The Ultimate Undecidability Result for the Halpern–Shoham Logic

Jerzy Marcinkowski, Jakub Michaliszyn
Institute of Computer Science
University Of Wrocław
{jma,jmi}@cs.uni.wroc.pl

***Abstract*—The Halpern–Shoham logic is a modal logic of time intervals. Some effort has been put in last ten years to classify fragments of this beautiful logic with respect to decidability of its satisfiability problem. We complete this classification by showing — what we believe is quite an unexpected result — that the logic of subintervals, the fragment of the Halpern–Shoham logic where only the operator "during", or D, is allowed, is undecidable over discrete structures. This is surprising as this, apparently very simple, logic is decidable over dense orders [4] and its reflexive variant is known to be decidable over discrete structures [12]. Our result subsumes a lot of previous negative results for the discrete case, like the undecidability for ABE [9], BD [11], ADB, $A\bar{A}D$, and so on [2], [5].**

I. Introduction

In classical temporal logic, structures are defined by assigning properties (propositional variables) to time points (time is an ordering, discrete or dense). However, not all phenomena can be well described by such logics. Sometimes, we need to talk about actions (processes) that take some time and we would like to be able to say that one such action takes place, for example, during or after another.

The Halpern–Shoham logic [9], which is the subject of this paper, is one of the modal logics of time intervals. Judging by the number of papers published, and by the amount of work devoted to the research on it, this logic is probably the most influential time interval logic. But historically it was not the first one. Actually, the earliest papers about intervals in context of modal logic were written by philosophers , e.g., [8]. In computer science, the earliest attempts to formalize time intervals were process logic [16], [17] and interval temporal logic [14]. Relations between intervals in linear orders from an algebraic point of view were first studied systematically by Allen [1].

The Halpern–Shoham logic is a modal temporal logic, where the elements of a model are no longer — like in classical temporal logics — points in time, but rather pairs of points in time. Any such pair — call it $[p, q]$, where q is equal to or later than p — can be viewed as a (closed) time interval, that is, the set of all time points between p and q. HS logic does not assume anything about order — it can be discrete or continuous, linear or branching, complete or not.

Halpern and Shoham introduce six modal operators acting on intervals. Their operators are "begins" B, "during" D, "ends" E, "meets" A, "later" L, "overlaps" O and the six inverses of those operators: $\bar{B}, \bar{D}, \bar{E}, \bar{A}, \bar{L}, \bar{O}$. It is easy to see that the set of operators is redundant. For example, A, B and E can define D (B and E suffice for that – a prefix of my suffix is my infix) and L (here A is enough –"later" means "meets an interval that meets"). The operator O can be expressed using E and $\bar{B}$.

In their paper, Halpern and Shoham show that (satisfiability of formulae of) their logic is undecidable. Their proof requires logic with three operators (B, E and A are explicitly used in the formulae and, as we mentioned above, once B, E and A are allowed, D and L come for free) so they state a question about decidable fragments of their logic.

Considerable effort has been put since then to settle this question. First, it was shown [10] that the BE fragment is undecidable. Recently, negative results were also given for the classes $B\bar{E}$, $\bar{B}\bar{E}$, $\bar{B}E$, $A\bar{A}D, \bar{A}D\bar{B}, \bar{A}DB, \bar{A}\bar{D}\bar{B}, \bar{A}\bar{D}B$ [2], [5], and BD [11]. Another elegant negative result was that $O\bar{O}$ is undecidable over discrete orders [3].

On the positive side, it was shown that some small fragments, like $B\bar{B}$ or $E\bar{E}$, are decidable and easy to translate into standard, point-based modal logic [7]. The fragment using only A and $\bar{A}$ is harder and its decidability was only recently shown [5], [6]. Obviously, this last result implies decidability of $L\bar{L}$ as L is expressible by A. Another fragment known to be decidable is $AB\bar{B}\bar{L}$ [15].

A very simple, interesting fragment of the Halpern and Shoham logic of unknown status was the fragment with the single operator D ("during"), which we call here *the logic of subintervals*. Since D does not seem to have much expressive power (a natural language account of a D-formula would be "each morning I spend a while thinking of you" or "each nice period of my life contains an unpleasant fragment"), logic of subintervals was widely believed to be decidable. A number of decidability results concerning variants of this logic has been published. For example, it was shown in [4], [13] that satisfiability of formulae of logic of subintervals is decidable over dense structures. In [12] decidability is proved for the (slightly less expressive) "reflexive D". The results in [18] imply that D (as well as some richer fragments of the HS logic) is decidable if we allow models in which not all the intervals defined by the ordering are elements of the Kripke structure. On the negative side, no nontrivial lower bound was known for

satisfiability of this logic.

In this paper, we show that satisfiability of formulae from the D fragment is undecidable over the class of finite orderings as well as over the class of all discrete orderings. Our result subsumes the negative results for the discrete case for ABE [9], BD [11] and ADB, $A\bar{A}D$ [2], [5]. The logic of subintervals for finite orderings is so simple that we are tempted to write that it is one of the simplest known undecidable logics.

A. Main theorems and an overview of the proofs

Our contribution consists of the proofs of the following two theorems:

Theorem 1: The satisfiability problem for the formulae of the logic of subintervals, over models which are suborders of the order $\langle \mathbb{Z}, \leq \rangle$, is undecidable.

Since truth value of a formula is defined with respect to a model and an initial interval in this model (see Preliminaries), and since the only allowed operator is D, which means that the truth value of a formula in a given interval depends only on the labeling of this interval and its subintervals, Theorem 1 can be restated as: *The satisfiability problem for the formulae of the logic of subintervals, over finite models is undecidable*, and it is this version that will be proved in Section III.

Theorem 2: The satisfiability problem for the formulae of the logic of subintervals, over all discrete models, is undecidable.

An overview of the proofs. One possible source of undecidability, and the one we are making use of, is the interaction of regularity and measurement. Consider the following example proposition:

Proposition 1: The problem:

For a given regular language $L \subseteq \Sigma^*$ and a given set $B \subseteq \Sigma^2$, do there exist a natural number n and a word $w \in L$ such that $|w|$ (the length of w) is greater than n and for each sub-word avb of w (where $a, b \in \Sigma$), if the length of avb is n, then $\langle a, b \rangle \in B$?

is undecidable.

The proposition is obvious – if we can make sure that any two symbols in the word, which are at distance n, are a "correct pair", then we can easily encode a Turing machine.

In Sections III-B and III-C, we show how is it possible, in the logic of subintervals, to encode any regular language.

But encoding the measurement is not that simple. The logic of subintervals is not able – as far as we know – to measure the length of each sub-word of w. We need to mark each endpoint of the measured interval by a symbol that does not occur inside this interval. This means that we can only afford a bounded number of measurements taking place at the same time. Imagine we had four identical hourglasses, which we are free to turn at any moment while reading consecutive symbols of the word. This would not be enough to directly encode a Turing machine, but still enough for undecidability. In Section III-A we describe a class of regular languages (one regular language for each Minsky machine) for which the possibility of such four simultaneous measurements leads to an undecidable non-emptiness problem. This property is stated in Lemma 1 which is a counterpart of Proposition 1.

In Section III-D, we define our measuring tool, which we call a cloud, and in Section III-E, which completes the proof of Theorem 1, we show how to use it. Actually, the idea here is very simple: how much you see is a monotonic function of how high you are.

In the proof of Theorem 1, the measuring device, the cloud, is existentially quantified. Its role is identical with the role of the number n in Proposition 1. "They" provide it, together with the word w, and we only check that all the specification conditions are met. This approach would not work in the situation of Theorem 2. The reason for that is that the logic of subintervals gives no means (that we are aware of) to specify the requirement that all the intervals of the cloud are finite (i.e., contain a finite number of elements of the order). Or – using other words – that time periods measured by the hourglasses are finite. This could lead to pathologies that we do not even want to think about. Instead, in Section IV, we build our own hourglass which we call the parabola. It is not as good as the cloud – its size increases from time to time. But a closer look at Lemma 1 shows that we can live with it. And, unlike the cloud, the parabola does not suffer from the possible pathologies of discrete orderings.

II. Preliminaries

Orderings. Originally, Halpern–Shoham logic was defined for any order that satisfy the "linear interval property", i.e. for each a, c_1, c_2, b if $a \leq c_1$, $a \leq c_2$, $c_1 \leq b$, and $c_2 \leq b$, then $c_1 \leq c_2$ or $c_2 \leq c_1$. In such orderings, when we restrict our attention to the operators that look only "inside" of an initial interval, such as D, the reachable part of the ordering is totally ordered. For that reason in the rest of this paper we consider only the total orderings.

As in [9], we say that a total order $\langle \mathbb{D}, \leq \rangle$ is *discrete* if each element is either minimal (maximal) or has a predecessor (successor); in other words for all $a, b \in \mathbb{D}$ if $a < b$, then there exist points a', b' such that $a < a'$, $b' < b$ and there exists no c with $a < c < a'$ or $b' < c < b$.

Semantics of the D fragment of logic HS (logic of subintervals). Let $\langle \mathbb{D}, \leq \rangle$ be a discrete ordered set [1]. An *interval* over $\mathbb{D}$ is a pair $[a, b]$, with $a, b \in \mathbb{D}$ and $a \leq b$. A *labeling* is a function $\gamma : \mathrm{I}(\mathbb{D}) \to \mathcal{P}(\mathcal{V}ar)$, where $\mathrm{I}(\mathbb{D})$ is the set of all intervals over $\mathbb{D}$ and $\mathcal{V}ar$ is a finite set of propositional

[1] To keep the notation light, we will identify the order $\langle \mathbb{D}, \leq \rangle$ with its set $\mathbb{D}$

variables. A structure of the form $\mathrm{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ is called a *model*.

We say that an interval $[a, b]$ is a *leaf* iff it has no subintervals (i.e., $a = b$).

The truth values of formulae are determined by the following (natural) semantic rules:

1) For all $v \in \mathcal{V}ar$, we have $\mathrm{M}, [a, b] \models v$ iff $v \in \gamma([a, b])$.
2) $\mathrm{M}, [a, b] \models \neg\varphi$ iff $\mathrm{M}, [a, b] \not\models \varphi$.
3) $\mathrm{M}, [a, b] \models \varphi_1 \wedge \varphi_2$ iff $\mathrm{M}, [a, b] \models \varphi_1$ and $\mathrm{M}, [a, b] \models \varphi_2$.
4) $\mathrm{M}, [a, b] \models \langle D \rangle \varphi$ iff there exists an interval $[a', b']$ such that $\mathrm{M}, [a', b'] \models \varphi$, $a \leq a'$, $b' \leq b$, and $[a, b] \neq [a', b']$.

Boolean connectives $\vee, \Rightarrow, \Leftrightarrow$ are introduced in the standard way. We abbreviate $\neg\langle D \rangle \neg\varphi$ by $[D]\varphi$ and $\varphi \wedge [D]\varphi$ by $[G]\varphi$.

Note that we use the proper subinterval relation D (the prefixes and suffixes are treat as subintervals), but our technique works also in the strict case, where instead of $[a, b] \neq [a', b']$ we assume that $a \neq a'$ and $b \neq b'$ — see Section V. On the other hand, if we remove the condition $[a, b] \neq [a', b']$, then the problem is known to be decidable[12].

A formula φ is said to be *satisfiable* in a class of orderings $\mathcal{D}$ if there exist a structure $\mathbb{D} \in \mathcal{D}$, a labeling γ, and an interval $[a, b]$, called *the initial interval*, such that $\langle \mathrm{I}(\mathbb{D}), \gamma \rangle, [a, b] \models \varphi$. A formula is satisfiable in a given ordering $\mathbb{D}$ if it is satisfiable in $\{\mathbb{D}\}$.

III. Proof of Theorem 1

In Section III we only consider finite orderings.

Our representation. We imagine the Kripke structure of intervals of a finite ordering as a directed acyclic graph, where intervals are vertices and each interval $[a, b]$ of length greater than 0 has two successors: $[a + 1, b]$ and $[a, b - 1]$. Each level of this representation contains intervals of the same length (see Fig. 1).

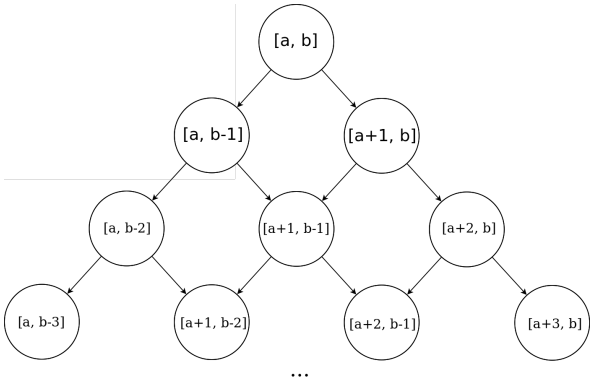

Figure 1. Our representation of order $\langle \{a, a + 1, \ldots, b\}, \leq \rangle$.

A. The Regular Language L_A

In this section, for a given two-counter finite automaton (Minsky machine) A we will define a regular language L_A. There is nothing about the logic of subintervals in this section – we are just preparing an undecidable problem which will be handy to encode.

Let Q be the set of states of A, and let $Q' = \{q' : q \in Q\}$. Define $B = \{f, f_r, s, s_r, x\}$ and $B' = \{b' : b \in B\}$

The alphabet Σ of L_A will consist of all the elements of $Q \cup Q'$ (jointly called *states*) and of all the subsets (possibly empty) of B and of B'. Talking about the subsets of B and B', we will not respect types, saying for example "f_r occurs in the the word v" rather than "there is a symbol in v that contains f_r".

Symbols (of Σ containing) f and f' (s and s') will be called *first* (resp., *second*) *counters*. Symbols f_r and f'_r (s_r and s'_r) will be called *first* (resp., *second*) *shadows* (or *shadows of the first/the second counter*). Symbols x and x' will be called X-symbols.

The language L_A consists of the words w over Σ that satisfy the following seven conditions:

C1. The first symbol of w is the initial state q_0 of A and the last symbol of w is either q or q', where q is one of the final states of A.

By a *configuration*, we will mean a maximal sub-word[2] of w, whose first element is a state (called *the state of the configuration*) and which contains exactly one state (so that w is split into disjoint configurations). A configuration will be called *even* if its state is from Q and *odd* if it is from Q'.

C2. Odd and even configurations alternate in w. All the non-state symbols occurring in even configurations are subsets of B and all the non-state symbols occurring in odd configurations are subsets of B'.

C3. Each configuration, except for the last one (which only consists of a state) contains exactly one first counter and exactly one second counter.

We want a word from L_A to encode a sequence of configurations of A which, once an additional distance constraint is satisfied (see Lemma 1), will be a correct accepting computation of A. So, except for a state of A, in each configuration, we need to remember the values of the two counters. We define the *value of the first counter* of a configuration as the number of symbols (strictly) between the state of the configuration and its first counter. The same applies to the second counter.

Example. A configuration with the state q, the first counter set to 3, and the second counter set to 4 can be stored as a word $q\emptyset\emptyset\emptyset\{f, f_r, s_r\}\{s\}\emptyset\emptyset\{x\}\emptyset\emptyset\emptyset$ (the meaning of f_r, s_r, and x will be defined later).

[2]By a sub-word, we mean a sequence of consecutive elements of a word, an infix (or prefix or suffix.

Using this language, we can state:

C4. In the first configuration, the value of both the counters is zero.

Which can also be read as: The second symbol of w contains f and s.

Now the role of shadows is going to be revealed:

C5. There are no shadows in the first and the last configuration. Each configuration, except for the first and the last, contains exactly one first shadow and exactly one second shadow.

In reading the next condition, it is good to have in mind that the position of a shadow in a given configuration, relative to the state of the configuration, will be enforced, by the distance constraints of Lemma 1, to be the same as the position of the corresponding counter in the previous configuration.

Since the format of an instruction of A is:

```
If in state q
the first counter
equals/does not equal 0 and
the second counter
equals/does not equal 0
then change the state to q1 and
decrease/increase/keep unchanged
the first counter and
decrease/increase/keep unchanged
the second counter.
```

it is clear what we mean, saying that *configuration C in word w matches the assumption of the instruction I*.

C6. If C and C_1 are consecutive configurations in w, and C matches the assumption of an instruction I, then:
- If I changes the state into q_1, then the state of C_1 is q_1.
- If I orders the first (second) counter to remain unchanged, then the first (resp., second) counter in C_1 coincides with the first (resp., second) shadow in C_1.
- If I orders the first (second) counter to be decreased, then the first (resp., second) counter in C_1 is the immediate predecessor of the first (resp., second) shadow in C_1.
- If I orders the first (second) counter to be increased, then the first (resp., second) counter in C_1 is the immediate successor of the first (resp., second) shadow in C_1.

One remaining condition is the following:

C7. There is exactly one x in each even configuration. All the counters and shadows of the same configuration are to the left of x. There are exactly 3 empty symbols between each x and the following state symbol. The same holds for odd configurations and x'.

This completes the definition of the language L_A. It is clear that it is regular – each of the seven conditions above can be checked by a very small finite automaton. Before we formulate Lemma 1, which will be our main tool, we need one more definition:

Definition 1: Let $w \in L_A$ and let cvd be a sub-word of w, (where $c, d \in \Sigma$). We will call cvd *an interesting infix* if there is exactly one X symbol in v and one of the following conditions holds:

1) c and d are states;
2) c is a first counter and d is a shadow of the first counter;
3) c is a second counter and d is a shadow of the second counter.

Notice that the condition that there is exactly one X symbol in v is a way of saying that positions of c and d belong to two consecutive configurations.

Lemma 1: The following two conditions are equivalent:

(i) Two-counter automaton A, starting from the initial state q_0 and empty counters, accepts.

(ii) There exists a word $w \in L_A$ and a natural number n such that the length of all the interesting infixes of w is n.

Proof: For the $\Rightarrow$ direction consider an accepting computation of A and take n as any number greater than all the numbers that appear on the two counters of A during this computation plus 6 (this is for X-symbols, states, empty symbols and the counters). For the $\Leftarrow$ direction, notice that the distance constraint from (ii) implies that the distance between a state and the subsequent first (second) shadow equals the value of the first (resp., second) counter in the previous configuration. Together with condition 5, defining L_A, this implies that the subsequent configurations in $w \in L_A$ can indeed be seen as subsequent configurations in the valid computation of A.

■

Since the halting problem for two-counter automata is undecidable, the proof of Theorem 1 will be completed once we write, for a given automaton A, a formula Ψ of the language of the logic of subintervals which is satisfiable (in a finite model) if and only if condition (ii) from Lemma 1 holds. Actually, what the formula Ψ is going to say is, more or less, that the word written (with the use of the labeling function γ) in the leaves of the model is a word w as described in Lemma 1, condition (ii).

In the following subsections, we are going to write formulae Φ_{orient}, Φ_{L_A}, Φ_{cloud}, and Φ_{length} such that $\Phi_{\text{orient}} \wedge \Phi_{L_A} \wedge \Phi_{\text{cloud}} \wedge \Phi_{\text{length}}$ will be the formula Ψ we want.

B. Orientation

As we said, we want to write a formula saying that the word written in the leaves of the model is the w described in Lemma 1, condition (ii).

The first problem we need to overcome is the symmetry of D – the operator does not see a difference between past and future, or between left and right, so how can we distinguish between the beginning of w and its end? We deal with this problem by introducing five variables L, R, s_0, s_1, s_2 and writing a formula Φ_{orient} which will be satisfied by an interval $[a, b]$ if $[a, a]$ is the only subinterval of $[a, b]$ that satisfies L and $[b, b]$ is the only subinterval of $[a, b]$ that satisfies R, or $[b, b]$ is the only subinterval of $[a, b]$ that satisfies L and $[a, a]$ is the only subinterval of $[a, b]$ that satisfies R, and all the following conditions hold:

- any interval that satisfies L or R satisfies also one of s_0, s_1, or s_2;
- each leaf is labeled either with s_0 or with s_1 or with s_2;
- each interval labeled with s_0 or with s_1 or with s_2 is a leaf;
- if c, d, e are three consecutive leaves of $[a, b]$ and if s_i holds in c, s_j holds in d and s_k holds in e then $\{i, j, k\} = \{0, 1, 2\}$;
- the initial interval has the length at least 3.

If $[a, b] \models \Phi_{\text{orient}}$, then the leaf of $[a, b]$ where L holds (resp., where R holds) will be called the left (resp., the right) end of $[a, b]$.

Let $exactly\_one\_of(X) = \bigvee_{x \in X}(x \wedge \bigwedge_{x' \in X \setminus \{x\}} \neg x')$ be a formula saying (which is not hard to guess) that exactly one variable from the set X is true in the current interval. Φ_{orient} is the conjunction of the following formulae:

(i) $\langle D\rangle\langle D\rangle\langle D\rangle\top$

(ii) $[G]((\![D]\!]\bot \Rightarrow exactly\_one\_of(\{s_0, s_1, s_2\})) \wedge (s_0 \vee s_1 \vee s_2 \Rightarrow [D]\bot))$

(iii) $[G](\langle D\rangle\langle D\rangle\top \Rightarrow \langle D\rangle s_0 \wedge \langle D\rangle s_1 \wedge \langle D\rangle s_2)$

(iv) $[G](L \vee R \Rightarrow s_0 \vee s_1 \vee s_2)$

(v) $\langle D\rangle R \wedge \langle D\rangle L$

(vi) $[G](L \Rightarrow \neg R)$

(vii) $\bigvee_{i \in \{0,1,2\}}(\langle D\rangle(L \wedge s_i) \wedge [D]([D][D]\bot \wedge \langle D\rangle L \Rightarrow \neg\langle D\rangle s_{(i-1) \bmod 3}))$

(viii) $\bigvee_{i \in \{0,1,2\}}(\langle D\rangle(R \wedge s_i) \wedge [D]([D][D]\bot \wedge \langle D\rangle R \Rightarrow \neg\langle D\rangle s_{(i+1) \bmod 3}))$

Formulae (i), (ii), (iii), and (iv) express the property defined by the conjunction of the five items above (notice, that $[D]\bot$ means that the current interval is a leaf).

Formula (v) says that there exists an interval labeled with R and an interval labeled with L.

Formula (vi) states that no interval satisfies both L and R.

Formula (vii) guarantees that no interval containing exactly 2 leaves, which is a superinterval of an interval labeled with L and s_i, can contain a subinterval labeled with $s_{(i-1) \bmod 3}$. It implies that an interval labeled with L can only have one superinterval containing exactly 2 leaves — if there were two, then their common superinterval containing 3 leaves would not have a subinterval labeled with $s_{(i-1) \bmod 3}$, thus contradicting (iii).

Finally, formula (viii) works like (vii) but for R.

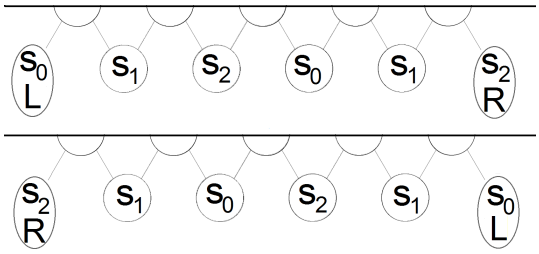

Figure 2. Two possible models that satisfy the formulae from Section III-B.

In the rest of paper, we restrict our attention to models satisfying formula Φ_{orient}, and treat the leaf labeled with L as the leftmost element of the model.

Notice that everything we did above can be applied not only to the whole model, but also to any subinterval of the model. We will say that set U *marks the left endpoint of interval* $[c, d]$ if some $u \in U$ holds in $[c, c]$ and no $u' \in U$ holds in any other subinterval of $[c, d]$. Analogously we define what it means that a set *marks the right endpoint of an interval*. What we proved in this section is:

Lemma 2: There exists a formula $mle(U)$ (and $mre(U)$) which is true in interval $[a, b]$ if and only if U marks the left (resp. the right) end of $[a, b]$.

Notice that we only know how to express the fact that $u \in U$ is valid in the left end of $[a, b]$ if u does not occur anywhere else in this interval.

C. Encoding a Finite Automaton

In this section, we show how to make sure that consecutive leaves of the model, read from L to R, are labeled with variables that represent a word of a given regular language.

Lemma 3: Let $\mathcal{A} = \langle \Sigma, \mathcal{Q}, q^0, \mathcal{F}, \delta\rangle$, where $q^0 \in \mathcal{Q}$, $\mathcal{F} \subseteq \mathcal{Q}$, $\delta \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$ be a finite–state automaton. There exists a formula $\psi_{\mathcal{A}}$ of the D fragment of Halpern–Shoham logic over alphabet $\mathcal{Q} \cup \Sigma$ that is satisfiable (with respect to the valuation of the variables from $\mathcal{Q}$) if and only if the word, over the alphabet Σ written in the leaves of the model, read from L to R, belongs to the language accepted by $\mathcal{A}$.

Proof: It is enough to write a conjunction of the following properties.

1) In every leaf, exactly one letter from Σ is satisfied (so there is indeed a word, written in the leaves). Moreover, the letters from Σ are true at leaves only.
2) Each leaf is labeled with exactly one variable from $\mathcal{Q}$. Moreover, the variable from $\mathcal{Q}$ are true at leaves only.
3) For each interval whose length is 1, if this interval contains an interval labeled with s_i, with $a \in \Sigma$, and with $q \in \mathcal{Q}$, and another interval labeled with $s_{(i+1) \bmod 3}$ and with $q' \in \mathcal{Q}$, then $\langle q, a, q' \rangle \in \delta$ (notice that we rely here on the assumption that Φ_{orient} holds in the model).
4) The interval labeled with R is labeled with such $q \in \mathcal{Q}$ and $a \in \Sigma$ such that $\langle q, a, q' \rangle \in \delta$ for some $q' \in \mathcal{F}$.
5) The interval labeled with L is labeled with q^0.

Clearly, a model satisfies properties 1-5 if and only if its leaves are labeled with an accepting run of $\mathcal{A}$ on the word over Σ written in its leaves. The formulae of the D fragment of Halpern–Shoham logic expressing properties 1-5 are not hard to write:

1) $[G](([D]\bot \Rightarrow exactly\_one\_of(\Sigma)) \wedge (\bigvee \Sigma \Rightarrow [D]\bot))$
2) $[G](([D]\bot \Rightarrow exactly\_one\_of(\mathcal{Q})) \wedge (\bigvee \mathcal{Q} \Rightarrow [D]\bot))$
3) $[G]([D][D]\bot \wedge \langle D\rangle s_i \wedge \langle D\rangle s_{i+1 \bmod 3} \Rightarrow \bigvee_{\langle q,a,q'\rangle \in \delta} \langle D\rangle(s_i \wedge q \wedge a) \wedge \langle D\rangle(s_{i+1 \bmod 3} \wedge q'))$, for each $i \in \{0, 1, 2\}$
4) $[G](R \Rightarrow \bigvee_{\langle q,a,q'\rangle \in \delta, q' \in \mathcal{F}}(q \wedge a))$
5) $[G](L \Rightarrow q^0)$

Now, let $\mathcal{A}$ be a finite automaton recognizing language L_A from Section III-A. We put $\Phi_{L_A} = \psi_{\mathcal{A}}$. ■

D. A Cloud

We still need to make sure that there exists n such that each configuration (but the last one) has length $n-1$ and that each interesting infix has length exactly n. Let us start with:

Definition 2: Let $\mathrm{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ be a model and p a propositional variable. We call p *a cloud* if there exists $k \in \mathbb{N}$ such that $p \in \gamma([a, b])$ if and only if the length of $[a, b]$ is exactly k.

So one can view a cloud as a set of all intervals of some fixed length. Notice, that if the current interval has length k then exactly $k+1$ leaves are reachable from this segment with the operator D.

We want to write a formula in the language of the D fragment of Halpern-Shoham logic saying that p is a cloud. In order to do that, we use an additional variable e. The idea is that an interval $[a, a+n]$ satisfies e iff $[a+1, a+n+1]$ does not.

Let Φ_{cloud} be the conjunction of the following formulae.

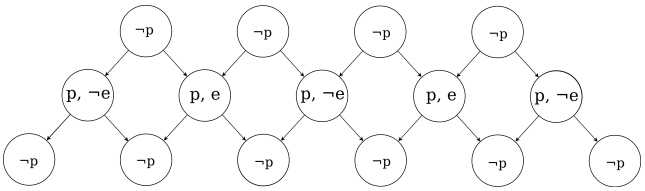

Figure 3. An example of a cloud.

1) $\langle D\rangle(p \wedge \langle D\rangle L)$ — there exists an interval that satisfies p and this interval contains the leftmost element of the model.
2) $[G](p \Rightarrow [D]\neg p)$ — intervals labeled with p cannot contain intervals labeled with p.
3) $[G](\langle D\rangle p \Rightarrow \langle D\rangle(p \wedge e) \wedge \langle D\rangle(p \wedge \neg e))$ — each interval that contains an interval labeled with p actually contains at least two such intervals — one labeled with e and one with $\neg e$.

Lemma 4: If $\mathrm{M}, [a_{\mathrm{M}}, b_{\mathrm{M}}] \models \Phi_{\text{cloud}}$, where a_{M} and b_{M} are endpoints of M, then p is a cloud.

Proof: We will prove that if an interval $[x, y]$ is labeled with p, then also $[x+1, y+1]$ is labeled with p. A symmetric proof shows that the same holds for $[x-1, y-1]$, so all the intervals of length equal to m, where m is the length of $[x, y]$, are labeled with p.

This will imply that no other intervals can be labeled with p and p is indeed a cloud. This is because each such interval either has a length greater than m, and thus contains an interval of length m, and as such labeled with p, or has a length smaller than m, and is contained in an interval labeled by p, in both cases contradicting 2.

Consider an interval $[x, y]$ labeled with p. Interval $[x, y+1]$ contains an interval labeled with p, so it has to contain two different intervals labeled with p – one labeled with e and the other one with $\neg e$. Suppose, without loss of generality, that $[x, y]$ is the one labeled with e, and let us call the second one $[u, t]$. If $t < y+1$, then $[u, t]$ is a subinterval of $[x, y]$ and is labeled with p, a contradiction. So $t = y+1$.

Let us assume that $u > x+1$. The interval $[u-1, y+1]$ must contain two different intervals labeled with p. One of them is $[u, y+1]$, and it cannot contain another interval labeled with p, so the other one must be $[u-1, y]$ or one of its subintervals. But then it is a subinterval of $[x, y]$ (because $u-1 > x+1-1 = x$) which also is labeled with p, but this leads to a contradiction. So $u = x+1$. ■

E. Using a cloud.

Let us now concentrate on models which satisfy $\Phi_{\text{orient}} \wedge \Phi_{L_A} \wedge \Phi_{\text{cloud}}$. Since Φ_{cloud} is satisfied, then p is a cloud. Let n denote the number of leaves contained in the intervals that form the cloud. Since Φ_{L_A} is satisfied, we know that the word written in the leaves of the model must belong to L_A. What remains to be done is writing a formula Φ_{length} that

would guarantee that the distance constraints from Lemma 1 are satisfied in this word.

The following lemma is just a restatement of Definition 1 in the language of the last paragraph of Section III-A:

Lemma 5: Let $w \in L_A$ and let v be a sub-word of w. Then v is an interesting infix if it contains exactly one X-symbol and one of the following conditions holds:

- one of the endpoints of v is marked with a state from Q and the other endpoint is marked with a state from Q';
- the left endpoint of v is marked with f (f', s, s') and the right endpoint is marked with f'_r (f_r, s'_r, s_r, resp.).

Using the formulae mle and mre from Section III-B, it is straightforward to translate the conditions of the lemma into a formula $interesting$ saying that the current interval is interesting:

$$
\begin{aligned}
& [G]((mle(Q) \wedge mre(Q') \vee (mle(Q') \wedge mre(Q)) \Rightarrow interesting) \\
\wedge\ & [G](mle(\{l \in \Sigma | f \in l\}) \wedge mre(\{l \in \Sigma | f'_r \in l\}) \Rightarrow interesting) \\
\wedge\ & [G](mle(\{l \in \Sigma | f' \in l\}) \wedge mre(\{l \in \Sigma | f_r \in l\}) \Rightarrow interesting) \\
\wedge\ & [G](mle(\{l \in \Sigma | s \in l\}) \wedge mre(\{l \in \Sigma | s'_r \in l\})) \Rightarrow interesting) \\
\wedge\ & [G](mle(\{l \in \Sigma | s' \in l\}) \wedge mre(\{l \in \Sigma | s_r \in l\}) \Rightarrow interesting)
\end{aligned}
$$

Note that the part about containing exactly one X-symbol comes for free here from the definition of the language and the properties of mle and mre. Now, we are ready to write Φ_{length}. It is the conjunction of the formula above and the following one:

$[G](interesting \Rightarrow p)$,

which means that if what you see is exactly an interesting interval, then you are exactly on the level of the cloud.

This ends the proof of Theorem 1.

IV. Proof of Theorem 2

The idea of the proof of Theorem 2 is exactly the same as of Theorem 1. But, because of the possible pathologies of discrete orders, almost all the details of the proof will now be much more complicated.

A. Damage assessment

Let us see which of the constructions form Section III can be saved in the new context.

Orientation In the new situation we still can, as we did in Section III-B, write formulae enforcing that the model has its left endpoint, marked with L, and its right endpoint, marked with R. But the trick with labeling each three consecutive elements with s_0, s_1 and s_2, which we used to define direction inside the model will, in the discrete case, orient only the locally finite fragments of the model (i.e., those maximal sets C of elements of the ordering such that, for each $a, b \in C$ the interval $[a, b]$ contains only finitely many elements).

On the other hand, if the model is infinite, then the left endpoint has its successor, which has a successor, etc. so that we have a copy of the ordered set of natural number as an initial fragment of the model. We will identify elements of this fragment with natural numbers. If formula Φ_{orient} is satisfied, then the set of natural numbers is oriented as in Section III-B.

It also turns out that we can actually force the model to be infinite. To do that, take a new variable nat, and write a formula Φ_{nat} saying that:

- nat only holds at leaves;
- L implies nat;
- if an interval contains two leaves, and in some of those two leaves nat holds, then it holds in all of them;
- there is an leaf where nat does not hold.

Let now Φ^d_{orient} be the formula $\Phi_{\text{orient}} \wedge \Phi_{nat}$. From now on, we assume that all the models under consideration satisfy Φ^d_{orient}.

The regular language L^d_A and the finite automaton. In the finite satisfiability case, the set of satisfiable formulae was recursively enumerable. Now it is co-re. This means that we now need, for a given Minsky machine A, to write a formula Ψ^d, of the logic of subintervals, which will be satisfiable if and only if A **does not** accept. We can assume that A has only one final (accepting) state q_f and that the machine runs forever if this state is not reached. So the formula we are going to write in this chapter should be satisfiable if and only if the machine A runs forever and never reaches q_f.

Since we still want to represent the computation of A as a word written in atoms of the model (to be more precise, in the atoms that are natural numbers), we must be ready to deal with an infinite word. The method the transition function of an automaton is encoded in Section III-C still works, so we can encode any automaton on infinite words with a "safety accepting condition", which means that it accepts a word if no forbidden state is entered during a run. Let L^d_A be the language of infinite words satisfying conditions [C1] – [C7] from Section III-A (with the obvious exception of the parts of conditions [C1], [C3] and [C5] which concern the final configuration) and additionally

C8. The sixth symbol of w is its first X-symbol.

Clearly, L^d_A can be recognized by an automaton with safety accepting condition. So we can write a formula $\Phi_{L^d_A}$ which will be satisfied in a model if and only if the word written in atoms being natural numbers belongs to L^d_A. Notice that $\Phi_{L^d_A}$ will be satisfied also by some words which only consist of finitely many configurations (last of them ending with infinitely many empty symbols). This cannot be prevented by a safety automaton, and we will need to find another way to forbid such words.

Lemma 1. The last remark leads to one change in Lemma

1. Another change will result from the fact that, in the new context we do not have the cloud anymore – the method it was defined does not translate to discrete orderings. So we no longer will be able to make sure that all the interesting infixes have the same length. But it turns out that we do not really need that much.

Definition 3: We say that an infinite word w is *nice* if for each pair v, u of interesting infixes such that v begins earlier than u, if k is the number of X-symbols between the left endpoint of v and the left endpoint of u then $|v| + k = |u|$.

The following version of Lemma 1 is easy to prove:

Lemma 6: The following two conditions are equivalent:

(i) Two-counter automaton A, started from the initial state q_0 and empty counters, runs forever.
(ii) There exists a nice word $w \in L_A^d$ with infinitely many X-symbols in w.

Notice that if there are two interesting infixes of a nice word, whose left ends are in the same configuration, then their lengths are equal. This is exactly what we need to be sure that the values of counters in each configuration are correctly reflected by the positions of shadows of the counters in the following configuration.

The second consequence of the fact that a word is nice is that the length of a subsequent configuration is always one plus the length of the previous one. This means that, if the first configuration was long enough to contain the values of the counters, then each configuration will be long enough, regardless of the possible unbounded growth of those values. And it follows from condition [C8] that the first configuration is long enough.

Having the idea on mind, proving Lemma 6 is straightforward.

B. The parabola

Let us remind that we identify the initial fragment of the model with the set $\mathbb{N}$.

Definition 4: Let $\mathrm{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ be a model and p, x, x' be a triple of variables. We call the triple p, x, x' *the parabola* if:

(i) only leaves are labeled with x or with x', the leaf $[6, 6]$ is labeled with x;

(ii) $[1, 10]$ is labeled with p;

(iii) if $[i, j]$ is labeled with p and $[i, i]$ is not labeled with an X-symbol, then $[i + 1, j + 1]$ is labeled with p;

(iv) if $[i, j]$ is labeled with p and $[i, i]$ is labeled with an X-symbol, then $[i + 1, j + 2]$ is labeled with p;

(v) if $[i, j]$ is labeled with p and x (resp., x') marks the left endpoint of $[i, j]$, then x' (resp., x) marks the right endpoint of $[i + 1, j + 1]$ (see Figure 4);

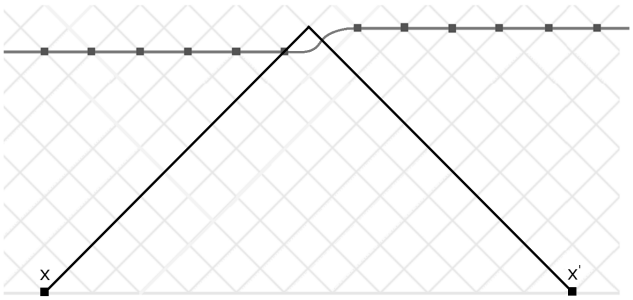

Figure 4. A fragment of the parabola.

(vi) no other interval whose left endpoint is a natural number are labeled with p.

Notice that the x, x' from the parabola coincide with the X-symbols from Σ, which means that if p, x, x' is the parabola then there are infinitely many X-symbols in w, and, in consequence, w consists of infinitely many configurations — a property that could not be enforced by a safety automaton alone.

Lemma 7: Let $\mathrm{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ be a model and p, x, x' be the parabola in M. Let $w \in L_A^d$ be the infinite word of symbols written in the natural numbers of M. Then the following two conditions are equivalent:

(i) w is nice;
(ii) each interesting infix of w is (as an interval) labeled with p.

Proof: Condition [C8] implies that the tenth symbol of w is a state symbol, and consequently, that $[1, 10]$ is the first interesting infix of w, and condition (ii) from the definition of the parabola implies that it is labeled by p.

Notice that conditions (iii) and (iv) from the definition of parabola guarantee that if $[i, j]$ is labeled with p, then there exists an intervals labeled with p that begins in $i+1$ and has the length greater by one than $[i, j]$ if there is an X-symbol in $[i, i]$ and has the same length as $[i, j]$ otherwise. It implies that the length of intervals labeled with p is increased by one with each X-symbol, so the length of two intervals labeled with p whose left ends are separated by k X-symbols differs by k — exactly as in the definition of a nice word. ■

In view of the last lemma, there are just two things that remain to be done. We need to write a formula Φ^d_{par} saying that p, x, x' is the parabola and write a formula Φ^d_{length}, saying that each interesting infix of w is labeled with p. Once we have them, we can finish the proof of Theorem 2 defining Ψ^d as:

$$\Phi^d_{\mathrm{orient}} \wedge \Phi_{L_A^d} \wedge \Phi^d_{\mathrm{par}} \wedge \Phi^d_{\mathrm{length}}$$

Formula Φ^d_{length} is easy to write.
$[G](\mathit{interesting} \Rightarrow p) \wedge [G](\langle D \rangle p \Rightarrow \bigvee_{q \in Q} \langle D \rangle q \wedge \bigvee_{q' \in Q'} \langle D \rangle q')$

The idea behind the formula Φ^d_{par} is simple — we use an auxiliary variable p_E to mark each interval $[x, y]$ such that $[x-1, y-1]$ is labeled with p and $[x, x]$ is labeled with an X-symbol, then we use p_E to guarantee that the interval $[x, y+1]$ is labeled with p. Those properties can be expressed using the same tricks as in the finite case, but now the resulting formula is a little bit more complicated (see the full version of the paper for details).

V. Other results

The trivial observation is that out technique still works if we exclude intervals of the form $[a, a]$. It also easy to see that it works if we allow the partial orders that match the definition from [9]. Two cases are more interesting — when we use strict D and when we use a superinterval relation $\bar{D}$.

A. Strict D

The strict D, denoted as $D_\subset$, is defined as follows.

$\mathrm{M}, [a, b] \models \langle D_\subset \rangle \varphi$ iff there exist an interval $[a', b']$ such that $\mathrm{M}, [a', b'] \models \varphi$ and $a < a' \leq b' < b$.

Our result holds also for $D_\subset$, there are just some minor technical details to handle. Here we will only describe how we label the leaves with a special variable l in that case — the remaining modifications are similar and are left to the reader.

In the D case, the labeling of leaves is easy — the formula $[G]([D]\bot \Leftrightarrow l)$ does it. But in the $D_\subset$ case, a similar formula would label also the intervals of length 1. To avoid it, we use auxiliary variables a, b, c, A, B and the conjunction of the following properties:

(i) Each interval of length at most 1 is labeled with exactly one of a, b, c, A, B.
(ii) No interval of length greater than 1 is labeled by a, b, c, A, or B.
(iii) Each interval of length at least 2 contains an interval labeled with a, b, or c.
(iv) Each interval of length at least 4 contains intervals with all five auxiliary symbols.

Condition (iii) guarantees that the intervals of length 0 are cannot be labeled with A or B. Observe that the intervals of length 4 contain exactly 2 strict subintervals of length 1 and exactly 3 strict subintervals of length 0, and due to (iv) those 5 intervals have to contain 5 auxiliary symbols, so the intervals of length 1 have to be labeled with A and B.

The formulae expressing properties (i)-(iv) are easy to express using $D_\subset$. Now, we can conclude the formula with conjunct $[G](l \Leftrightarrow a \vee b \vee c)$.

B. Superinterval relation

Our theorems hold also for $\bar{D}$ instead of D. But the changes to the proof need to be more significant. Consider, for example, formulae $[D]\langle D\rangle\top$ and $[\bar{D}]\langle\bar{D}\rangle\top$. The first one is not satisfied in the discrete case, but the second one is satisfied over, e.g., $\mathbb{N}$. In other words, the leaves are not always well-defined in case of $\bar{D}$ and we have to pay more attention while encoding a regular language.

To handle it, we use a cloud to define *pseudo-leaves* — a set of intervals on the same level that satisfies a special variable $leaf$. Once we defined leaves, we guarantee that nothing wrong happens above the leaves, i.e. $[\bar{D}](leaf \Rightarrow [\bar{D}] \bigwedge_{v \in \mathcal{V}ar \setminus \{leaf\}} \neg v)$. Finally, we can simply use the formula from the proof of Theorem 2, replacing D by $\bar{D}$, $[D]\bot$ by $leaf$ and so on, to proof undecidability. Again, details will be presented in the full paper.

C. Arbitrary orderings

The question whether the D fragment is decidable over the class of all (total) orderings is still open. However, our technique can be used to proof the following proposition.

Proposition 2: The satisfiability problem for the formulae of the $D\bar{D}$ fragment of Halpern–Shoham logic over the class of all total orderings is undecidable.

Proof: For the strict D case, consider the formula φ defined as follows

$$[G]([D_\subset]\bot \Rightarrow \langle \bar{D}_\subset\rangle([D][D]\bot)).$$

This formula is satisfiable in orderings such that for each reachable interval $[a, a]$ there exist b, c such that $b < a < c$ and the interval $[b, c]$ contains at most 4 points (including a, b, c). It implies that a has both predecessor and successor. Therefore the reachable part of the ordering is discrete.

Now we would like to say that all discrete orderings satisfy φ, no matter which initial interval we choose. It is not entirely true — the formula is not satisfied if the interval $[x, x]$ is reachable, where x is the maximal or the minimal point. But this is the case only if the interval $[x, x]$ is initial, so we can simply fix that: let $\varphi' = \varphi \vee ([D_\subset]\bot \wedge [\bar{D}_\subset]\bot)$.

Now we can use the formula $\Psi^d \wedge \varphi'$ (where Ψ^d is the formula from the proof of the undecidability of the D fragment in the discrete case) to proof the undecidability.

The proper D case can be solved in the same way, however the proof is much more technical. ■

The proof, however easy, bases on the fact that we allow the intervals of the form $[a, a]$. The question of what happens if we exclude such intervals remains open.

Acknowledgment

This research work was supported by Polish Ministry of Science and Higher Education research project N N206 371339.

References

[1] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (11) (1983) 832-843.

[2] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, G. Sciavicco, Decidable and Undecidable Fragments of Halpern and Shoham's Interval Temporal Logic: Towards a Complete Classification, in: Proc. of 15th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning, Vol. 5330 of LNCS, Springer, 2008, pp. 590-604.

[3] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, G. Sciavicco, Undecidability of the Logic of Overlap Relation over Discrete Linear Orderings. Proceedings of M4M 6: 6th Workshop on Methods for Modalities, November 2009.

[4] D. Bresolin, V. Goranko, A. Montanari, P. Sala, Tableau-based decision procedures for the logics of subinterval structures over dense orderings. Journal of Logic and Computation, vol. 20, n. 1, 2010, pp. 133-166.

[5] D. Bresolin, V. Goranko, A. Montanari, G. Sciavicco, Propositional Interval Neighborhood Logics: Expressiveness, Decidability, and Undecidable Extensions, Annals of Pure and Applied Logic, Vol.161(3), 2009, pp. 289-304.

[6] D. Bresolin, A. Montanari, P. Sala, G. Sciavicco, Optimal Tableaux for Right Propositional Neighborhood Logic over Linear Orders, in: Proc. of the 11th European Conference on Logics in AI, Vol. 5293 of LNAI, Springer, 2008, pp. 62-75.

[7] V. Goranko, A. Montanari, and G. Sciavicco, A road map of interval temporal logics and duration calculi. Journal of Applied Non-Classical Logics, 14(1-2):9-54, 2004.

[8] C. L. Hamblin, Instants and intervals. Studium Generale, 27:127-134, 1971.

[9] J. Halpern, Y. Shoham, A propositional modal logic of time intervals, Journal of the ACM 38 (4) (1991) 935-962.

[10] K. Lodaya, Sharpening the undecidability of interval temporal logic. In Proc. of 6th Asian Computing Science Conference, volume 1961 of LNCS, pages 290-298. Springer, 2000.

[11] J. Marcinkowski, J. Michaliszyn, and E. Kieronski, B and D Are Enough to Make the Halpern–Shoham Logic Undecidable. In Proc. 37th International Colloquium on Automata, Languages and Programming, ICALP 2010, Proceedings, Part II, LNCS 6199, pages 357-368.

[12] A. Montanari, I. Pratt-Hartmann and P. Sala, Decidability of the Logic of the Reflexive Subinterval Relation over Finite Linear Orders, Proceedings of the 17th International Symposium on Temporal Representation and Reasoning (TIME), Paris, France, September 2010, pp. 27-34.

[13] A. Montanari, G. Puppis, P. Sala, A decidable spatial logic with cone-shaped cardinal directions, in: 18th Annual Conference of the EACSL, Vol. 5771 of LNCS, 2009, pp. 394-408.

[14] B. C. Moszkowski, Reasoning about Digital Circuits. PhD thesis, Stanford University, Computer Science Department, July 1983.

[15] A. Montanari, Gabriele Puppis, and Pietro Sala, Maximal Decidable Fragments of Halpern and Shoham's Modal Logic of Intervals. In Proc. 37th International Colloquium on Automata, Languages and Programming, ICALP 2010, Proceedings, Part II, LNCS 6199, pages 345-356.

[16] R. Parikh. A decidability result for second order process logic. In Proc. 19th FOCS, pages 177-183. IEEE, October 1978.

[17] V. R. Pratt. Process logic. In Proc. 6th POPL, pages 93-100. ACM, January 1979.

[18] T. Schwentick, T. Zeume: Two-Variable Logic with Two Order Relations – (Extended Abstract). In Proc. 24th International Workshop of Computer Science Logic, LNCS 6247, pages 499-513.