \in \Delta_0$ and $1 \leq i \leq k$,
- $d_0(\alpha,\alpha) = 0$ and,
- $d_0(\alpha,\beta) = 1$ for $\beta \notin \{\alpha, \alpha 1, \dots, \alpha k\}$.

It is not difficult to see that (Δ_0, d_0) is a (non-symmetric) min-space.

For every type t such that $t = t_{K(\alpha)}$ for some $\alpha \in \Delta_0$, we fix exactly one α with this property. Let Δ be the set of the selected α. Construct a finite distance model from $\mathcal{C}$

$$\mathfrak{I} \;=\; (\Delta, d, \ell_1^{\mathfrak{I}}, \dots, p_1^{\mathfrak{I}}, \dots)$$

by taking $p_i^{\mathfrak{I}} = \{\alpha \in \Delta \mid p_i \in t_{K(\alpha)}\}$, $\ell_i^{\mathfrak{I}} = \alpha$ for the unique $\alpha \in \Delta$ with $\{\ell_i\} \in t_{K(\alpha)}$, and, for $\alpha, \beta \in \Delta$,

$$d(\alpha,\beta) = d_0(\alpha, \{\beta' \in \Delta_0 \mid t_{K(\beta')} = t_{K(\beta)}\}).$$

Now, given a subterm ρ of τ, one can prove by induction on the construction of ρ that $\alpha \in \rho^{\mathfrak{I}}$ iff $\rho \in t_{K(\alpha)}$. Therefore, τ is satisfied in $\mathfrak{I}$.

The implication (c) $\Rightarrow$ (a) is clear.

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 [15, 13]. 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 [13]: given a looping automaton for k-ary trees, decide whether the set of trees it accepts is empty.

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

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

It follows immediately from Lemma 1 and the given definitions that the runs of $\mathcal{A}^B_\tau$ on Σ-labelled trees are exactly the B-labelled Hintikka-trees satisfying τ.

Lemma 2. *A term τ is satisfiable in a min-model (with at least two types) iff there exists a nominal ready set B such that $\mathcal{A}^B_\tau$ accepts at least one tree.*

As there are only exponentially many different nominal ready sets B and as $\mathcal{A}^B_\tau$ is only exponential in $|cl\tau|$, the satisfiability problem in min- (and finite) models is decidable in ExpTime.

4.2 Symmetric case

The construction is more involved if we deal with the class of symmetric $\mathcal{CSL}$-models. Suppose that B is a nominal ready set of τ-bouquets, $|cl\tau| = k$, and $K : \{1, \ldots, k\}^* \to B$ is a B-labelled Hintikka tree with $K(\alpha) = (T_\alpha, \leq_\alpha)$ and $t_\alpha = t_{K(\alpha)}$, for $\alpha \in \{1, \ldots, k\}^*$.

We 'paint' each node of K in one of three 'colours:' inc (for increasing), const (for constant), and dec (for decreasing). The colour of a node α will be denoted by $c(\alpha)$. It is defined by induction as follows. The root ϵ and its immediate successors are painted with the same colour, say, $c(\epsilon) = c(1) = \cdots = c(k) = \mathsf{inc}$. Suppose now that we have already defined $c(\alpha i)$. Then, for $1 \leq j \leq k$, we set

– $c(\alpha ij) = \mathsf{const}$ iff $t_{\alpha ij} \sim_{\alpha i} t_\alpha$,
– $c(\alpha ij) = \mathsf{dec}$ iff $t_\alpha >_{\alpha i} t_{\alpha ij}$,
– $c(\alpha ij) = \mathsf{inc}$ iff $t_\alpha <_{\alpha i} t_{\alpha ij}$.

Intuitively, the colours determine whether in the symmetric space (Δ_0, d_0) to be constructed from $\{1,\ldots,k\}^*$ we have $d_0(\alpha,\alpha i) = d_0(\alpha i,\alpha ij)$ (the constant case), $d_0(\alpha,\alpha i) < d_0(\alpha i,\alpha ij)$ (the increasing case), or $d_0(\alpha,\alpha i) > d_0(\alpha i,\alpha ij)$ (the decreasing case).

We call K a *min-tree* if its every branch with infinitely many dec nodes also contains infinitely many inc nodes.

We require two simple observations; see [11] for proofs. First, the Hintikka tree K constructed in the proof of Lemma 1 starting from a *symmetric* min-model $\mathfrak{I}$ is a min-tree. And second, if there is a sequence $\alpha, \alpha i_1, \ldots, \alpha i_1 \cdots i_{n+1}$ (for $1 \leq i_j \leq k$) of nodes of K such that

$$(K(\alpha), K(\alpha i_1)) \;=\; (K(\alpha i_1 \cdots i_n), K(\alpha i_1 \cdots i_{n+1}))$$

then by 'cutting off' the nodes $\alpha, \ldots, \alpha i_1 \cdots i_{n-1}$ we obtain again a B-labelled Hintikka tree such that the colours of the (renamed) nodes do not change.

We are now in a position to prove a symmetric analogue of Lemma 1.

Lemma 3. *For every term τ, the following conditions are equivalent:*

(a) *τ is satisfiable in some symmetric min-model (with at least two distinct types);*
(b) *there exists a Hintikka min-tree satisfying τ over $\{1,\ldots,k\}^*$, where $k = |cl\tau|$;*
(c) *τ is satisfiable in a finite symmetric model (with at least two different types).*

Proof. (a) $\Rightarrow$ (b) is established in precisely the same way as in the proof of Lemma 1 using the first observation above that if we start with a symmetric model then the resulting Hintikka tree is a min-tree. (c) $\Rightarrow$ (a) is again trivial.

(b) $\Rightarrow$ (c) Suppose that $K : \{1,\ldots,k\}^* \to B$ is a Hintikka min-tree satisfying τ with

$$K(\alpha) = (T_\alpha, \leq_\alpha) \quad \text{and} \quad t_\alpha = t_{K(\alpha)}.$$

By the second observation above, without loss of generality we may assume that if no node in a path of the form $\alpha, \alpha i_1, \ldots, \alpha i_1 \cdots i_n$ is inc then no two dec nodes βij and $\beta' ij$ in it can have predecessors $(\beta, \beta i)$ and $(\beta', \beta' j)$ such that $(K(\beta), K(\beta i)) = (K(\beta'), K(\beta' j))$. It follows that there is a number n_τ (exponential in $|cl\tau|$) which bounds the numbers of dec nodes in each such path.

Now we define a symmetric distance space (Δ_0, d_0) with $\Delta_0 = \{1,\ldots,k\}^*$ (symmetry means that $d_0(\alpha,\alpha i) = d_0(\alpha i,\alpha)$ for $\alpha \in \Delta_0$ and $1 \leq i \leq k$). First we take a set $D \subset (9,10)$ of cardinality $n_\tau \times |cl\tau|$. For all $1 \leq i \leq k$ we define $d_0(\epsilon, i)$ to be the maximal numbers in D such that we can satisfy the constraint: $d_0(\epsilon,i) < d_0(\epsilon,j)$ iff $t_{K(i)} <_\epsilon t_{K(j)}$, for $1 \leq i,j \leq k$. Suppose now that $d_0(\alpha,\alpha i) \in D$ is defined. Then we define $d_0(\alpha i,\alpha ij)$ to be the maximal number in D such that we can satisfy the constraints

- $d_0(\alpha i,\alpha ij) = d_0(\alpha,\alpha i)$ for $t_{K(\alpha ij)} = t_{K(\alpha)}$ and
- $d_0(\alpha i,\alpha ij) < d_0(\alpha i,\alpha ij')$ iff $t_{K(\alpha ij)} <_{\alpha i} t_{K(\alpha ij')}$, for $1 \leq j,j' \leq k$.

Notice that this is possible by the definition of n_τ. Finally, set $d_0(\alpha,\alpha) = 0$ and $d_0(\alpha,\beta) = 10$ for all remaining $\alpha \neq \beta$.

Now construct a finite symmetric model $\mathfrak{I} = (\Delta, d, p_1^{\mathfrak{I}}, \dots, \ell_1^{\mathfrak{I}}, \dots)$ as follows. Let $\sim$ be the equivalence relation on Δ_0 defined by taking $\alpha \sim \beta$ iff $t_{K(\alpha)} = t_{K(\beta)}$. Then we set

$$[\alpha] = \{\beta \in \Delta_0 \mid \alpha \sim \beta\}, \quad \Delta = \{[\alpha] \mid \alpha \in \{1,\dots,k\}^*\}, \quad d([\alpha],[\beta]) = d_0([\alpha],[\beta])$$

and $[\alpha] \in p_i^{\mathfrak{I}}$ iff $p_i \in t_{K(\alpha)}$, and $\ell_i^{\mathfrak{I}} = [\alpha]$ for the uniquely determined $[\alpha]$ such that $\{\ell_i\} \in t_{K(\alpha)}$. We leave it to the reader to check that this model is as required.

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 in Section 4.1,
- 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 [4], the emptiness problem for single complemented pair automata is decidable in polynomial time. We show now how to reduce the satisfiability problem for $\mathcal{CSL}$-terms in symmetric models to the emptiness problem for these automata.

A *coloured τ-bouquet* is a pair $(\mathfrak{B}, c)$ where $\mathfrak{B} = (T_{\mathfrak{B}}, \leq_{\mathfrak{B}})$ is a τ-bouquet and c is a function from $T_{\mathfrak{B}}$ to $\{\mathsf{dec}, \mathsf{inc}, \mathsf{const}\}$.

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

- Σ to be the set of types occurring in coloured bouquets of B,
- $Q = B$,
- $Q_0 = \{(\mathfrak{B}, c) \in B \mid \tau \in t_{\mathfrak{B}},\ \mathfrak{B}$ contains a type with ℓ for every ℓ in $\tau\}$,
- $\mathsf{dec} = \{(\mathfrak{B}, c) \in B \mid c(t_{\mathfrak{B}}) = \mathsf{dec}\}$,
- $\mathsf{inc} = \{(\mathfrak{B}, c) \in B \mid c(t_{\mathfrak{B}}) = \mathsf{inc}\}$,
- $\big(t, (\mathfrak{B}_0, c_0), (\mathfrak{B}_1, c_1), \dots, (\mathfrak{B}_k, c_k)\big) \in \Gamma \quad$ iff $\quad t_{\mathfrak{B}_0} = t$,

$$T_{\mathfrak{B}_0} \setminus \{t_{\mathfrak{B}_0}\} \;=\; \{t_{\mathfrak{B}_i} \mid 1 \leq i \leq k\},$$

$t_{\mathfrak{B}_0} \in T_{\mathfrak{B}_i} \setminus \{t_{\mathfrak{B}_i}\}$, $c_i(t_{\mathfrak{B}_i}) = c_0(t_{\mathfrak{B}_i})$ for $1 \leq i \leq k$, and for all $t' \in T_{\mathfrak{B}_i} \setminus \{t_{\mathfrak{B}_i}\}$,

- $c_i(t') = \mathsf{inc} \quad$ iff $\quad t <_{\mathfrak{B}_i} t'$,
- $c_i(t') = \mathsf{const} \quad$ iff $\quad t' \sim_{\mathfrak{B}_i} t$,
- $c_i(t') = \mathsf{dec} \quad$ iff $\quad t' <_{\mathfrak{B}_i} t$.

It follows immediately from Lemma 3 and the given definitions that the runs of $\mathcal{A}_\tau^B$ on Σ-labelled trees are exactly the B-labelled Hintikka-trees satisfying τ.

Lemma 4. *A term τ is satisfiable in a symmetric min-model (with at least two distinct types) iff there exists a nominal ready set B of coloured τ-bouquets such that $\mathcal{A}_\tau^B$ accepts at least one tree.*

As there are only exponentially many different nominal ready sets B of coloured τ-bouquets and as $\mathcal{A}_\tau^B$ is only exponential in $|cl\tau|$, the satisfiability problem in symmetric min-models is decidable in ExpTime.

This completes the proof of Theorem 1.

5 Proof of Theorem 2

Here we only discuss the idea of the proof; see [11] for details. The proof is by reduction of the decision problem for Diophantine equations (Hilbert's 10th problem) which was shown to be undecidable by Matiyasevich–Robinson–Davis–Putnam (see [9, 2] and references therein). More precisely, we will use the following (still undecidable) variant of this problem:

given arbitrary polynomials g and h with coefficients from $\mathbb{N} \setminus \{0,1\}$, decide whether the equation $g = h$ has a solution in the set $\mathbb{N} \setminus \{0,1\}$.

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 $\mathcal{R}$ be the class of min-models based on subspaces of $\mathbb{R}$. We begin by considering the satisfiability problem for $\mathcal{CSL}$-terms in $\mathcal{R}$; then we discuss how to deal with finite models only. It is easy to see that any polynomial equation can be rewritten equivalently as a set of elementary equations of the form

$$x = y + z, \quad x = y \cdot z, \quad x = y, \quad x = n, \tag{4}$$

where x, y, z are variables and $n \in \mathbb{N} \setminus \{0,1\}$. Now we are facing the following three tasks:

(a) to ensure that a given model is based on a space similar to $\mathbb{Z}$;
(b) to define in such a model sets of the form $\{lk + j \mid k \in \mathbb{Z}\}$ that will be used to encode the number l;
(c) to encode addition and multiplication on such sets.

For (a) we set $\mathsf{Base}(\mathbf{p})$ to be the term

$$\exists p_0 \sqcap \exists p_1 \sqcap \exists p_2 \sqcap \forall(p_0 \sqcup p_1 \sqcup p_2) \sqcap \forall\bigsqcap_{i<3}\big(p_i \to \neg p_{i\oplus 1} \sqcap (p_{i\ominus 1} \leftrightarrows p_{i\oplus 1})\big),$$

where $\mathbf{p}$ is stands for p_0, p_1, p_2, and $\oplus$, $\ominus$ denote addition and subtraction modulo 3. Then a model $\mathfrak{I} \in \mathcal{R}$ satisfies $\mathsf{Base}(\mathbf{p})$ iff $\mathfrak{I}$ coincides (modulo an affine transformation) with a model $\mathfrak{Z}$ such that $\Delta^{\mathfrak{Z}} = \mathbb{Z}$ and $p_i^{\mathfrak{Z}} = \{3k + i \mid k \in \mathbb{Z}\}$, $i < 3$. Now we can simulate the functions '$+1$' and '-1' with the following analogues of the 'next-time' operator and its inverse (τ is an arbitrary term):

$$\bigcirc\tau = \bigsqcap_{i<3}\big(p_i \to (p_{i\oplus 1} \leftrightarrows (p_{i\oplus 1} \sqcap \tau))\big), \quad \bigcirc^{-1}\tau = \bigsqcap_{i<3}\big(p_i \to (p_{i\ominus 1} \leftrightarrows (p_{i\ominus 1} \sqcap \tau))\big).$$

Next, to fix an origin and an orientation for our model we take a fresh atom p and consider the term $\exists(p_2 \sqcap \neg p \sqcap \bigcirc p) \sqcap \forall(p \to \bigcirc p)$. It is satisfied in a model $\mathfrak{Z}$ of the above form iff $p^{\mathfrak{Z}} = \{k, k+1, \dots\}$ for some $k \in \mathbb{Z}$, $k \equiv 0 (\mathrm{mod}\, 3)$. Thus we can assume that $p^{\mathfrak{Z}} = \mathbb{N}$. Then $\mathsf{Zero} = p \sqcap \bigcirc^{-1} \neg p$ defines $\{0\}$.

For (b) we take the term

$$\mathsf{Seq}(\mathbf{q}) \;=\; \forall \prod_{i<3} \big(q_i \;\to\; \neg q_{i\oplus 1} \sqcap (q_{i\oplus 1} \leftrightarrows q_{i\ominus 1})\big) \;\sqcap\; \exists\big(q_0 \sqcap p \sqcap (q_2 \leftleftarrows (q_2 \sqcap p))\big)$$

which is satisfied in $\mathfrak{Z}$ iff $q_i^{\mathfrak{Z}} = \{lk + j \mid k \equiv i (\mathrm{mod}\, 3)\}$, $i < 3$, for some $j < l$ in $\mathbb{N}$. We say in this case that $\mathbf{q}$ (and $\{lk + j \mid k \in \mathbb{Z}\}$) *encodes* l with *indent* j. For each term τ, we let $\hat{\tau} = \tau \sqcap p$. If $\mathbf{q}$ encodes l with indent j then j, $j + l$, $j + 2l$ are the nearest points to 0 satisfying $\hat{q}_0$, $\hat{q}_1$ and $\hat{q}_2$, respectively. Clearly, two sets encode the same number iff they either coincide or strictly alternate. This can be expressed by the term

$$\mathsf{Alt}(\mathbf{q}, \mathbf{q}') \;=\; \forall \prod_{i<3} \big((q_i \to (q'_i \leftleftarrows q_{i\oplus 1})) \;\sqcap\; (q'_i \to (q_i \leftleftarrows q'_{i\oplus 1}))\big)$$

in conjunction with $\mathsf{Seq}(\mathbf{q})$ and $\mathsf{Seq}(\mathbf{q}')$. Note also that to encode a number n we can use the term $\mathsf{Seq}(\mathbf{q}) \sqcap \forall(\mathsf{Zero} \to (\hat{q}_1 \leftrightarrows \bigcirc^{-n} \mathsf{Zero}))$.

For (c), let $\mathbf{q}$, $\mathbf{r}$, $\mathbf{s}$ encode, respectively, the numbers u, v, w with indent 0. Let $v < w$ (which can be expressed by $\forall(\mathsf{Zero} \to (\hat{r}_1 \leftleftarrows \hat{s}_1))$). Then the term

$$\mathsf{Seq}(\mathbf{s}') \;\sqcap\; \mathsf{Alt}(\mathbf{s}, \mathbf{s}') \;\sqcap\; \forall(\mathsf{Zero} \to (\hat{s}'_0 \leftrightarrows \hat{r}_1) \sqcap (\hat{s}'_1 \leftrightarrows \hat{q}_1))$$

(with some fresh $\mathbf{s}'$) says that $\mathbf{s}'$ encodes w with indent v and that $v + w = u$. The case $w < v$ is symmetrical, while the case $v = w$, i.e., $u = v + v$, can be expressed by $\forall(\mathsf{Zero} \to (\hat{s}_1 \leftrightarrows \hat{r}_1) \sqcap (\hat{s}_2 \leftrightarrows \hat{q}_1))$. To encode multiplication, we use

Fact 1 *Let v, w be integer numbers with $0 < v < w - 1$. Then*

1) *$u = vw$ is the least solution to $u \equiv 0 (\mathrm{mod}\, w)$, $u \equiv v (\mathrm{mod}\, (w-1))$.*
2) *$u = (w-1)w$ is the least positive solution to $u \equiv 0 (\mathrm{mod}\, w)$, $u \equiv 0 (\mathrm{mod}\, (w-1))$.*
3) *$u = w^2$ is the least solution to $u \equiv 0 (\mathrm{mod}\, w)$, $u \equiv 1 (\mathrm{mod}\, (w-1))$, $x > w$.*

Suppose that $v < w - 1$. Take fresh $\mathbf{s}'$, $\mathbf{s}''$ and consider the term

$$\mathsf{Seq}(\mathbf{s}') \;\sqcap\; \mathsf{Seq}(\mathbf{s}'') \;\sqcap\; \mathsf{Alt}(\mathbf{s}', \mathbf{s}'') \;\sqcap\; \forall\big(\mathsf{Zero} \to s'_0 \sqcap (\hat{s}'_1 \leftrightarrows \bigcirc \hat{s}_1) \sqcap (\hat{s}''_0 \leftrightarrows \hat{r}_1)\big) \quad (5)$$

saying that $\mathbf{s}'$ and $\mathbf{s}''$ encode the same number, this number is $w - 1$, $\mathbf{s}'$ encodes it with indent 0, while $\mathbf{s}''$ with indent v. Let $s_* = s_0 \sqcup s_1 \sqcup s_2$, $s''_* = s''_0 \sqcup s''_1 \sqcup s''_2$. Then, in view of Fact 1 (1), term (5) means that $v \cdot w$ is the nearest point to 0 satisfying $\hat{s}_* \sqcap \hat{s}''_*$. Therefore, $\forall\big(\mathsf{Zero} \to (\hat{q}_1 \leftrightarrows (\hat{s}_* \sqcap \hat{s}''_*))\big)$ in conjunction with (5) ensures that $u = v \cdot w$.

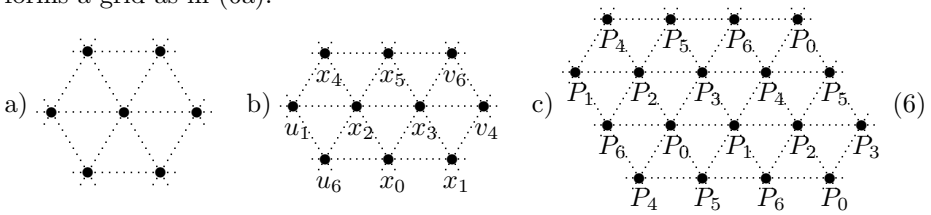
For the cases $v = w - 1$, $v = w$ we use similar constructions by applying Fact 1 (2,3). Thus we can encode any elementary equation from (4), and so the whole system of such equations representing the given polynomial equation.

In the finite case, we modify the terms $\mathsf{Base}(\mathbf{p})$, $\mathsf{Seq}(\mathbf{q})$, $\mathsf{Alt}(\mathbf{q}, \mathbf{q}')$ to take care of the endpoints. And to ensure that two encoding sets represent the same number, we require additionally that they are sufficiently long.

In fact, the undecidability for min-subspaces of $\mathbb{R}^n$, $n \geq 2$, can be proved by reduction of the $\mathbb{Z} \times \mathbb{Z}$ tiling problem. To simulate the $\mathbb{Z} \times \mathbb{Z}$ grid we use

$$\tau \quad = \quad \exists p_0 \sqcap \exists p_1 \ \sqcap \ \forall \bigsqcap \big\{ p_i \to \neg p_{i\oplus 1} \sqcap (p_{i\oplus 1} \leftrightarrows p_j) \mid i, j < 7, j \neq i, i\oplus 1 \big\},$$

where $\oplus$ is addition modulo 7. Let $\mathfrak{T}$ be a min-model satisfying τ and based on a subspace of $\mathbb{R}^2$, and let $P_i = p_i^{\mathfrak{T}}$ $(i < 7)$. Then one can show that $P_0 \cup \dots \cup P_6$ forms a grid as in (6a).



(6)

To encode tilings, we need to fix some concrete partition of this grid into the sets $P_0, \dots, P_6$. First we note that, in fact, it suffices to fix such a partition for a few points only. Indeed, suppose that $x_0 \in P_0$, $\dots$, $x_5 \in P_5$ are located as in (6b). Then we have $u_1 \in P_1$, $v_4 \in P_4$, $u_6, v_6 \in P_6$. This means that neither of the two six-point figures $u_6, x_0, x_1, x_2, x_3, v_4$ and $u_1, x_2, x_3, x_4, x_5, v_6$ contain a pair of points from the same P_i. It follows that the entire grid is as in (6c).

To ensure that such $x_0, \dots, x_5$ exist, we set, for distinct $i, j, k < 7$,

$$\mu_{ijk} \quad = \quad (p_i \leftrightarrows p_j) \sqcap (p_i \leftrightarrows p_k) \sqcap \bigsqcap \{ p_i \leftleftarrows p_l \mid l < 7, \ l \neq i, j, k \}.$$

Then $x \in \mathfrak{T}$ satisfies μ_{ijk} iff x is the centre of some small triangle $x_i x_j x_k$ with $x_i \in P_i$, $x_j \in P_j$, $x_k \in P_k$. Consider now the term $\sigma = \mu_{103} \sqcap (\mu_{032} \leftrightarrows p_3)$. Then σ is satisfied in $\mathfrak{T}$ iff there exist small triangles $x_1 x_0 x_3$, $x_0' x_3' x_2$ $(x_i, x_i' \in P_i)$ in our grid such that their centres x and x' belong to $\mathfrak{T}$, and the distances from x to x_3 and x' coincide. This can only be possible if $x_0 = x_0'$, $x_3 = x_3'$. Using this idea one can easily construct a term τ_1 that enforces a configuration of points $x_0 \in P_0$, $\dots$, $x_5 \in P_5$ as in (6b). Then every model satisfying $\tau \sqcap \tau_1$ will contain a grid of the form (6c), and we can encode the $\mathbb{Z} \times \mathbb{Z}$ tiling problem.

6 Outlook

In this paper, we have investigated the computational complexity of the basic logic $\mathcal{CSL}$ for comparative similarity. The final verdict is that this logic behaves similarly to standard description logics (is ExpTime-complete) over general classes of (finite or min-) distance spaces, but becomes undecidable when interpreted over (finite or min-) subspaces of Euclidean spaces.

Starting from the positive results, one can now investigate combinations of $\mathcal{CSL}$ with 'quantitative' similarity logics from [17, 8] as well as with description logics. On the other hand, it would be interesting to find out how one can avoid the 'negative' results for subspaces of $\mathbb{R}^n$. One promising route is to impose restrictions on the interpretations of set variables. For example, in many

applications it seems natural to assume that variables are interpreted as intervals in (subspaces of) $\mathbb{R}$. In this case decidability would follow immediately. Another related question is whether the computational behaviour of the logics depends on the 'crisp' truth-conditions. Exploring more relaxed 'non-punctual' truth-conditions could be important as well in order to take into account unprecise measurements, vagueness, and paradoxes of similarity such as the Sorites paradox.

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. A. Cohn and S. Hazarika. Qualitative spatial representation and reasoning: an overview. *Fundamenta Informaticae*, 43:2–32, 2001.
2. M. Davis. Unsolvable problems. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 567–594. North-Holland, Amsterdam, 1977.
3. J. P. Delgrande. Preliminary considerations on the modelling of belief change operators by metric spaces. In *NMR*, pages 118–125, 2004.
4. E. Emerson and C. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal of Computing*, 29:132–158, 1999.
5. N. Friedman and J. Halpern. On the complexity of conditional logics. In *Proceedings of KR'94*, pages 202–213, 1994.
6. 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.
7. D. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.
8. 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.
9. Yu. V. Matiyasevich. Enumerable sets are Diophantine. *Soviet Mathematics Doklady*, 11:354–358, 1970.
10. K. Schlechta. *Coherent Systems*. Elsevier, 2004.
11. M. Sheremet, D. Tishkovsky, F. Wolter, and M. Zakharyaschev. Comparative similarity, tree automata, and diophantine equations. Available at `http://www.csc.liv.ac.uk/~frank/publ/publ.html`, 2005.
12. E. Spaan. *Complexity of Modal Logics*. PhD thesis, University of Amsterdam, 1993.
13. W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, pages 133–191. Elsevier, 1990.
14. A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.
15. M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.
16. T. Williamson. First-order logics for comparative similarity. *Notre Dame Journal of Formal Logic*, 29:457–481, 1988.
17. 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.
18. F. Wolter and M. Zakharyaschev. A logic for metric and topology. *Journal of Symbolic Logic*, 70:795–828, 2005.