B and D are enough to make the Halpern–Shoham logic undecidable

Jerzy Marcinkowski, Jakub Michaliszyn, Emanuel Kieroński

Institute of Computer Science,
University Of Wroclaw,
ul. Joliot-Curie 15, 50-383 Wroclaw, Poland
{jma,jmi,kiero}@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 contribute to this effort by showing (among other results), that the BD fragment (where only the operators "begins" and "during" are allowed) is undecidable over discrete structures.

1 Introduction

In classical temporal logic structures are defined by assigning properties (propositional variables) to points of time (which 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 [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 of time interval logics. 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., [10]. In computer science, the earliest attempts to formalize time intervals were process logic [15,17] and interval temporal logic [13]. 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 not earlier 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: D can be expressed using B and E (a prefix of my suffix is my infix), L can be expressed by A ("later" means "meets an interval that meets") and 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 five 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.

Some effort has been put since this time to settle this question. First, it was shown [12] that the BE fragment is undecidable. Recently, negative results were also given for the classes $B\bar{E}$, $\bar{B}E$, $\bar{B}E$, $A\bar{A}D$, $\bar{A}D^*\bar{B}$, $\bar{A}D^*B$ [3,7]. Another surprising negative result was that $O\bar{O}$ is undecidable over discrete orders [4,5].

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 [9]. The fragment using only A and $\bar{A}$ is a bit harder and its decidability was only recently shown [7,8]. Obviously, this last result implies decidability of $L\bar{L}$ as L is expressible by A. The last remaining fragment with just one operator is D — in this case, it was only shown that satisfiability is decidable over dense structures ([2,6]). Another interesting decidable fragment $AB\bar{B}$ [14].

In this paper we show that the BD fragment (and so DE) is undecidable. To make the presentation simpler, instead of D we consider the operator $D_{\subset}$ (see Section 2.1), which is more convenient for out purposes. As explained later, our technique can be easily modified to handle the case of D.

In Sections $2.2 - 2.4$ we present the proof of the following result:

Theorem 1. *Satisfiability of the $BD_{\subset}$ fragment of the logic HS is undecidable over the class of all finite orders.*

In Section 2.5 we show how to modify the proof of Theorem 1 to get:

Theorem 2. *Satisfiability of the $BD_{\subset}$ fragment of the HS logic is undecidable over the class of all discrete orderings.*

Finally, in Section 2.6 we formulate, as exercises for the reader, more results that can be proved by slight modifications of our technique. They concern some logics similar to $BD_{\subset}$, most importantly BD, but also $\bar{B}D$, DE and others, and different classes of orderings. In Exercise 5 we show that Theorem 2 remains true even if the class of all discrete orderings is replaced by any nontrivial class of discrete orderings.

What remains open is the status of the D fragment over discrete orders — a slightly more expressible fragment BD is shown undecidable in this paper, but on the other hand the $D\bar{D}$ fragment over dense orderings is known to be decidable ([2,6]).

2 Technical Part

2.1 Preliminaries

Orderings. As in [11], we say that a total[1] order $\langle \mathbb{D}, \leq \rangle$ is *discrete* if each element is either minimal (maximal) or has a unique 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 logic HS. Let $\langle \mathbb{D}, \leq \rangle$ be a discrete ordered set[2].

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 a set of all intervals over $\mathbb{D}$ and $\mathcal{V}ar$ is a finite set of variables. A structure of the form $\mathrm{M} = \langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ is called a *model*.

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. $\mathrm{M}, [a, b] \models \langle B \rangle \varphi$ iff there exists an interval $[a, b']$ such that $b' < b$ and $\mathrm{M}, [a, b'] \models \varphi$.
5. $\mathrm{M}, [a, b] \models \langle E \rangle \varphi$ iff there exists an interval $[a', b]$ such that $a < a'$ and $\mathrm{M}, [a', b] \models \varphi$.

Boolean connectives $\vee, \Rightarrow, \Leftrightarrow$ are introduced in the standard way. We also use the operators $\langle D \rangle \varphi \equiv \langle B \rangle \langle E \rangle \varphi$ and $\langle D_{\subset} \rangle \varphi \equiv \langle D \rangle \varphi \vee \langle E \rangle \varphi$. For $X \in \{B, E, D, D_{\subset}\}$ we abbreviate $\neg \langle X \rangle \neg \varphi$ by $[X]\varphi$. By $\bar{B}, \bar{D}, \bar{D}_{\subset}$, and $\bar{E}$ we denote inversed operators.

A formula φ is said to be *satisfiable* with respect to a class of orderings $\mathcal{D}$ if there exist a structure $\mathbb{D} \in \mathcal{D}$, a labeling γ, and an interval $[a, b]$ 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 with respect to $\{\mathbb{D}\}$.

2.2 Convenient automata

As for the tool in our undecidability proofs, we are going to use undecidability of the problem of establishing whether a given two-counter automaton with a finite set Q of states, starting from a given initial state q_0 and two empty counters, will ever halt. An instruction of such an automaton has the format:

if the current state is q, 1st counter is/isn't 0, 2nd counter is/isn't 0, then change state to q' and increase/decrease 1st/2nd counter

Notice that we assume, for simplicity, that exactly one counter is changed in a single step.

[1] Halpern and Shoham consider also partial orders. Our techniques can be easily modified to handle them. See Exercise 7 in Section 2.6.

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

There is, however, from the point of view of our encoding, a slight inconvenience in this format: being in the configuration after step $k+1$ of the computation, we will not be able to easily check if the counters after step k were empty or not. For this reason we introduce the following notion of *convenient two-counter finite automaton* (or just *convenient automaton*):

A convenient automaton is given by four **disjoint** sets of states Q_{00}, Q_{01}, Q_{10}, and Q_{11}, and by a set of instructions of the form:

if the current state is q, then increase/decrease 1st/2nd counter and change the state to one of $\{q^{00}, q^{01}, q^{10}, q^{11}\}$, where $q^{ij} \in Q_{ij}$ for $i, j \in \{0, 1\}$

We assume here that there is exactly one instruction of this form for each non-final state q. For each *final state* there is exactly one instruction of the form "remain in the current state and do not change the counters".

Formally, the state-transition relation is given by a set $\mathcal{C} \subseteq Q^2$, where $Q = Q_{00} \cup Q_{01} \cup Q_{10} \cup Q_{11}$, and a function $\nu : Q \to \{d_f, d_s, i_f, i_s, \perp\}$, where $d/i_{f/s}$ means "decrease/increase first/second counter", respectively, and $\perp$ means "do nothing". We assume that there is no $q \in Q_{00} \cup Q_{01}$ with $\nu(q) = d_f$ and there is no $q \in Q_{00} \cup Q_{10}$ with $\nu(q) = d_s$.

Definition 1. *A configuration $\langle q, x, y\rangle$ of a convenient automaton is* admissible, *with $q \in Q_{ij}$ being a state and x, y being the numbers on the counters, if $i = 0 \Leftrightarrow x = 0$ and $j = 0 \Leftrightarrow y = 0$. We assume that $q_0 \in Q_{00}$, where q_0 is the initial state of the automaton.*

The halting problem is the problem of establishing whether, for a given convenient automaton with a single final state q_F, there exists a computation starting from $\langle q_0, 0, 0\rangle$, consisting only of admissible configurations, and ending in a configuration with the final state. By an obvious simulation of a standard two-counter automaton we get:

Lemma 1. *The halting problem for convenient automata is undecidable.*

In Section 2.4 we are going to write, for a given convenient automaton with a single final state, a formula of logic $BD_{\subset}$ which will be satisfiable if and only if the automaton halts, in the sense of the above lemma.

In Section 2.5 we will consider automata with two final states q_F and q_G. We say that q_F accepts and q_G rejects. The formula we are going to write for such an automaton will be satisfiable if and only if the automaton does not reject. Clearly such non-rejectance problem is also undecidable.

2.3 Finite orders. Intervals, slices, and configurations.

In this and the next section, we will prove Theorem 1. Suppose a convenient automaton $\mathcal{A}$, with the set of states $Q = Q_{00} \cup Q_{01} \cup Q_{10} \cup Q_{11}$, is given. We consider any finite total order consisting of $N+1$ elements $0, \ldots N$, in this order. Let $\mathrm{I}(\mathbb{D})$ be the set of all the intervals with endpoints in the set $\mathbb{D} = \{0, \ldots N\}$. The intervals will be labeled with a set $\mathcal{V}ar$ of propositional variables consisting of the following elements

- A variable q for every $q \in Q$. Variables of this kind will be called, not surprisingly, *states*.
- A variable $c_{q,q'}$ for each pair q, q' such that there is an instruction in $\mathcal{A}$, allowing changing a state from q to q'. Variables of this sort will be called *step controllers*. The set of all step controllers will be denoted as $\mathcal{C}$.
- Variables $f_0, f_1, f_0^l, f_1^l, f_0^u, f_1^u$, called *f-marks*. Variables f_1, f_1^l, f_1^u will be also called *f-ones* and f_0, f_0^l, f_0^u will be also called *f-nulls*. Variables f_1^u, f_0^u will be called *f-upper critical* and variables f_1^l, f_0^l will be called *f-lower critical*.
- Variables $s_0, s_1, s_0^l, s_1^l, s_0^u, s_1^u$ called *s-marks*. Variables s_1, s_1^l, s_1^u will be also called *s-ones* and s_0, s_0^l, s_0^u will be also called *s-nulls*.

The letter f above stands for "the first counter" and the letter s stands for "the second counter". From now on, we will usually only bother with first counter — the way one deals with second one is completely analogous.

For any i, with $0 \leq i \leq N$, the set of intervals $v_i = \{[i,i], [i,i+1], \ldots [i,N]\}$ will be called a *slice*. Slices are going to serve us as devices to store the configurations of $\mathcal{A}$. Slices v_i and v_{i-1} will be called *consecutive*. We will force consecutive slices to store consecutive configurations of $\mathcal{A}$. Notice that if slices w and w' are consecutive, then w' is longer by one element than w — this is as it should be, since to store future configurations of $\mathcal{A}$ we may possibly need more and more room for counters (notice that the direction of time is slightly counterintuitive here).

Definition 2. *Let $\gamma : \mathrm{I}(\mathbb{D}) \to \mathcal{P}(\mathcal{V}ar)$ be a labeling of intervals with propositional variables. We will say that γ is slicewise correct if for each slice v_i there exist $q, q' \in Q$, with $c_{q',q} \in \mathcal{C}$, such that the following conditions hold:*

(i) $q \in \gamma([i,i])$ *and for all other pairs* q'', j, *where* $q'' \in Q$ *and* $i \leq j \leq N$, *we have* $q'' \notin \gamma([i,j])$.

(ii) $c_{q',q} \in \gamma([i,i+1])$ *and for all other tuples* q_1, q_2, j, *where* $q_1, q_2 \in Q$ *and* $i \leq j \leq N$, *we have* $c_{q_1,q_2} \notin \gamma([i,j])$.

(iii) For each j, where $i \leq j \leq N$, there is exactly one f-mark y such that $y \in \gamma([i,j])$.

(iv) If $y \in \gamma([i,j])$ for some $j > i$ and $y \in \{f_1, f_0^l, f_1^l\}$, then $f_1 \in \gamma([i,k])$ for each $i \leq k < j$. If $y \in \gamma([i,j])$ for some $j > i$ and $y \in \{f_0^u, f_1^u\}$, then for all $i \leq k < j$ if $z \in \gamma([i,k])$ then $z \in \{f_1, f_0^l, f_1^l\}$.

(v) If $f_0 \in \gamma([i,j])$ for some $j > i$, then $y \in \gamma([i,k])$ for some $i \leq k < j$ and upper-critical y. If $y \in \gamma([i,j])$ for some $j > i$ and upper-critical y, then $z \in \gamma([i,k])$ for some $i \leq k < j$ and lower-critical z.

(vi) Let $y \in \gamma([i,i])$ be an f-mark. Then y is an f-null if and only if $q \in Q_{00} \cup Q_{01}$.

(vii) Let $y \in \gamma([i,j])$ for some f-upper critical y. Then $y = f_1^u$ if and only if the instruction of $\mathcal{A}$ concerning the action from the state q' tells that the first counter must be increased (i.e. $\nu(q') = i_f$).

(viii) *Let $y \in \gamma([i,j])$ for some f-lower critical y. Then $y = f_0^l$ if and only if the instruction of $\mathcal{A}$ concerning the action from the state q' tells that the first counter must be decreased (i.e. $\nu(q') = d_f$).*

(ix) *Conditions (iii)-(viii) hold analogously for s-marks and the second counter.*

The way the conditions are written is not the simplest possible one. Actually, they are not meant to be simple. They are meant to be expressible in the logic $BD_{\subset}$ (we will make use of it in Section 2.4). So we feel we owe the reader some explanation.

A labeled slice which satisfies the conditions from the above definition can naturally be understood as an admissible configuration of the convenient automaton $\mathcal{A}$. We imagine v_i as a tape of $N - i + 1$ cells, where the interval $[i,i]$ represents the first cell, $[i,i+1]$ represents the second, and so on. If the labeling is slicewise correct, then in each of those cells we keep one f-mark (and one s-mark). The number of f-ones on the tape is understood to be the value of the first counter. Conditions (iii)-(v) imply that the marks occur in some fixed order: first there are f-ones in the cells $[i,i]$ to $[i,j]$, for some j, and then f-nulls in the cells $[i,j+1]$ to $[j+1,N]$. Most of those f-marks are either equal f_1 or f_0 — only near the border between ones and nulls typically there are two f-critical marks: one lower, and one upper (this follows from the conditions (iv) and (v)). Those critical marks can be both ones, both nulls or the lower can be a one and the upper a null — depending on the action of $\mathcal{A}$ in the state q' which is assumed to be the previous state of $\mathcal{A}$. This is a crucial part of a mechanism that will be used in Lemma 2. In fact, for technical reasons, in some borderline cases (namely, in slices representing initial configurations, and successor configurations of configurations with empty counter) one of critical marks will not appear.

Notice that the condition (vi) together with its counterpart concerning the second counter, imply that the configuration described by the labeling is admissible (in the sense of Definition 1).

In the figure below we present a slicewise correct labeling of $\mathrm{I}(\mathbb{D})$ for $\mathbb{D} = \{0,1,\ldots,8\}$. The intervals are arranged in the triangular table, such that interval $[i,j]$, is located in the i-th row and j-th column. Notice that each row corresponds to a slice. For transparency we do not present the values of s-marks.

0	1	2	3	4	5	6	7	8	
f_0^l, q_8	f_0^u, c_{q_7,q_8}	f_0	f_0	f_0	f_0	f_0	f_0	f_0	0
	f_1, q_7	f_0^l, c_{q_6,q_7}	f_0^u	f_0	f_0	f_0	f_0	f_0	1
		f_1, q_6	f_1, c_{q_5,q_6}	f_0^l	f_0^u	f_0	f_0	f_0	2
			f_1, q_5	f_1, c_{q_4,q_5}	f_1^l	f_0^u	f_0	f_0	3
				f_1, q_4	f_1^l, c_{q_3,q_4}	f_1^u	f_0	f_0	4
					f_1, q_3	f_1^l, c_{q_2,q_3}	f_0^u	f_0	5
						f_1^l, q_2	f_1^u, c_{q_1,q_2}	f_0	6
							f_1^u, q_1	f_0, c_{q_0,q_1}	7
								f_0^u, q_0	8

In fact, this model represents the following computation of an automaton (again, we consider only one counter): $q_0 \xrightarrow{+1} q_1 \xrightarrow{+1} q_2 \xrightarrow{=} q_3 \xrightarrow{+1} q_4 \xrightarrow{=}$

$q_5 \xrightarrow{-1} q_6 \xrightarrow{-1} q_7 \xrightarrow{-1} q_8$. It appears that the labeling is also *stepwise correct* in the following sense.

Definition 3. *Let $\gamma : \mathrm{I}(\mathbb{D}) \to \mathcal{P}(\mathcal{V}ar)$ be a labeling of intervals with propositional variables. We will say that γ is stepwise correct if it is slicewise correct and for each two consecutive slices v_{i+1}, v_i the following conditions hold:*

(i) If q, q' are states, such that $q \in \gamma([i+1, i+1])$ and $q' \in \gamma([i,i])$, then $c_{q,q'} \in \gamma([i, i+1])$.
(ii) For each j, where $i < j \leq N$, the two conditions are equivalent:
- *$y \in \gamma([i+1, j])$ for some y being an f-null;*
- *$f_0 \in \gamma([i, j])$.*

(iii) Same as (ii) but for the second counter.

The following crucial lemma establishes the correspondence between stepwise correct labelings and computations of the automaton $\mathcal{A}$.

Lemma 2. *Let γ be a slicewise correct labeling. The following two conditions are equivalent:*

(1) γ is stepwise correct.
(2) Any pair of consecutive slices v_{i+1}, v_i represent two consecutive (in the sense of the transitions of the convenient automaton $\mathcal{A}$) admissible configurations of $\mathcal{A}$.

Due to the space limit we are not able to present all details of the formal proof. Instead, using our example, we only illustrate the mechanism required for the $\Rightarrow$ direction.

Consider for example the transition from slice v_5, which represents the automaton in the configuration in state q_3 and the counter equal to 2. At this point our automaton should increase the value of its counter. Let us see that v_4 needs to look exactly as in our figure. Conditions (i) and (ii) of Definition 2 and condition (i) of Definition 3 say that the state variable and the step controller of v_4 agree with the transition function, so in our case they have to be q_4 and $c_{q_3 q_4}$, respectively. By condition (ii) of Definition 3, the two rightmost positions in v_4 must be f_0-s. The same condition forbids f_0-s at earlier positions in v_4. By condition (v) of Definition 2, f_0-s have to be preceded by a lower-critical and by an upper-critical marks. Since we impose a specific order on marks, they have to be located exactly at intervals $[4, 6]$ and $[4, 7]$. By (vii) and (viii) those critical marks have to be f_1^l and f_1^u, respectively. Finally, by (iv) the critical positions may be proceeded by f_1-s only.

Once we know how to simulate the steps of the automaton, it is time for:

Definition 4. *Let $\gamma : \mathrm{I}(\mathbb{D}) \to \mathcal{P}(\mathcal{V}ar)$ be a labeling of intervals with propositional variables. We will say that γ is globally correct if it is stepwise correct and there exist $0 \leq i < j \leq N$ such that $q_0 \in \gamma([j, j])$ and $q_F \in \gamma([i, i])$.*

By Lemma 2, a globally correct labeling γ of $\mathbb{D}$ exists for some N if and only if the convenient automaton $\mathcal{A}$ halts after at most N computation steps.

What remains to be done, in order to end the proof of Theorem 1, is writing down a formula of logic $BD_{\subset}$ which is satisfiable if and only if a globally correct labeling γ exists for some N.

2.4 Finite orders. The formula.

The formula ϕ we are going to write will be of the form $\phi^1 \wedge \phi^2 \wedge \phi^3$, where ϕ^1 will be satisfied in models whose labeling is slicewise correct, ϕ^2 will be satisfied in models whose labeling is stepwise correct, provided it is slicewise correct, and ϕ^3 will be satisfied in models whose labeling is globally correct, provided it is stepwise correct. Formulae ϕ^1 and ϕ^3 are straightforward to write. To be able to write ϕ^2 we need one more easy lemma:

Lemma 3. *Let γ be a stepwise correct labeling, and let $0 \leq i < j \leq N$.*
Then the following two conditions are equivalent:

(1) $y \in \gamma([i+1, j])$, for some f-null y.
(2) There exists $i < k \leq j$ and an f-null variable x such that $x \in \gamma([k, j])$

Proof. Obviously (1) implies (2). Implication in the opposite direction follows from condition (ii) of Definition 3. But one can also think, that it reflects the fact, that if the first counter (in v_k) stored a number smaller than $j-k+1$ $k-i-1$ moves ago, then now (in v_{i+1}) it stores a number smaller than $j-i$. □

First notice that we can define, as $\langle B_G \rangle \psi = \psi \vee \langle B \rangle \psi$, an operator saying that ψ holds true in some (interval which is) prefix of the current interval. Let also $atMostOne(X) = \bigwedge_{x \in X}(x \Rightarrow \bigwedge_{x' \in X \setminus \{x\}} \neg x')$ be an operator saying that at most one variable from the set X is true in the current interval. Another useful abbreviation is the operator "globally" $[G]\varphi = \varphi \wedge [D_{\subset}]\varphi \wedge [B]\varphi$ - it says that φ is satisfied in every (reachable) interval.

Now we are able to write ϕ. Define $\phi^1 = \phi^1_{(i)} \wedge \phi^1_{(ii)} \wedge \ldots \wedge \phi^1_{(viii)} \wedge \phi^1_{(ix)}$, where the subformulae mimic the conditions from Definition 2:

$$\phi^1_{(i)} = [G](([B]\bot \Leftrightarrow \bigvee_{q \in Q} q) \wedge atMostOne(Q))$$

$$\phi^1_{(ii)} = [G](\bigvee_{c \in \mathcal{C}} c \Leftrightarrow \langle B \rangle \top \wedge [B] \bigvee_{q \in Q} q) \wedge [G] atMostOne(\mathcal{C})$$

$$\phi^1_{(iii)} = [G](atMostOne(\{f_1, f_0, f_1^l, f_1^u, f_0^l, f_0^u\}) \wedge (f_1 \vee f_0 \vee f_1^l \vee f_1^u \vee f_0^l \vee f_0^u))$$

$$\phi^1_{(iv)} = [G]((f_1 \vee f_1^l \vee f_0^l) \Rightarrow [B]f_1) \wedge [G]((f_0^u \vee f_1^u) \Rightarrow [B](f_1 \vee f_1^l \vee f_0^l))$$

$$\phi^1_{(v)} = [G](\langle B \rangle \top \Rightarrow (f_0 \Rightarrow \langle B \rangle (f_1^u \vee f_0^u)) \wedge ((f_0^u \vee f_1^u) \Rightarrow \langle B \rangle (f_1^l \vee f_0^l)))$$

$$\phi^1_{(vi)} = [G]([B]\bot \wedge (f_0 \vee f_0^l \vee f_0^u) \Leftrightarrow \bigvee_{q \in Q_{00} \cup Q_{01}} q)$$

Let us split $\mathcal{C}$ into $\mathcal{C}_{if}, \mathcal{C}_{is}, \mathcal{C}_{df}, \mathcal{C}_{ds}, \mathcal{C}_{\perp}$ where $\mathcal{C}_{if}$ contains variables related to instructions that increase the first counter, $\mathcal{C}_{df}$ contains variables related to instructions that decrease the first counter, $\mathcal{C}_{is}$ and $\mathcal{C}_{ds}$ contain variables related to instructions increasing and decreasing second counter, respectively, and $\mathcal{C}_{\perp}$ contains variables related to the remaining instructions.

$$\phi^1_{(vii)} = [G]\left(\left(\bigwedge_{c \in \mathcal{C}_{if}} \langle B_G \rangle c \Rightarrow \neg \langle B_G \rangle f^u_0 \right) \wedge \left(\bigwedge_{c \in \mathcal{C} \setminus \mathcal{C}_{if}} \langle B_G \rangle c \Rightarrow \neg \langle B_G \rangle f^u_1 \right) \right)$$

$$\phi^1_{(viii)} = [G]\left(\left(\bigwedge_{c \in \mathcal{C}_{df}} \langle B_G \rangle c \Rightarrow \neg \langle B_G \rangle f^l_1 \right) \wedge \left(\bigwedge_{c \in \mathcal{C} \setminus \mathcal{C}_{df}} \langle B_G \rangle c \Rightarrow \neg \langle B_G \rangle f^l_0 \right) \right)$$

And $\phi^1_{(ix)}$ is analogous to a conjunction of $\phi^1_{(iii)}$ to $\phi^1_{(viii)}$, but concerns the second counter.

Define $\phi^2 = \phi^2_{(i)} \wedge \phi^2_{(ii)}$ where $\phi^2_{(i)}$ reflects condition (i) from Definition 3 and $\phi^2_{(ii)}$ reflects conditions (ii) and (iii):

$\phi^2_{(i)} = [G]\left(\bigwedge_{c_{q_1,q_2} \in \mathcal{C}} c_{q_1,q_2} \Rightarrow \langle B \rangle q_2 \wedge \langle D_{\subset} \rangle q_1 \right)$

$\phi^2_{(ii)} = [G]\left(f_0 \Leftrightarrow \langle D_{\subset} \rangle (f^l_0 \vee f^u_0 \vee f_0) \right) \wedge [G]\left(s_0 \Leftrightarrow \langle D_{\subset} \rangle (s^l_0 \vee s^u_0 \vee s_0) \right)$

Finally, let $\phi^3 = \langle B \rangle q_F \wedge \langle D_{\subset} \rangle q_0$.

Now, it follows from the construction, that the formula ϕ is satisfiable *over some finite ordering* if and only if the convenient automaton $\mathcal{A}$ halts. This ends the proof of Theorem 1.

2.5 Infinite discrete orders

The formula ϕ we wrote in Section 2.4 works fine for orders in which each interval (with endpoints among the elements of the ordered set) contains only finitely many points — for example for finite sets. But if arbitrary discrete orders are allowed, then it may very well happen that ϕ will be satisfiable even if $\mathcal{A}$ does not halt. Actually, as the following example shows, it will be satisfiable if there exists a configuration $\mathbf{c}$ of $\mathcal{A}$, which is final, in the sense that the state is q_F, and such that there exists an "infinite computation" which ends in $\mathbf{c}$ but never begins (we mean here an infinite sequence $\mathbf{c}= c_0, c_1, c_2 \ldots$ of configurations, such that for each i the configuration c_i is a result of one computation step performed in c_{i+1}).

Example. Fix an "infinite computation" as above and imagine an order $\langle V, \leq \rangle$ consisting of elements $a_0, a_1, a_2, \ldots$ and $b_0, b_1, b_2, \ldots$ with $a_i < a_j$ and $b_j < b_i$ for $i < j$ and with $a_i < b_j$ for any i, j. Let $d \in V$. Like in Section 2.3, we can view the set of sequences $\{[d, x] : x \leq b_0\}$ as a slice, and encode any configuration of A as a labeling of this slice. Let us label the slice $\{[a_i, x] : x \leq b_0\}$ as the configuration c_i of the above infinite sequence, and the slice $\{[b_i, x] : x \leq b_0\}$ as the configuration which is reached by $\mathcal{A}$ after i computation steps, if started in the beginning configuration. The described labeling turns out to satisfy ϕ.

Obviously, some details are left here for the reader to fill — for example how to label the intervals of the form $[a_j, b_j]$. □

Still, the proof of Theorem 1 from Sections 2.2 - 2.4 can be easily modified so that it proves Theorem 2.

It is enough to consider convenient automaton with two final states: q_F (the accepting state) and q_G (the rejecting state). Let us remind that we assume that there is one instruction of automaton for each state — also for the two final states — but in the final states the instruction tells the automaton to freeze, that is to leave the counters unchanged and remain in the same state.

The undecidable problem we will use now, is the problem if this new convenient automaton $\mathcal{A}$ ever rejects. More precisely, we write a formula ϕ' such that ϕ' is satisfiable in some discrete order if and only if A (started from the beginning configuration and visiting only admissible configurations) does not reject (i.e. it accepts or runs forever). As it turns out, the only change we need to make in ϕ concerns the subformula ϕ^3: let $\phi' = \phi^1 \wedge \phi^2 \wedge \psi^3$, where $\psi^3 = \langle B \rangle q_F \wedge \langle D_\subset \rangle q_0 \wedge \neg \langle D_\subset \rangle q_G$.

Now, suppose $\mathcal{A}$ does not reject. There are two possible cases: either it accepts (after some finite number of steps reaches a configuration with q_F) or it does not halt. In the former case we can build a finite model, like in Section 2.3. In the latter, we proceed like in the example above — notice that, since the automaton can freeze in the state q_F, we can be sure that an "infinite computation" $\mathbf{c}$ exists.

What remains to be proved is that if $\mathcal{A}$ rejects then ϕ' does not have a model. Suppose it rejects after N steps, and that there is a model $\langle \mathrm{I}(\mathbb{D}), \gamma \rangle$ of ϕ' where $a < b$ are such elements of $\mathbb{D}$ that $q_F \in \gamma([a, a])$ and $q_0 \in \gamma([b, b])$. Let q^i be the state of $\mathcal{A}$ after i steps of the rejecting computation (so that $q^0 = q_0$ and $q^N = q_G$.)

Let $b_0 = b$ and let b_{i+1}, for $0 \leq i < N$, be the predecessor of b_i in the order, if such a predecessor exists. Actually, this is exactly what we need q_F for — to make sure that there are many enough elements of $\mathbb{D}$ (smaller than b), to accommodate the rejecting computation:

Lemma 4. *Let* $0 \leq i \leq N$

(1) If b_i *exists then* $q^i \in \gamma([b_i, b_i])$.
(2) If b_i *exists and* $0 \leq j \leq i$ *then* $q_F \notin \gamma([b_j, b_j])$.
(3) If b_i *exists then* $a < b_i$.
(4) If b_i *exists then* b_{i+1} *exists.*
(5) b_N *exists, and* $q_G \in \gamma([b_N, b_N])$.

The proof of the lemma is by easy induction. Claim (1) of the induction step follows from the construction of ϕ' — remember that the interval $[b_i, b]$ is finite and all the arguments from Section 2 apply. Claim (2) follows from (1), and from the fact that a rejecting computation of $\mathcal{A}$ never enters the state q_F. Claim (3) follows from (2) and from the inequality $a < b$. Claim (4) follows from (3) and from the assumption that each element of $\mathbb{D}$ is either the smallest or has a predecessor. Finally, since $q^N = q_G$ claim (5) follows from (4).

But it follows from the lemma, that there exists $d \in \mathbb{D}$ such that $q_G \in \gamma([d,d])$, which is not allowed by ϕ'. This contradiction ends the proof of Theorem 2.

2.6 More results (in exercises).

Exercise 1. Show that the satisfiability problem for the logic $BD_{\subseteq}$ is undecidable, where $D_{\subseteq}$ is the *reflexive* variant of D, i.e., $\mathrm{M}, [a,b] \models \langle D_{\subseteq}\rangle\varphi$ iff there exist a', b' such that $a \leq a' \leq b' \leq b''$, and $\mathrm{M}, [a',b'] \models \varphi$. *Hint*: In our proof we use the convenient property that intervals visible by B are not visible by $D_{\subset}$. This is not the case for $BD_{\subseteq}$. Use an additional variable p and formulae $[G](\langle B\rangle p \to p) \wedge [G](\langle B\rangle \neg p \to \neg p)$ and $[G]((\bigvee_{c\in C} C) \Rightarrow \langle D\rangle p \wedge \langle D\rangle \neg p)$ to distinguish between the current slice and the previous one.
Exercise 2. Show that the satisfiability problem for the logic BD is undecidable. Recall that $\langle D\rangle\varphi \equiv \langle B\rangle\langle E\rangle\varphi$. Such variant of D is called *strict*. *Hint*: The idea is to use three "critical levels" instead of two.
Exercise 3. Show that the satisfiability problem for the logic $\bar{B}D$ over discrete orderings is undecidable. *Hint*: Modify the formula for BD. The formula ϕ^1, can be easily expressed — use $\bar{B}$ to define the order on the marks and D to label intervals with the states and the step controllers (note that the property $\phi^1_{(v)}$ needs to be slightly modified). Finally, formula ϕ^3 can be replaced by $\langle D\rangle(q_F \wedge \langle \bar{B}\rangle\langle D\rangle q_0)$, and $\phi^2_{(i)}$ can be adjusted in the same way.
Exercise 4. Show that the satisfiability problem for logic DE and $D\bar{E}$ over discrete orderings is undecidable. *Hint*: Actually, no hint is needed here. Just replace every occurrence of B by E in the formula.
Exercise 5. Show that for any class of discrete orderings $\mathcal{D}$, BD is undecidable over $\mathcal{D}$ iff $\mathcal{D}$ contains orderings with arbitrarily large chains. *Hint*: For undecidability result, consider two cases: if there exist $\mathbb{D} \in \mathcal{D}$ and $a, b \in \mathbb{D}$ such that $\{c | a \leq c \leq b\}$ if infinite, then you can use the proof of Theorem 2, otherwise you can use the proof of Theorem 1. For decidability result observe that the number of non-isomorphic chains with bounded size is bounded.
Exercise 6. Show that $BD\bar{D}$ logic is undecidable over the class of all ordering. *Hint*: Consider the formula $[G](length_0 \Rightarrow \langle \bar{D}\rangle(p \wedge length_1) \wedge \langle \bar{D}\rangle(\neg p \wedge length_1))$, where $length_i$ is true in intervals with the length i.
Exercise 7. In this paper we focused on total orderings. Originally, HS logic was defined for any order that satisfy the following condition. For each a_1, a_2, a_3, a_4 if $a_1 \leq a_2$, $a_1 \leq a_3$, $a_2 \leq a_4$, and $a_3 \leq a_4$, then $a_2 \leq a_3$ or $a_3 \leq a_2$. Show that our theorems still hold in this case. *Hint*: Again, no hint is needed here — just read carefully the definition above.

References

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

2. 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.
3. 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.
4. D. Bresolin, D. DellaMonica, V. Goranko, A.Montanari, G. Sciavicco, Undecidability of Interval Temporal Logics with the Overlap Modality, in: Proc. of 16th International Symposium on Temporal Representation and Reasoning - TIME 2009, IEEE Computer Society Press, 2009, pp. 88-95.
5. D. Bresolin, D. DellaMonica, 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. C. L. Hamblin. Instants and intervals. Studium Generale, 27:127-134, 1971.
11. J. Halpern, Y. Shoham, A propositional modal logic of time intervals, Journal of the ACM 38 (4) (1991) 935-962.
12. 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.
13. B. C. Moszkowski. Reasoning about Digital Circuits. PhD thesis, Stanford University, Computer Science Department, July 1983
14. A. Montanari, G. Puppis, P. Sala, G. Sciavicco. Decidability of the Interval Temporal Logic $AB\bar{B}$ on Natural Numbers. In Proc. of the 27th Symposium on Theoretical Aspects of Computer Science (STACS 2010), pp. 597-608.
15. R. Parikh. A decidability result for second order process logic. In Proc. 19th FOCS, pages 177-183. IEEE, October 1978.
16. A. Pnueli. A temporal logic of programs. In Proc. 18th FOCS, pages 46-57. IEEE, October 1977.
17. V. R. Pratt. Process logic. In Proc. 6th POPL, pages 93-100. ACM, January 1979.
18. A. N. Prior. Past, Present and Future. Clarendon Press, Oxford, 1967.