Fundamenta Informaticae XX (2013) 1–24
DOI 10.3233/FI-2012-0000
IOS Press

The Undecidability of the Logic of Subintervals

Jerzy Marcinkowski
Institute of Computer Science
University Of Wrocław
jma@cs.uni.wroc.pl

Jakub Michaliszyn
Institute of Computer Science
University Of Wrocław
jmi@cs.uni.wroc.pl

Keywords: Halpern–Shoham logic, logic of subintervals, undecidable logic

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 and its reflexive variant is known to be decidable over discrete structures. Our result subsumes a lot of previous undecidability results of fragments that include D.

1. Introduction

In classical temporal logic, structures are defined by assigning properties (propositional variables) to time points (time is assumed to be an ordered set of points, 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 [11], 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., [12]. In computer science, the earliest attempts to formalize time intervals were process logic [21, 22] and interval temporal logic [20]. 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 $\bar{B}$. The expressive power of HS operators has been studied in [16].

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 [14] that the BE fragment is undecidable in the dense case. Recently, it was shown that the satisfiability problem for the HS fragments O^*, A^*D^*, B^*E^* is undecidable in any class of linear orders that contains, for each $n > 0$, at least one linear order with length greater than n [6], where each X^* may be replaced by X or $\bar{X}$ (i.e. "A^*D^* is undecidable" means that AD, $A\bar{D}$, $\bar{A}D$, and $\bar{A}\bar{D}$ are undecidable). The fragment BD was shown to be undecidable over discrete structures. [15].

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 [8]. The fragment using only A and $\bar{A}$ is harder and its decidability was only recently shown [5, 7]. Obviously, this last result implies decidability of $L\bar{L}$ as L is expressible by A. Another fragment known to be decidable is $AB\bar{B}\bar{L}$ [19].

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, 18] that satisfiability of formulae of logic of subintervals is decidable over dense structures. In [17] decidability is proved for the (slightly less expressive) "reflexive D". The results in [23] 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 [11], BD [15] and ADB, $A\bar{A}D$ [3, 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.

1.1. Main theorems and an overview of the proofs

As in [11], we say that an order $\langle \mathbb{D}, \leq \rangle$ is (weakly) *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 $b' < c < b$. We say that an order is *strongly discrete* if each interval of this order is finite, i.e., contains only finitely many points.

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

Theorem 1.1. The satisfiability problem for the formulae of the logic of subintervals is undecidable in any class of strongly discrete orders that contains, for each $n > 0$, at least one chain with length greater than n.

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 1.1 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 3.

Theorem 1.2. The satisfiability problem for the formulae of the logic of subintervals, over any class of discrete models that contains at least one order containing infinite interval, is undecidable.

An overview of the proofs. A popular source of undecidability, and the one we are making use of, is the interaction of regularity and measurement. Consider the following example problem:

Proposition 1.1. The following problem is undecidable.

For a given regular language $L \subseteq \Sigma^*$ and a 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 $\langle a, b \rangle \in B$?

The proposition can be easily proved by reducing the domino problem. In the reduction, each world of the regular language is meant to represent consecutive rows of the tiled grid, and the relation B controls vertical adjacencies. See [2, 10] for more details about the domino problems.

In Sections 3.2 and 3.3, we show how is it possible, in the logic of subintervals, to encode any regular language.

However, 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 five identical hourglasses, which we are free to turn at any moment while reading consecutive symbols of the word. This is enough for undecidability. In Section 3.1 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 3.2 which is a counterpart of Proposition 1.1.

In Section 3.4, we define our measuring tool, which we call a cloud, and in Section 3.5, which completes the proof of Theorem 1.1, we show how to use it.

In the proof of Theorem 1.1, the measuring device, the cloud, is existentially quantified. Its role is identical with the role of the number n in Proposition 1.1. This approach would not work in the situation of Theorem 1.2. 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. To avoid such pathologies, in Section 4, we build our own hourglass which we call the curve. It is not as good as the cloud – its size increases from time to time. But a closer look at Lemma 3.2 shows that we can live with it. And, unlike the cloud, the curve does not suffer from the possible pathologies of discrete orderings.

2. 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.

Semantics of the D fragment of logic HS (logic of subintervals). Let $\langle \mathbb{D}, \leq \rangle$ be a discrete ordered set. To keep the notation light, we will identify the order $\langle \mathbb{D}, \leq \rangle$ with its set $\mathbb{D}$. 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 variables. A structure of the form $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ is called a *model*.

We say that an interval $[a, b]$ is a *point interval* 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 $\mathfrak{M}, [a, b] \models v$ iff $v \in \gamma([a, b])$.
2. $\mathfrak{M}, [a, b] \models \neg\varphi$ iff $\mathfrak{M}, [a, b] \not\models \varphi$.
3. $\mathfrak{M}, [a, b] \models \varphi_1 \wedge \varphi_2$ iff $\mathfrak{M}, [a, b] \models \varphi_1$ and $\mathfrak{M}, [a, b] \models \varphi_2$.
4. $\mathfrak{M}, [a, b] \models \langle D \rangle\varphi$ iff there exists an interval $[a', b']$ such that $\mathfrak{M}, [a', b'] \models \varphi$, $a \leq a'$, $b' \leq b$, 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 6. On the other hand, if we remove the condition $[a, b] \neq [a', b']$, then the problem is known to be decidable[17].

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 a singleton class $\{\mathbb{D}\}$.

Useful formulae. We will often use the formulae of the form λ_i that are satisfied in the intervals with the specific length. We define those formulae as $\lambda_i = \langle D^i \rangle \top \wedge \neg \langle D^{i+1} \rangle \top$, where the exponentiation D^k is defined as usual: $\langle D^0 \rangle \phi = \phi$, and $\langle D^k \rangle \phi = \langle D \rangle \langle D^{k-1} \rangle \phi$ for all $k > 0$. It is readily checked that λ_k is as required — for example, $\lambda_0 = \top \wedge [D] \bot$ — it is satisfied in the intervals with no subintervals, i.e. the intervals with the length 0.

3. Proof of Theorem 1.1

In Section 3 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). As usual, each vertex is associated with a subset of propositional variables.

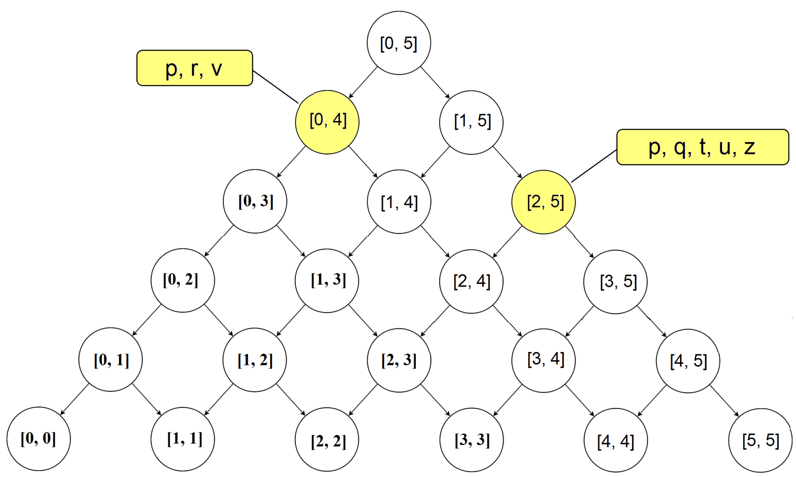

Figure 1. Our representation of order $\langle \{0, 1, \dots, 5\}, \leq \rangle$.

3.1. 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 $B = \{f, f_r, s, s_r\}$ and $B' = \{f', f'_r, s', s'_r\}$

The alphabet Σ of L_A will consist of all the elements of $Q \cup Q'$ (jointly called *states*), symbols x and x' (jointly called X-symbols) 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 of Σ containing f_r and f'_r (s_r and s'_r) will be called *first* (resp., *second*) *shadows*.

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 (i.e. a sequence of consecutive elements of a word) 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 3.2), 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 $q\emptyset\emptyset\emptyset\{f, f_r, s_r\}\{s\}\emptyset\emptyset x$ (the meaning of f_r, s_r, and x will be defined later).

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.

Shadows are meant to represent values of counters in the previous configuration. We cannot, however, force such a property in the definition of the regular language L_A. As for now, we define only some basic properties of shadows.

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 3.2, 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 the state is q
 and the first counter equals/does not equal 0
 and the second counter equals/does not equal 0
then change the state to q'
 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. Each x is followed by a 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 3.2, which will be our main tool, we need one more definition:

Definition 3.1. 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 first shadow;
3. c is a second counter and d is a second shadow.

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 3.2. 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.

Using the language from the introduction, the number n corresponds to the size of hourglasses. If the property (ii) is satisfied, then each letter of w is a member of at most five interesting infixes — one starting with a state, and four starting with f, f', s, and s'. This corresponds to the five hourglasses mentioned before.

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 3 (this is for X-symbols, states 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. $\square$

Since the halting problem for two-counter automata is undecidable, the proof of Theorem 1.1 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 3.2 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 point intervals of the model is a word w as described in Lemma 3.2, 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.

3.2. Orientation

As we said, we want to write a formula saying that the word written in the point intervals of the model is the w described in Lemma 3.2, 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 point interval 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 point interval;
- if c, d, e are three consecutive point intervals 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 point interval of $[a, b]$ where L holds (resp., where R holds) will be called the left (resp., the right) end of $[a, b]$.

Let $exactly\_one\_of(Y) = \bigvee_{y \in Y}(y \wedge \bigwedge_{y' \in Y \setminus \{y\}} \neg y')$ be a formula saying (which is not hard to guess) that exactly one variable from the set Y 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) $[G]((\lambda_0 \Rightarrow exactly\_one\_of(\{s_0, s_1, s_2\})) \wedge (s_0 \vee s_1 \vee s_2 \Rightarrow \lambda_0))$

(iii) $[G](\lambda_2 \Rightarrow \langle D\rangle s_0 \wedge \langle D\rangle s_1 \wedge \langle D\rangle s_2)$

(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) $\bigvee_{i\in\{0,1,2\}}(\langle D\rangle(L \wedge s_i) \wedge [D](\lambda_1 \wedge \langle D\rangle L \Rightarrow \neg\langle D\rangle s_{(i-1)\ \mathtt{mod}\ 3}))$

(viii) $\bigvee_{i\in\{0,1,2\}}(\langle D\rangle(R \wedge s_i) \wedge [D](\lambda_1 \wedge \langle D\rangle R \Rightarrow \neg\langle D\rangle s_{(i+1)\ \mathtt{mod}\ 3}))$

Formulae (i), (ii), (iii), and (iv) express the property defined by the conjunction of the five items above (notice, that λ_0 means that the current interval is a point interval).

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 point intervals, which is a superinterval of an interval labeled with L and s_i, can contain a subinterval labeled with $s_{(i-1)\ \mathtt{mod}\ 3}$. It implies that an interval labeled with L can only have one superinterval containing exactly 2 point intervals — if there were two, then their common superinterval containing 3 point intervals would not have a subinterval labeled with $s_{(i-1)\ \mathtt{mod}\ 3}$, thus contradicting (iii).

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

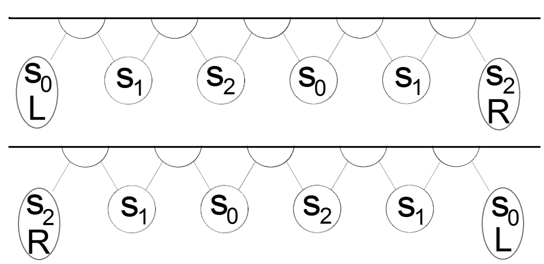

Figure 2. Two possible models that satisfy the formulae from Section 3.2.

In the rest of paper, we restrict our attention to models satisfying formula Φ_{orient}, and treat the point interval 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 3.3. 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 $[a, b]$.

The precise definitions of $mle(U)$ and $mre(U)$ are as follows.

$$mle(U) = \bigvee_{i\in\{0,1,2\}} (\langle D\rangle(\bigvee U \wedge s_i) \wedge [D](\lambda_1 \wedge \langle D\rangle \bigvee U \Rightarrow \neg\langle D\rangle s_{(i-1) \texttt{ mod } 3}))$$

$$mre(U) = \bigvee_{i\in\{0,1,2\}} (\langle D\rangle(\bigvee U \wedge s_i) \wedge [D](\lambda_1 \wedge \langle D\rangle \bigvee U \Rightarrow \neg\langle D\rangle s_{(i+1) \texttt{ mod } 3}))$$

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. This is why we encode Minsky machine rather than Turing machine.

3.3. Encoding a Finite Automaton

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

Lemma 3.4. 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 point intervals 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 point interval, exactly one letter from Σ is satisfied (so there is indeed a word, written in the point intervals). Moreover, the letters from Σ are true at point intervals only.

2. Each point interval is labeled with exactly one variable from $\mathcal{Q}$. Moreover, the variable from $\mathcal{Q}$ are true at point intervals only.

3. For each interval whose length is 1, if this interval contains an interval labeled with s_i, a and q, and another interval labeled with $s_{(i+1) \texttt{ mod } 3}$ and q', where $i \in \{0, 1, 2\}$, $a \in \Sigma$, and $q, q' \in \mathcal{Q}$, 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 q and a 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 point intervals are labeled with an accepting run of $\mathcal{A}$ on the word over Σ written in its point intervals. The formulae of the D fragment of Halpern–Shoham logic expressing properties 1-5 are not hard to write:

1. $[G]((\lambda_0 \Rightarrow exactly\_one\_of(\Sigma)) \wedge (\bigvee \Sigma \Rightarrow \lambda_0))$

2. $[G]((\lambda_0 \Rightarrow exactly\_one\_of(\mathcal{Q})) \wedge (\bigvee \mathcal{Q} \Rightarrow \lambda_0))$

3. $[G](\lambda_1 \wedge \langle D\rangle s_i \wedge \langle D\rangle s_{i+1 \texttt{ mod } 3} \Rightarrow \bigvee_{\langle q,a,q'\rangle\in\delta} \langle D\rangle(s_i \wedge q \wedge a) \wedge \langle D\rangle(s_{i+1 \texttt{ mod } 3} \wedge q'))$, 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 3.1. We put $\Phi_{L_A} = \psi_{\mathcal{A}}$. $\square$

3.4. 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 3.5. Let $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma\rangle$ 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 $k+1$ point intervals 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.

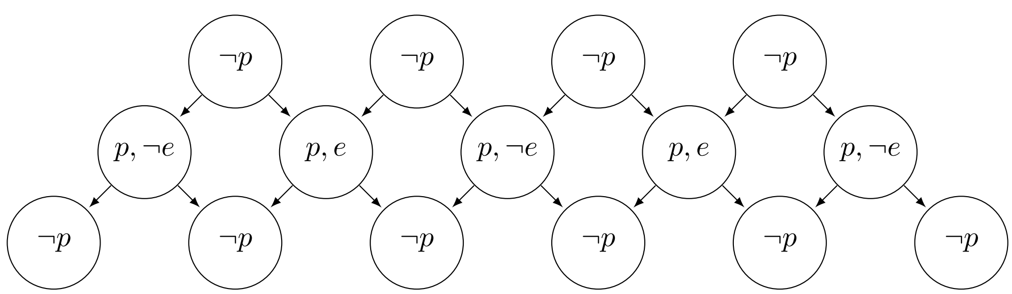

Figure 3. An example of a cloud.

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

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 3.6. If $\mathfrak{M}, [a_{\mathfrak{M}}, b_{\mathfrak{M}}] \models \Phi_{\text{cloud}}$, where $a_{\mathfrak{M}}$ and $b_{\mathfrak{M}}$ are endpoints of $\mathfrak{M}$, 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 $u = x + 1$. □

3.5. Using a cloud.

3.5.1. An example

Before we proceed into the technical aspects of encoding the automata, let us show a simple example of the usage of a cloud.

Let $\mathcal{A}$, be a finite automaton that recognizes the language defined by the regular expression $(ac^*bc^*)^*$. Consider the formula $\Phi_{\text{orient}} \wedge \Phi_{L_{A_0}}$. In each model of ρ, point intervals contain some word w accepted by $\mathcal{A}$. Our goal is to force the following property.

(e) There exists $n \in \mathbb{N}$ such that each maximal block c that is between a and b has the length n.

We use a cloud. In fact, n will be equal to the length of the intervals in the cloud. Property (e) can be expressed as a conjunction of Φ_{cloud} and the following formulae.

(i) $[G](p \Rightarrow \neg(\langle D\rangle a \wedge \langle D\rangle b))$

(ii) $[G](p \Rightarrow \langle D\rangle(a \vee b))$

Now, consider two examples in Figure 4. At the left side, the red vertex contains no subinterval that satisfies a or b — it contradicts Formula (ii). At the right side, the red vertex contain a subinterval that satisfies a and a subinterval that satisfies b — it contradicts Formula (i). Therefore, this block of c has to have the length 2, equal to the length of the intervals in the cloud.

3.5.2. Encoding two-counter automaton

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 point intervals contained in the intervals that form the cloud. Since Φ_{L_A} is satisfied, we know that the word written in the point intervals of the model must belong to L_A.

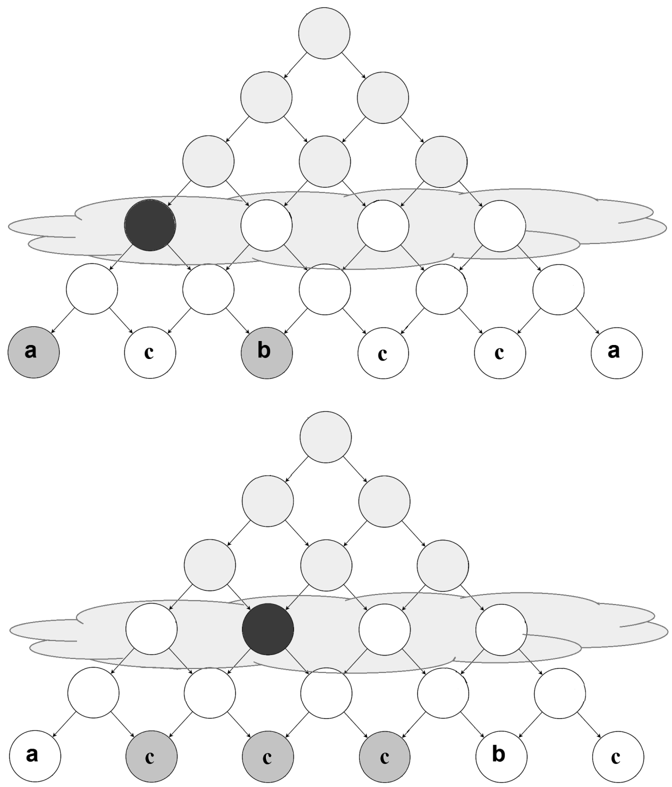

Figure 4. An example of using a cloud.

What remains to be done is writing a formula Φ_{length} that would guarantee that the distance constraints from Lemma 3.2 are satisfied in this word.

The following lemma is just a restatement of Definition 3.1 in the language of the last paragraph of Section 3.1:

Lemma 3.7. 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 3.2, it is straightforward to translate the conditions of the lemma into a formula $\Phi_{\mathrm{interesting}}$ that labels all the interesting intervals with the propositional variable $interesting$:

$$[G]((mle(Q) \wedge mre(Q') \vee (mle(Q') \wedge mre(Q)) \Rightarrow interesting)$$

$\wedge\ [G](mle(\{l \in \Sigma | f \in l\}) \wedge mre(\{l \in \Sigma | f'_r \in l\}) \Rightarrow interesting)$

$\wedge\ [G](mle(\{l \in \Sigma | f' \in l\}) \wedge mre(\{l \in \Sigma | f_r \in l\}) \Rightarrow interesting)$

$\wedge\ [G](mle(\{l \in \Sigma | s \in l\}) \wedge mre(\{l \in \Sigma | s'_r \in l\})) \Rightarrow interesting)$

$\wedge\ [G](mle(\{l \in \Sigma | s' \in l\}) \wedge mre(\{l \in \Sigma | s_r \in l\}) \Rightarrow interesting)$

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}.

$$\Phi_{\text{length}} = \Phi_{\text{interesting}} \wedge [G](interesting \Rightarrow p)$$

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.

4. Proof of Theorem 1.2

The idea of the proof of Theorem 1.2 is exactly the same as of Theorem 1.1. But, because of the possible pathologies of discrete orders, almost all the details of the proof will now be much more complicated.

4.1. Damage assessment

Let us see how much of the construction form Section 3 can be saved in the new context.

Orientation In the new situation we still can, as we did in Section 3.2, write formulae enforcing that the model has its left endpoint, marked with L, and its right endpoint, marked with R. However, the formula that we use to label each three consecutive elements with s_0, s_1 and s_2 will, in the arbitrary discrete case, label 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 3.2.

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

Those properties can by expressed using D in the following way:

- $[G](nat \Rightarrow \lambda_0)$
- $[G](L \Rightarrow nat)$;
- $[G](\lambda_1 \wedge \langle D \rangle nat \Rightarrow [D]nat)$;
- $\langle D \rangle(\lambda_0 \wedge \neg nat)$

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, as we will see later, it is coRE-hard. 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 3.3 still works, so we can encode any automaton on infinite words with a safety accepting condition. Such an automaton contains a set of rejecting states instead of a set of accepting states, and a (infinite) run of such automata is said to be accepting iff it visits no rejecting states. Let L^d_A be the language of infinite words satisfying conditions [C1] – [C7] from Section 3.1 (with the obvious exception of the parts of conditions [C1], [C3] and [C5] which concern the final configuration) and additionally

C8. The third 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 3.2. The last remark leads to one change in Lemma 3.2. 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 4.1. 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 3.2 is easy to prove:

Lemma 4.2. 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 $w \in L^d_A$ 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 corresponding shadows 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 4.2 is straightforward.

Example 4.3. Let $(q_0, 0, 0), (q_1, 1, 0), (q_2, 2, 1)$ be an initial fragment of an infinite run of an automaton, where a triple (q, f, s) represents a configuration in which the automaton is in a state q, its first counter has a value f, and its second counter has a value s. The world w that encodes this fragment is

Position	1	2	3	4	5	6	7	8	9	10	11	12
Letter	$\mathbf{q}_0$	$\{f, f_r, s, s_r\}$	x	$\mathbf{q}_1'$	$\{f_r', s', s_r'\}$	$\{f'\}$	x'	$\mathbf{q}_2$	$\{s_r\}$	$\{f_r, s\}$	$\{f\}$	x

Let us denote by $w[n : m]$ the fragment of w that starts at a position n and end at a position m. The following infixes of w are interesting: $w[1 : 4]$, $w[2 : 5]$. $w[4 : 8]$, $w[5 : 9]$, and $w[6 : 10]$. Clearly, the interesting infixes starting after the first x are longer by 1, as required in the definition of the niceness.

4.2. The curve

Let us remind that we identify the initial fragment of the model with the set $\mathbb{N}$. Unlike clouds, the curve is not a straight line. The curve may be seen as a variable that labels three intervals of length 3 ($[1, 4]$, $[2, 5]$, and $[3, 6]$), four intervals of length 4 ($[4, 8]$, $[5, 9]$, $[6, 10]$, and $[7, 11]$), and so on. Formally:

Definition 4.4. Let $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ be a model and p, p_E, x, x' be a quadruple of variables. We call the quadruple p, p_E, x, x' *the curve* if:

(i) only point intervals are labeled with x or with x', and the point interval $[3, 3]$ is labeled with x;

(ii) $[1, 4]$ is labeled with p;

(iii) if $[i, j]$ is labeled with p and $[i, i]$ is not labeled with x or with x', then $[i + 1, j + 1]$ is labeled with p;

(iv) each interval labeled with p contain a subinterval labeled with x or with x', but no such interval marks the right endpoint with x or with x';

(v) all intervals labeled with p_E mark the right endpoint with x or with x';

(vi) no subinterval of an interval labeled with p can contain a subinterval with x and a subinterval with x';

(vii) if $[i, j]$ is labeled with p and x (resp., x') marks the left endpoint of $[i, j]$, then $[i + 1, j + 1]$ is labeled by p_E (see Figure 5);

(viii) if $[i, j]$ is labeled with p_E, then $[i, j + 1]$ is labeled with p;

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

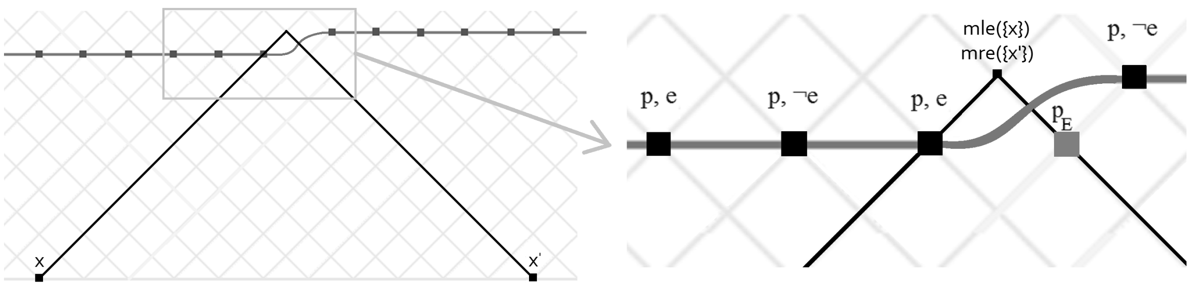

Figure 5. A fragment of the curve.

Notice that the x, x' from the curve coincide with the X-symbols from Σ, which means that if p, p_E, x, x' is the curve 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.

The variable p_E is auxiliary in some sense - we use it to mark in each row that contains p, the first column after the columns with p. In other words, an interval $[i, j]$ will be labeled with p_E if $[i-1, j-1]$ is labeled with p and $[i-1, i-1]$ is labeled with x or x' (so, therefore, $[i, j+1]$ should be labeled with p) — see Figure 5.

The role of variable p is as the role of cloud — we will use it to guarantee that the interesting infixes have the right length.

Lemma 4.5. Let $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ be a model and p, p_E, x, x' be the curve in $\mathfrak{M}$. Let $w \in L^d_A$ be the infinite word of symbols written in the natural numbers of $\mathfrak{M}$. Then the following two conditions are equivalent:

(i) w is a nice word with infinitely many X-symbols;

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

Proof:
Condition [C8] implies that the fourth symbol of w is a state symbol, and consequently, that $[1, 4]$ is the first interesting infix of w, and condition (ii) from the definition of the curve implies that it is labeled by p.

Notice that conditions from the definition of curve guarantee that if $[i, j]$ is labeled with p, then there exists an interval 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 $[i,i]$ (conditions (vi)-(viii)) and has the same length as $[i,j]$ otherwise (condition (iii)). 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. The condition (iv) guarantee that there are infinitely many X-symbols. □

The idea behind the formula Φ^d_{par} is simple. At first, note that we still can use formulae mle and mre defined exactly as in the definition of a cloud to define the curve.

Let us Φ^d_{par} be the conjunction of the following formulae (they correspond to the properties from Definition 4.4).

(i) $[G](x \vee x' \Rightarrow \lambda_0) \wedge \langle D\rangle(\lambda_3 \wedge \langle D\rangle L \wedge mre(\{x\}))$;

(ii) $\langle D\rangle(p \wedge \langle D\rangle L \wedge \lambda_3)$ (recall that by definition L is satisfied in [1, 1]);

(iii) $[G]((\langle D\rangle p \wedge \neg mle(\{x\}) \wedge \neg mle(\{x'\})) \Rightarrow \langle D\rangle(p \wedge e) \wedge \langle D\rangle(p \wedge \neg e))$ (we use an auxiliary variable e in the same way as for cloud);

(iv) $[G]((mre(\{x\}) \vee mre(\{x'\})) \Rightarrow \neg p) \wedge [G](p \Rightarrow \langle D\rangle(x \vee x'))$;

(v) $[G](p_E \Rightarrow mre(\{x\}) \vee mre(\{x'\}))$;

(vi) $[G](p \Rightarrow [D]([D]\neg x \vee [D]\neg x'))$;

(vii) $[G]((mle(\{y\}) \wedge \langle D\rangle(p \wedge mle\{y\}) \wedge [D][D]\neg p) \Rightarrow \langle D\rangle p_E \wedge [D][D]\neg p_E)$ for each $y \in \{x, x'\}$;

(viii) $[G](\langle D\rangle p_E \wedge [D][D]\neg p_E \wedge \neg(mle(\{x\}) \vee mre(\{x'\})) \Rightarrow p)$;

(ix) $[G](p \Rightarrow [D]\neg p)$ (it works as for cloud).

It turns out that we can simply put $\Phi^d_{\text{length}} = \Phi_{\text{length}}$. Now we can finish the proof of Theorem 1.2 defining

$$\Psi^d = \Phi^d_{\text{orient}} \wedge \Phi_{L^d_A} \wedge \Phi^d_{\text{par}} \wedge \Phi^d_{\text{length}}$$

5. Superinterval relation

Our theorems hold also for $\bar{D}$ instead of D. But the changes to the proof need to be 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., $\mathbb{N}$.

Another important difference between the models for the logic with D and for the logic with $\bar{D}$ is that in linear discrete orders all intervals have either 0 or 2 subintervals with the maximal size, while an interval may have only one superinterval with the minimal size (e.g. over the naturals, the interval $[1,5]$ has only one such superinterval — $[1,6]$).

In other words, the point intervals are not always well-defined in case of $\bar{D}$. There can be no interval that satisfies $[\bar{D}]\top$ in a model, and even if there is such interval, then it is the only one. Therefore we have to pay more attention while encoding a regular language.

To handle it, we use a cloud to define *pseudo point intervals* — a set of intervals on the same level that satisfies a special variable ppi. Once we defined point intervals, we guarantee that nothing wrong happens above the point intervals, i.e. $[\bar{D}](ppi \Rightarrow [\bar{D}] \bigwedge_{v \in \mathcal{V}ar} \neg v)$. The idea is that having pseudo point intervals, we can simply adopt the formula from the proof of Theorem 2 to prove undecidability, replacing D by $\bar{D}$ and λ_i by λ'_i, where $\lambda'_i = \langle D^i \rangle ppi \wedge \neg \langle D^{i+1} \rangle ppi$.

As the *successor* (*predecessor*) function is well defined in discrete linear orders for all points except for the maximal (minimal, resp.) one, it should be clear what we mean by $c + k$ ($c - k$, resp.), where c is a point and k is a natural number — it is just a successor (predecessor, resp.) function iterated k times. Now we are going to write a formula Φ^s_{ppi} that is satisfied only in models with pseudo point intervals. More precisely, $\mathfrak{M}, [a, b] \models \Phi^s_{\text{ppi}}$ iff there exists a superinterval $[c, d]$ of $[a, b]$ such that:

(i) $\mathfrak{M}, [c, d] \models L \wedge ppi \wedge \neg l_e$

(ii) For each even $k \in \mathbb{Z}$ such that $[c + k, d + k]$ is in the model, $\mathfrak{M}, [c + k, d + k] \models ppi \wedge \neg l_e$

(iii) For each odd $k \in \mathbb{Z}$ such that $[c + k, d + k]$ is in the model, $\mathfrak{M}, [c + k, d + k] \models ppi \wedge l_e$

So all the point intervals are at the same level at least in some fragment of the model that contains an interval which labeled with L (so the place where a word from a regular language starts) and is long enough to contain the infinite world.

The technical aspects of Φ^s_{ppi} are not surprising — there are almost the same as for cloud. Note that the last property is a little bit stronger, as explained before. In this case, $[G]\varphi$ stands for $\varphi \wedge [\bar{D}]\varphi$.

- $\langle \bar{D} \rangle (ppi \wedge L)$
- $[G](\langle \bar{D} \rangle ppi \Rightarrow \langle \bar{D} \rangle (ppi \wedge l_e) \wedge \langle \bar{D} \rangle (ppi \wedge \neg l_e))$
- $[G](ppi \Rightarrow [\bar{D}] \bigwedge_{v \in \mathcal{V}ar} \neg v)$

Now, for a given automaton $\mathcal{A}$, we define Φ^s_{orient} ($\Phi^s_{L^d_{\mathcal{A}}}$, Φ^s_{par}, Φ^s_{length}) as Φ^d_{orient} ($\Phi_{L^d_{\mathcal{A}}}$, Φ^d_{par}, Φ^d_{length}, resp.) by replacing every occurrence of D by $\bar{D}$ and every occurrence of λ_i by λ'_i. Let $\Psi^s = \Phi^s_{\text{ppi}} \wedge \Phi^s_{\text{orient}} \wedge \Phi^s_{L^d_{\mathcal{A}}} \wedge \Phi^s_{\text{par}} \wedge \Phi^s_{\text{length}}$. It is easy to check that the formula Ψ^s is satisfiable if and only if $\mathcal{A}$ started from the initial state q_0 and empty counters, runs forever. We conclude this discussion with the following theorem.

Theorem 5.1. The satisfiability problem for the formulae of the logic of superintervals, over all discrete models, is undecidable.

6. Strict D

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

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

Our undecidability result holds also for $D_{\subset}$, there are just some minor technical details to handle. Here we will only describe how we label the point intervals 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 point intervals is easy — the formula λ_0 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 one 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. Figure 6 contains an example of such labeling.

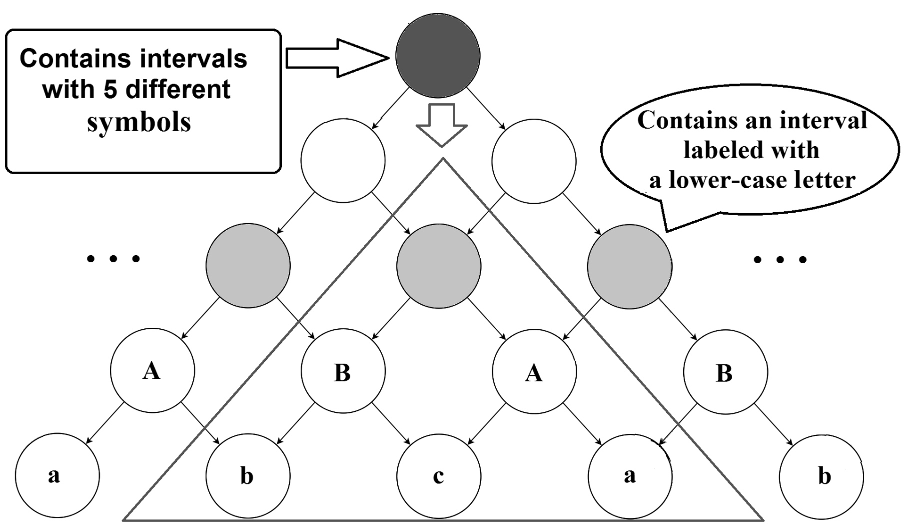

Figure 6. A labeling of point intervals in the strict case.

The formulae expressing properties (i)-(iv) are easy to express using $D_\subset$. Finally, we can define $\lambda_0'' = a \vee b \vee c$.

7. Global satisfiability

The *global satisfiability* problem is defined as follows. For a given formula φ, does there exist a structure $\mathbb{D}$ such that every point of $\mathbb{D}$ satisfies φ? The question about the global satisfiability of formulae has been studied in literature, e.g., in [9] or [13]. For a basic modal logic the global satisfiability problem is EXPTIME-complete, while the classical (local) satisfiability problem is PSPACE-complete.

We show that the global satisfiability problem is very easy for D, a little bit more complicated for $\bar{D}$ (but still in NP), but becomes undecidable as soon as we allow both D and $\bar{D}$.

We start with the following proposition.

Proposition 7.1. The global satisfiability problem for the logic of subintervals is NP-complete in any non-empty class of orders.

Proof:
Let φ be a formula and $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ a model of φ such that for all $w \in \mathrm{I}(\mathbb{D})$ we have $\mathfrak{M}, w \models \varphi$. Let $i \in \mathbb{D}$. We define $\mathfrak{M}' = \langle \mathrm{I}(\{i\}), \gamma' \rangle$ where $\gamma'([i,i]) = \gamma([i,i])$. It is easy to see that $\mathfrak{M}', w \models \varphi$.

Therefore in this case we have a single-interval model property, and checking for the existence such a model can be done in NP. The NP lower bound comes from a trivial reduction from SAT. □

Now we prove that if we have both D and $\bar{D}$, then the global satisfiability problem is much harder.

Proposition 7.2. The global satisfiability problem for the fragment $D\bar{D}$ of the Halpern–Shoham logic is undecidable in any class of orders that contains, for each $n > 0$, at least one chain with length greater than n.

Proof:
For a given two-counter automaton $\mathcal{A}$, let Ψ^d be a formula defined in section 4. Let u be a fresh variable. Define

$$\Psi^g = (u \Rightarrow \Psi^d) \wedge (\neg u \Rightarrow \langle D \rangle \langle \bar{D} \rangle u \vee \langle \bar{D} \rangle \langle D \rangle u)$$

Clearly, any model $\mathfrak{M}$ that globally satisfies Ψ^g contains an interval w labeled by u and $\mathfrak{M}, w \models \Psi^d$. For the other direction, if we have a model $\mathfrak{M}$ and an interval w such that $\mathfrak{M}, w \models \Psi^d$, then we can create a model $\mathfrak{M}'$ from $\mathfrak{M}$ by labeling w by u and all other intervals by $\neg u$. Then $\mathfrak{M}'$ globally satisfies Ψ^g. □

Both proofs are almost straightforward. Now we prove less obvious proposition about $\bar{D}$.

Proposition 7.3. The global satisfiability problem for the logic of superintervals is NP-complete in any non-empty class of discrete orders.

Proof:
Let φ be a formula and $\mathfrak{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ a model of φ such that for all $w \in \mathrm{I}(\mathbb{D})$ we have $\mathfrak{M}, w \models \varphi$. If there is a maximal interval in $\mathrm{I}(\mathbb{D})$, then we simply proceeder like for D and obtain a single-point model.

Suppose that there is no maximal interval in $\mathrm{I}(\mathbb{D})$. We define a *modal type* of an interval w, denoted by mt_w, as a set of subformulae of φ of the form $\langle \bar{D} \rangle \varphi'$ satisfied in w. Note that if w is a superinterval of w', then $mt_w \subseteq mt_{w'}$. Therefore there exists an interval $[a_0, b_0]$ such that for all superintervals v' of v we have $mt_v = mt_{v'}$.

We define a set of intervals $\mathcal{T}$ in the following way. For each ψ, which is a subformula of φ, if there exists a superinterval of $[a_0, b_0]$ that satisfies ψ, then we put one such interval in $\mathcal{T}$. Observe that the size of is polynomial in the size of φ.

Let $\{t_0, t_1, \ldots, t_{k-1}\}$ be a set $\mathcal{T}$ written more explicit and $f : \mathrm{I}(\mathbb{Z}) \rightarrow$ be a function such that for each $w \in \mathrm{I}(\mathbb{Z})$ we have $f(w) = t_{|[a,b]| \mod k}$.

We define a structure $\mathfrak{M}' = \langle \mathrm{I}(\mathbb{Z}), \gamma' \rangle$, where $\gamma'([a,b]) = \gamma(f([a,b]))$. We claim that $\mathfrak{M}'$ is a model of φ.

We prove by the induction of the structure of φ, that for all ψ, which is a subformula of φ, and all w we have $\mathfrak{M}', w \models \psi$ iff $\mathfrak{M}, f(w) \models \psi$.

Let $[a', b']$ be an interval of $\mathfrak{M}'$.

- If ψ is a propositional variable p, then $\mathfrak{M}, f([a', b']) \models \psi$ iff $p \in \gamma(f([a', b'])) = \gamma'([a', b'])$ iff $\mathfrak{M}', [a', b'] \models \psi$.
- The cases when $\psi = \neg\psi'$ or $\psi = \psi' \vee \psi''$ are straightforward.
- If $\psi = \langle D \rangle \psi'$ and $\mathfrak{M}', [a', b'] \models \psi$, then there exists a superinterval $[c', d']$ of $[a', b']$ such that $\mathfrak{M}', [c', d'] \models \psi'$. By the inductive assumption, $\mathfrak{M}, f([c', d']) \models \psi'$. The intervals $f([c', d'])$ is a subinterval of $[a_0, b_0]$, and therefore $mt_{[a_0,b_0]}$ contains ψ. Since $mt_{[a_0,b_0]} = mt_{f([a',b'])}$, we have $\mathfrak{M}, f([a', b']) \models \psi$.

  If $\psi = \langle D \rangle \psi'$ and $\mathfrak{M}, f([a', b']) \models \psi$, then there exists an superinterval $[a, b]$ of $f([a', b'])$ such that $\mathfrak{M}, [a, b] \models \psi'$. ψ' is a subformula o φ, so there is an interval $[c, d] \in \mathcal{T}$ that satisfies ψ'. Let $[c', d']$ be a superinterval of $[a', b']$ such that $f([c', d']) = [c, d]$. By the inductive assumptions, $[c', d']$ satisfies ψ' and therefore $\mathfrak{M}', [a', b'] \models \psi$.

The existence of a model of the form $\langle \mathrm{I}(\mathbb{Z}), \gamma' \rangle$ can be checked in NP. The algorithm simply guesses a modal type of all intervals in the model and a set $\mathcal{T}$, and then simply verify that the modal type is consistent with the types in $\mathcal{T}$.

□

8. 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 prove the following proposition.

Proposition 8.1. 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

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

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 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.

9. Conclusion

Our main contribution can be summarized by the following corollary.

Corollary 9.1. The D fragment of Halpern-Shoham logic, called the logic of subintervals, is undecidable in any class of discrete orders that contains, for each $n > 0$, at least one chain with length greater than n.

For any class that contain only strongly discrete orders, the corollary follows from Theorem 1.1. For the classes that contains at least one order containing infinite interval, the result follows from Theorem 1.2.

As it was shown in Section 6, the result holds for both strict and proper semantics of D. It also does not depended on weather we allow for point intervals or not. The reduction in the latter case is exactly the same as in the former case, only the interpretation changes — the configuration of an automaton is stored in intervals of the form $[a, a+1]$ rather than $[a, a]$.

In Section 7, we considered the global satisfiability problem in discrete case, and we proved that it is NP-complete for the fragments D and $\bar{D}$, and it is undecidable for $D\bar{D}$.

In Section 8, we proved that the $D\bar{D}$ fragment of Halpern–Shoham logic, is undecidable in the class of all orders. An interesting open question is weather the logic of subintervals is decidable in this class.

Acknowledgment

The first author was supported by Polish Ministry of Science and Higher Education research project N N206 371339. The second author was supported by Polish National Science Center based on the decision number DEC-2011/03/N/ST6/00415.

References

[1] Allen, J. F.: Maintaining knowledge about temporal intervals, *Commun. ACM*, **26**(11), November 1983, 832–843, ISSN 0001-0782.

[2] Berger, R.: The undecidability of the domino problem, *Mem. AMS*, **66**, 1966.

[3] Bresolin, D., Della Monica, D., Goranko, V., Montanari, A., Sciavicco, G.: Decidable and Undecidable Fragments of Halpern and Shohams Interval Temporal Logic: Towards a Complete Classification, in: *Logic for Programming, Artificial Intelligence, and Reasoning*, vol. 5330 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2008, 590–604, 10.1007/978-3-540-89439-1_41.

[4] Bresolin, D., Goranko, V., Montanari, A., Sala, P.: Tableau-based decision procedures for the logics of subinterval structures over dense orderings, *Journal of Logic and Computation*, 2008.

[5] Bresolin, D., Goranko, V., Montanari, A., Sciavicco, G.: Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions, *Annals of Pure and Applied Logic*, **161**, 2009, 289–304.

[6] Bresolin, D., Monica, D. D., Goranko, V., Montanari, A., Sciavicco, G.: The Dark Side of Interval Temporal Logic: Sharpening the Undecidability Border, *Proceedings of the 2011 Eighteenth International Symposium on Temporal Representation and Reasoning*, TIME '11, IEEE Computer Society, Washington, DC, USA, 2011, ISBN 978-0-7695-4508-0.

[7] Bresolin, D., Montanari, A., Sala, P., Sciavicco, G.: Optimal Tableaux for Right Propositional Neighborhood Logic over Linear Orders, *Logics in Artificial Intelligence*, 2008.

[8] Goranko, V., Montanari, A., Sciavicco, G.: A Road Map of Interval Temporal Logics and Duration Calculi, *Journal of Applied Non-classical Logics*, **14**, 2004, 9–54.

[9] Goranko, V., Passy, S.: Using the Universal Modality: Gains and Questions, *Journal of Logic and Computation*, **2**(1), 1992, 5–30.

[10] Gurevich, Y. S., Koryakov, I. O.: Remarks on Berger's paper on the domino problem, *Siberian Mathematical Journal*, **13**, 1972, 319–321.

[11] Halpern, J. Y., Shoham, Y.: A propositional modal logic of time intervals, *Journal of The ACM*, **38**, 1991, 935–962.

[12] Hamblin, C. L.: Instants and intervals, *Studium Generale*, **27**, 1971, 127–134.

[13] Hemaspaandra, E.: The Price of Universality, *Notre Dame Journal of Formal Logic*, **37**, 1996, 174–203.

[14] Lodaya, K.: Sharpening the Undecidability of Interval Temporal Logic, *Asian Computing Science Conference*, 2000.

[15] Marcinkowski, J., Michaliszyn, J., Kieronski, E.: B and D Are Enough to Make the Halpern-Shoham Logic Undecidable, *ICALP (2)* (S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, P. G. Spirakis, Eds.), 6199, Springer, 2010.

[16] Monica, D. D., Goranko, V., Montanari, A., Sciavicco, G.: Expressiveness of the interval logics of Allen's relations on the class of all linear orders: complete classification, *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, AAAI Press, 2011, ISBN 978-1-57735-514-4.

[17] Montanari, A., Pratt-Hartmann, I., Sala, P.: Decidability of the Logics of the Reflexive Sub-interval and Super-interval Relations over Finite Linear Orders, *International Syposium on Temporal Representation and Reasoning*, 2010, 27–34, ISSN 1530-1311.

[18] Montanari, A., Puppis, G., Sala, P.: A decidable spatial logic with cone-shaped cardinal directions, *Proceedings of the 23rd CSL international conference and 18th EACSL Annual conference on Computer science logic*, CSL'09/EACSL'09, Springer-Verlag, Berlin, Heidelberg, 2009, ISBN 3-642-04026-8, 978-3-642-04026-9.

[19] Montanari, A., Puppis, G., Sala, P.: Maximal Decidable Fragments of Halpern and Shoham's Modal Logic of Intervals., *ICALP (2)* (S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, P. G. Spirakis, Eds.), 6199, Springer, 2010, ISBN 978-3-642-14161-4.

[20] Moszkowski, B. C.: *Reasoning about digital circuits*, Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1983.

[21] Parikh, R.: A decidability result for a second order process logic, *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, USA, 1978.

[22] Pratt, V. R.: Process logic: preliminary report, *Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, POPL '79, ACM, New York, NY, USA, 1979.

[23] Schwentick, T., Zeume, T.: Two-Variable Logic with Two Order Relations, in: *Computer Science Logic* (A. Dawar, H. Veith, Eds.), vol. 6247 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, 499–513, 10.1007/978-3-642-15205-4_38.