Kalpa Publications in Computing

Volume 9, 2018, Pages 91–106

LPAR-22 Workshop and Short Paper Proceedings

On Disallowing Punctual Intervals in Reflexive Semantics of Halpern-Shoham Logic

Przemysław Andrzej Wałęga[1,2]

[1] University of Oxford, United Kingdom
[2] University of Warsaw, Poland

Abstract

Halpern-Shoham logic (HS) is a very expressive and elegant formalism for interval temporal reasoning in which the satisfiability problem is undecidable. One of the methods to obtain HS-fragments of lower computational complexity is to adopt the softened (reflexive) semantics of the accessibility relations. In the paper we consider disallowing punctual intervals in reflexive semantics. We show that in this case we gain additional expressive power, which over discrete orders of time points results in PSPACE-hardness of the Horn fragment of HS without diamond modal operators is and in undecidability of the core fragment of HS.

1 Introduction

The logic of Halpern and Shoham (HS in short) is one of the most well-known interval modal logics for temporal knowledge representation and reasoning [9, 8]. Its modal operators correspond to the set of binary relations between intervals known as the *Allen's relations*, namely *begins* ($\mathsf{rel_B}$), *during* ($\mathsf{rel_D}$), *ends* ($\mathsf{rel_E}$), *overlaps* ($\mathsf{rel_O}$), *adjacent to* ($\mathsf{rel_A}$), *later than* ($\mathsf{rel_L}$), and their converses: $\mathsf{rel}_{\overline{\mathsf{B}}}$, $\mathsf{rel}_{\overline{\mathsf{D}}}$, $\mathsf{rel}_{\overline{\mathsf{E}}}$, $\mathsf{rel}_{\overline{\mathsf{O}}}$, $\mathsf{rel}_{\overline{\mathsf{A}}}$, $\mathsf{rel}_{\overline{\mathsf{L}}}$ [1]. A model of HS may be seen as consisting of two layers. The first layer is a time-line, i.e., a set of time points ordered by an 'earlier-later' relation, and the second layer is a set of intervals over this time-line and a set of relations between intervals. In the seminal paper of Halpern and Shoham, the authors assumed that the order of time points is almost arbitrary (only imposed the so-called linear interval property is assumed), an interval is any pair of time points $\langle x, y\rangle$ such that y is not before x (hence, the punctual intervals starting and ending in the same time point are allowed), and that the relations between intervals correspond to the Allen's relations [9].

Since the satisfiability problem of HS-formulas is undecidable a number of syntactical and semantical modifications of the logic have been studied [8, 3, 4, 5, 6, 11]. One of the ideas is to weaken semantics of Allen's relations, by the so-called *softening* [10]. The obtained relations are known as *reflexive semantics*, as most of them become reflexive. Other modifications related to this paper consist of imposing additional conditions on the order of time points (e.g., discreteness or density), disallowing punctual intervals, and syntactically restricting the set of well-formed formulas [2, 5].

The most relevant results for this paper are depicted in Table 2, where $<$ denotes irreflexive semantics, $\leq$ reflexive semantics, $\mathsf{Non\text{-}S}$ (non-strict semantics) allowing punctual intervals,

S (strict semantics) disallowing punctual intervals, Dis discrete timelines, and Den dense timelines. The satisfiability problem of HS-formulas in the Horn form (this fragment is denoted by HS_{horn}) is undecidable under any combination of the above mentioned distinctions, whereas the satisfiability of HS-formulas in the core form (HS_{core}) is undecidable under irreflexive semantics and its complexity under reflexive semantics is an open problem. Disallowing diamond modal operators (i.e., allowing only boxes) in the languages of HS_{horn} and HS_{core} results in $\mathsf{HS}^{\Box}_{horn}$ and $\mathsf{HS}^{\Box}_{core}$, respectively. The satisfiability problem in both of them is tractable under $(<, \mathsf{Non\text{-}S}, \mathsf{Den})$, $(<, \mathsf{S}, \mathsf{Den})$, $(\leq, \mathsf{Non\text{-}S}, \mathsf{Dis})$, $(\leq, \mathsf{Non\text{-}S}, \mathsf{Den})$, and $(\leq, \mathsf{S}, \mathsf{Den})$.

Table 1: Computational complexity of HS fragments, where 'undec', 'h', and 'co' stand for undecidable, hard, and complete, respectively. Our results are written in bold.

	Irreflexive ($<$)				Reflexive ($\leq$)			
	Non-Strict (Non-S)		Strict (S)		Non-Strict (Non-S)		Strict (S)	
	Dis	Den	Dis	Den	Dis	Den	Dis	Den
HS_{horn}	undec	undec	undec	undec	undec	undec	undec	undec
$\mathsf{HS}^{\Box}_{horn}$	undec	P-co	undec	P-co	P-co	P-co	**PSpace-h**	P-co
HS_{core}	undec	undec	undec	undec	PSpace-h	PSpace-h	**undec**	PSpace-h
$\mathsf{HS}^{\Box}_{core}$	PSpace-h	in P	PSpace-h	in P	in P	in P	NL-h	in P

The main results of this paper are that under $(\leq, \mathsf{S}, \mathsf{Dis})$ the satisfiability problem in:

- $\mathsf{HS}^{\Box}_{horn}$ is PSpace-hard;
- HS_{core} is undecidable.

Our results show that the combination of a lack of punctual intervals and discreteness of a time-line gives additional expressive power which was not available in other cases under reflexive semantics. In particular, we will show that under $(\leq, \mathsf{S}, \mathsf{Dis})$ the satisfiability problem of $\mathsf{HS}^{\Box}_{horn}$-formulas reduces to the PSpace-complete problem of checking whether a Turing machine which uses a polynomial memory on the empty input diverges on the empty input and the satisfiability problem for HS_{core} formulas reduces to the undecidable problem of checking whether a given Turing machine diverges on the empty input.

To obtain the new results we show a trick that enables us to mimic under $(\leq, \mathsf{S}, \mathsf{Dis})$ proof techniques used in the irreflexive semantics [5]. We observe that under reflexive semantics the adjacent relation between intervals is not irreflexive if punctual intervals are allowed. Indeed, in this case each punctual interval is adjacent to itself. However, if punctual intervals are disallowed, then the adjacent relation becomes irreflexive. This observation allows us to introduce a formula forcing an alternating placement of two propositional variables over a sequence of subsequent unit intervals (i.e., intervals of length 1). Then, we show how to use these alternating propositional variables to pass information from one interval to another, and consequently how to simulate computation of a Turing machine.

The remaining part of the paper is organized as follows. In Section 2 we introduce syntax and semantics of HS and its fragments. Afterwards, we show in Section 3 that under $(\leq, \mathsf{S}, \mathsf{Dis})$ $\mathsf{HS}^{\Box}_{horn}$ is PSpace-hard and HS_{core} is undecidable. We conclude the paper in Section 4.

2 Halpern-Shoham logic

The language of Halpern-Shoham logic consists of a set of propositional variables PROP, classical propositional connectives $\neg, \wedge$, and 12 modal operators of the form $\langle \mathsf{R} \rangle$, where $\mathsf{R} \in \{\mathsf{B}, \overline{\mathsf{B}}, \mathsf{D}, \overline{\mathsf{D}}, \mathsf{E}, \overline{\mathsf{E}}, \mathsf{O}, \overline{\mathsf{O}}, \mathsf{A}, \overline{\mathsf{A}}, \mathsf{L}, \overline{\mathsf{L}}\}$ (in what follows, we denote this set by $\mathsf{HS}_{\mathsf{rel}}$). Well-formed HS-formulas are defined by the following abstract grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{R} \rangle \varphi,$$

where $p \in \mathsf{PROP}$ and $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. $\top$, $\bot$, $\vee$, and $\rightarrow$ are defined as usual, and $[\mathsf{R}]$ is a dual modal operator to $\langle \mathsf{R} \rangle$ for any $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. An HS-frame is a tuple $\mathcal{F} = (\mathbb{D}, I(\mathbb{D}), \{\mathsf{rel}_{\mathsf{R}}\}_{\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}})$ such that:

- $\mathbb{D} = (D, \leq)$ is an unbounded linear order;
- $I(\mathbb{D}) \subseteq D \times D$ is a set of intervals over $\mathbb{D}$;
- $\mathsf{rel}_{\mathsf{R}} \subseteq I(\mathbb{D}) \times I(\mathbb{D})$ is a binary relation between distinct intervals over $\mathbb{D}$, for any $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$.

An HS-model is a tuple of the form $\mathcal{M} = (\mathbb{D}, I(\mathbb{D}), \{\mathsf{rel}_{\mathsf{R}}\}_{\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}}, V)$, where $(\mathbb{D}, I(\mathbb{D}), \{\mathsf{rel}_{\mathsf{R}}\}_{\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}})$ is an HS-frame and $V : \mathsf{PROP} \longrightarrow \mathcal{P}(I(\mathbb{D}))$. The satisfaction relation for an HS-model $\mathcal{M} = (\mathbb{D}, I(\mathbb{D}), \{\mathsf{rel}_{\mathsf{R}}\}_{\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}}, V)$ and an interval $\langle x, y \rangle \in I(\mathbb{D})$ is defined inductively as follows:

$$\begin{array}{lll}
\mathcal{M}, \langle x, y \rangle \models p & \text{iff} & \langle x, y \rangle \in V(p), \text{ for any } p \in \mathsf{PROP}; \\
\mathcal{M}, \langle x, y \rangle \models \neg\varphi & \text{iff} & \mathcal{M}, \langle x, y \rangle \not\models \varphi; \\
\mathcal{M}, \langle x, y \rangle \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, \langle x, y \rangle \models \varphi \text{ and } \mathcal{M}, \langle x, y \rangle \models \psi; \\
\mathcal{M}, \langle x, y \rangle \models \langle \mathsf{R} \rangle \varphi & \text{iff} & \text{there is } \langle x', y' \rangle \text{ such that } \langle x, y \rangle \mathsf{rel}_{\mathsf{R}} \langle x', y' \rangle \text{ and } \mathcal{M}, \langle x', y' \rangle \models \varphi;
\end{array}$$

for any $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$. A convenient representation of an HS-frame is obtained by treating an interval $\langle x, y \rangle$ as a point in a two-dimensional Cartesian space $D \times D$ such that the abscissa of this point has value x and its ordinate has value y [13]. In compass representation non-punctual intervals correspond to points lying in the north-western half-plane of $D \times D$ (the points whose abscissa is strictly smaller than ordinate). Points lying on the diagonal correspond to punctual intervals. Let us fix any interval $\langle x, y \rangle$. Then, intervals accessible from $\langle x, y \rangle$ with HS modal operators may be determined on the basis of the relative position of the corresponding points in the two-dimensional Cartesian space as presented in Figure 1.

It is easy to see that any HS-formula can be transformed into an equisatisfiable formula which is a conjunction of implications (clauses), and vice versa (in the spirit of separation normal form introduced by [7]), i.e., into a formula generated by the following grammar:

$$\varphi := \lambda \mid \neg\lambda \mid [\mathsf{U}](\lambda \wedge \ldots \wedge \lambda \rightarrow \lambda \vee \ldots \vee \lambda) \mid \varphi \wedge \varphi, \tag{1}$$

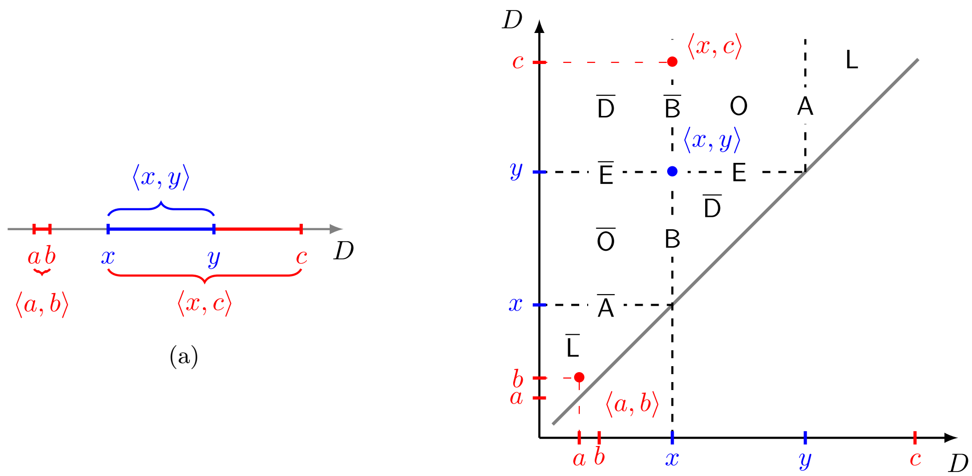
where $[\mathsf{U}]$ is the universal modality, i.e., $[\mathsf{U}]\psi$ is satisfied iff ψ is satisfied in every $\langle x, y \rangle \in I(\mathbb{D})$ whereas λ, the so-called *positive temporal literal*, is a formula defined by the grammar:

$$\lambda := \top \mid \bot \mid p \mid \langle \mathsf{R} \rangle \lambda \mid [\mathsf{R}]\lambda, \tag{2}$$

where $p \in \mathsf{PROP}$ and $\mathsf{R} \in \mathsf{HS}_{\mathsf{rel}}$.

- HS_{horn} is obtained by restricting (1) to the grammar:

$$\varphi := \lambda \mid [\mathsf{U}](\lambda \wedge \ldots \wedge \lambda \rightarrow \lambda) \mid \varphi \wedge \varphi.$$



(a) (b)

Figure 1: Standard (a) and two-dimensional (b) representations of an HS-model, in which $\langle a, b\rangle$ is earlier than $\langle x, y\rangle$, and $\langle x, c\rangle$ is begun by $\langle x, y\rangle$.

- $\mathsf{HS}^{\Box}_{horn}$ is obtained by additional limitation imposed on HS_{horn}, namely the grammar of positive temporal literals (2) is restricted to:

$$\lambda := \top \mid \bot \mid p \mid [\mathsf{R}]\lambda.$$

- HS_{core} is obtained by restricting (1) to the grammar:

$$\varphi := \lambda \mid [\mathsf{U}](\lambda \to \lambda) \mid [\mathsf{U}](\lambda \lor \lambda) \mid [\mathsf{U}](\lambda \land \lambda \to \bot) \mid \varphi \land \varphi.$$

In what follows we define restrictions imposed on HS semantics. First, the distinction between irreflexive and reflexive semantics is obtained by adopting the definitions of rel_R for $\mathsf{R} \in \mathsf{HS}_\mathsf{rel}$ as presented in Table 2.
Second, in the non-strict semantics the set $I(\mathbb{D})$ is defined as:

$$\{\langle x, y\rangle \mid x, y \in D \text{ and } x \leq y\},$$

whereas in strict semantics $I(\mathbb{D})$ is defined as:

$$\{\langle x, y\rangle \mid x, y \in D \text{ and } x < y\}.$$

Hence, in non-strict semantics punctual intervals are allowed, whereas in strict semantics they are forbidden.

3 Computational complexity

In what follows we show that under ($\leq$,S,Dis) the satisfiability problem for $\mathsf{HS}^{\Box}_{horn}$-formulas is PSPACE-hard and for HS_{core}-formulas the problem is undecidable. The proofs are based on encodings of a Turing machine computation.

Table 2: Definitions of irreflexive and reflexive semantics of intervals relations.

Irreflexive semantics:			Reflexive semantics:		
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{L}}} \langle x',y'\rangle$	iff	$y' < x$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{L}}} \langle x',y'\rangle$	iff	$y' \leq x$
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{A}}} \langle x',y'\rangle$	iff	$x' < y',\ y' = x$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{A}}} \langle x',y'\rangle$	iff	$x' \leq y',\ y' = x$
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{O}}} \langle x',y'\rangle$	iff	$x' < x < y' < y$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{O}}} \langle x',y'\rangle$	iff	$x' \leq x \leq y' \leq y$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{B}} \langle x',y'\rangle$	iff	$x = x',\ y' < y$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{B}} \langle x',y'\rangle$	iff	$x = x',\ y' \leq y$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{D}} \langle x',y'\rangle$	iff	$x < x',\ y' < y$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{D}} \langle x',y'\rangle$	iff	$x \leq x',\ y' \leq y$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{E}} \langle x',y'\rangle$	iff	$x < x',\ y = y'$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{E}} \langle x',y'\rangle$	iff	$x \leq x',\ y = y'$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{O}} \langle x',y'\rangle$	iff	$x < x' < y < y'$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{O}} \langle x',y'\rangle$	iff	$x \leq x' \leq y \leq y'$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{A}} \langle x',y'\rangle$	iff	$y = x',\ x' < y'$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{A}} \langle x',y'\rangle$	iff	$y = x',\ x' \leq y'$
$\langle x,y\rangle \mathsf{rel}_{\mathsf{L}} \langle x',y'\rangle$	iff	$y < x'$	$\langle x,y\rangle \mathsf{rel}_{\mathsf{L}} \langle x',y'\rangle$	iff	$y \leq x'$
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{E}}} \langle x',y'\rangle$	iff	$x' < x,\ y = y'$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{E}}} \langle x',y'\rangle$	iff	$x' \leq x,\ y = y'$
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{D}}} \langle x',y'\rangle$	iff	$x' < x,\ y < y'$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{D}}} \langle x',y'\rangle$	iff	$x' \leq x,\ y \leq y'$
$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{B}}} \langle x',y'\rangle$	iff	$x = x',\ y < y'$	$\langle x,y\rangle \mathsf{rel}_{\overline{\mathsf{B}}} \langle x',y'\rangle$	iff	$x = x',\ y \leq y'$

Let us fix the notation, in which a deterministic Turing machine is $M = (\Gamma, Q, q_{start}, q_{halt}, \delta)$, where Γ is M's alphabet (containing the blank symbol $\sqcup$ and the start symbol $\rhd$), Q is the set of M's states, $q_{\mathsf{start}} \in Q$ and $q_{\mathsf{halt}} \in Q$ are start and halting states, respectively, and $\delta : \{Q - q_{\mathsf{halt}}\} \times \Gamma \longrightarrow Q \times \{\Gamma \cup \{\mathsf{L}, \mathsf{R}\}\}$ is a transition function which given a state of the machine and a symbol read by its head determines a new state and a symbol to be written in the current position of the head or movement of the head one cell to the left (L) or one cell to the right (R). We will denote M's configuration in the n'th step of computation by an infinite sequence $C_n(1), C_n(2), C_n(3), \ldots$ where $C_n(m) \in \Gamma^+$ and $\Gamma^+ = \Gamma \cup \{Q \times \Gamma\}$ is the content of the m'th cell in the n'th step such that (q, x) denotes that the cell contains x, the head of M is above this cell, and M is in the state q. Hence, the initial configuration is $(q_{\mathsf{start}}, \rhd), \sqcup, \sqcup, \sqcup, \ldots$. If M starts with empty input and with head above the first cell, then in the first n steps the head may visit only the first n cells. It follows that in the n-th step of computation all cells to the right of the n-th cell contain the blank symbol $\sqcup$.

3.1 Horn fragment without diamonds

It is well known that given a Turing machine M and a polynomial f such that the computation of M with empty input uses at most $f(|M|)$ amount of memory, where $|M|$ is the size of M, checking whether M diverges on empty input is a PSpace-complete problem [12]. We will denote this problem by PSpace-Bound BlankNon-Halting and in what follows we will show that it reduces to the satisfiability problem of $\mathsf{HS}^{\Box}_{horn}$-formulas under ($\leq$,S,Dis).

The main part of the proof consists of showing that $\mathsf{HS}^{\Box}_{horn}$ under ($\leq$,S,Dis) enables us to force an alternating placement of propositional variables unit_1 and unit_2 in the consecutive unit intervals (i.e., intervals of length 1) as depicted in Figure 2. Then, we will show that such a placement of unit_1 and unit_2 allows us to use a technique know from the literature [5, Theorem 4.1 and Theorem 4.2] to encode PSpace-Bound BlankNon-Halting.

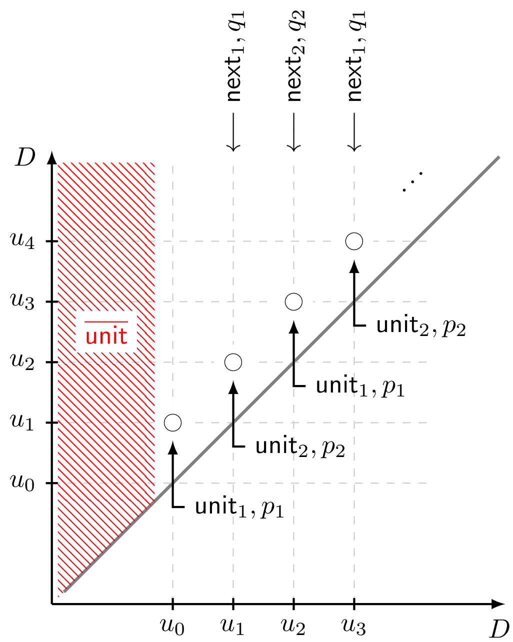


Figure 2: Placement of propositional variables, which is forced by $\varphi_{unit1,2}$ satisfied in $\langle u_0, u_1\rangle$, where '$\longleftarrow \mathsf{next}_k, q_k$' for $k \in \{1,2\}$ denotes a vertical line such that next_k and q_k are satisfied in all points belonging to this line.

Fix a polynomial f such that the Turing machine M starting its computation with empty input uses at most $f(|M|) = N$ tape cells. Notice that in step i of M's computation it suffices to consider $C_i(j)$ such that $j < N$ since the rest of the cells in i'th step contain the blank symbol. The formula forcing intended placement of unit_1 and unit_2 is defined as follows, where $k \in \{1,2\}$:

$$\begin{aligned}
\varphi_{unit1,2} :=& \mathsf{unit}_1 \wedge && (3)\\
& [\mathsf{U}](\mathsf{unit}_k \to [\mathsf{E}]p_k) \wedge [\mathsf{U}](\mathsf{unit}_k \to [\mathsf{B}][\mathsf{A}]q_k) \wedge && (4)\\
& [\mathsf{U}](p_k \wedge q_k \to \bot) \wedge && (5)\\
& [\mathsf{U}](\mathsf{unit}_1 \to [\mathsf{A}]\mathsf{next}_1) \wedge [\mathsf{U}]([\mathsf{E}]\mathsf{next}_1 \to \mathsf{unit}_2) \wedge && (6)\\
& [\mathsf{U}](\mathsf{unit}_2 \to [\mathsf{A}]\mathsf{next}_2) \wedge [\mathsf{U}]([\mathsf{E}]\mathsf{next}_2 \to \mathsf{unit}_1) \wedge && (7)\\
& [\mathsf{U}](\mathsf{unit}_1 \wedge \mathsf{unit}_2 \to \bot) \wedge && (8)\\
& [\overline{\mathsf{A}}][\overline{\mathsf{B}}][\mathsf{B}]\overline{\mathsf{unit}} \wedge [\mathsf{U}](\overline{\mathsf{unit}} \wedge \mathsf{unit}_k \to \bot). && (9)
\end{aligned}$$

Lemma 1. *Let $\mathcal{M}$ be any* HS*-model under ($\leq$,S,Dis) semantics. Assume that $\mathcal{M}, \langle u_0, u_1\rangle \models \varphi_{unit1,2}$ for some interval $\langle u_0, u_1\rangle$ and let $u_1 < u_2 < \ldots$ be the infinite sequence of immediate $<$-successors. Then, u_1 is the immediate $<$-successor of u_0 and for any interval $\langle x, y\rangle$ the following conditions are satisfied:*

1) $\mathcal{M}, \langle x, y\rangle \models \mathsf{unit}_1$ *iff* $x = u_i$ *and* $y = u_{i+1}$ *for some even* $i \in \mathbb{N}$*;*

2) $\mathcal{M}, \langle x, y\rangle \models \mathsf{unit}_2$ *iff* $x = u_i$ *and* $y = u_{i+1}$ *for some odd* $i \in \mathbb{N}$*.*

Proof sketch. Let $\varphi_{unit1,2}$ be satisfied in an interval $\langle u_0, u_1\rangle$ and let and $u_0, u_1, u_2, \ldots$ be the sequence of immediate $<$-successors. By (3) the propositional variable unit_1 is satisfied in $\langle u_0, u_1\rangle$, and by (4) and (5) each interval which satisfies unit_1 or unit_2 is punctual, so $\langle u_0, u_1\rangle$ is punctual. Then, (6) and (7) states that an interval with unit_1 is always followed by an interval with unit_2, and vice versa. By (8) propositional variables unit_1 and unit_2 cannot be satisfied in the same interval and by (9) all intervals $\langle x, y\rangle$ such that $x < u_0$ do not satisfy unit_1 nor unit_2. It follows that unit_1 is satisfied exactly in intervals $\langle u_i, u_{i+1}\rangle$ for even i and unit_2 in intervals $\langle u_i, u_{i+1}\rangle$ for odd i. □

Next, we will slightly modify formulas from the proofs in [5, Theorem 4.1 and Theorem 4.2] to encode computation of a Turing machine M. For $k \in \{1, 2\}$ define φ_{hornM} as the conjunction of the following formulas, where $N = f(|M|)$:

$$\mathsf{cell}_1^{1,(q_{\mathsf{start}},\rhd)} \bigwedge_{i<N \mid i \geq 1} \mathsf{cell}_1^{i,\sqcup} \tag{10}$$

$$[\mathsf{U}](\mathsf{unit}_k \wedge \mathsf{cell}_k^{i,x} \rightarrow [\mathsf{A}]\mathsf{cell}_k^{i,x,aux}) \tag{11}$$

$$[\mathsf{U}]([\mathsf{E}]\mathsf{cell}_1^{i,x,aux} \rightarrow \overline{\mathsf{cell}}_2^{i,x}) \wedge [\mathsf{U}]([\mathsf{E}]\mathsf{cell}_2^{i,x,aux} \rightarrow \overline{\mathsf{cell}}_1^{i,x}) \tag{12}$$

$$[\mathsf{U}](\overline{\mathsf{cell}}_k^{i,x} \rightarrow \mathsf{unit}_k) \tag{13}$$

$$\bigwedge_{i<N, x \in \Gamma^+} [\mathsf{U}](\mathsf{cell}_k^{i,(q_{\mathsf{halt}},x)} \rightarrow \bot), \tag{14}$$

and for any $(q, x), (q', x') \in \Gamma^+$ such that $\delta(q, x) = (q', x')$:

$$\bigwedge_{i<N} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \rightarrow \mathsf{cell}_k^{i,(q',x')}) \tag{15}$$

$$\bigwedge_{i,j<N \mid i \neq j} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \wedge \overline{\mathsf{cell}}_k^{j,y} \rightarrow \mathsf{cell}_k^{j,y}), \tag{16}$$

and for any $(q, x) \in \Gamma^+$, $q' \in Q$ such that $\delta(q, x) = (q', \mathsf{R})$:

$$\bigwedge_{i<N} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \rightarrow \mathsf{cell}_k^{i,x}) \tag{17}$$

$$\bigwedge_{i<N-1} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \wedge \overline{\mathsf{cell}}_k^{i+1,y} \rightarrow \mathsf{cell}_k^{i+1,(q',y)}) \tag{18}$$

$$\bigwedge_{i<N-1, j<N \mid j \neq i, i+1} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \wedge \overline{\mathsf{cell}}_k^{j,y} \rightarrow \mathsf{cell}_k^{j,y}), \tag{19}$$

and for any $(q, x) \in \Gamma^+$, $q' \in Q$ such that $\delta(q, x) = (q', \mathsf{L})$:

$$\bigwedge_{i<N|i\neq 0} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \rightarrow \mathsf{cell}_k^{i,x}) \tag{20}$$

$$\bigwedge_{i<N|i\neq 0} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \wedge \overline{\mathsf{cell}}_k^{i-1,y} \rightarrow \mathsf{cell}_k^{i-1,(q',y)}) \tag{21}$$

$$\bigwedge_{i<N-1,j<N|i\neq 0 j\neq i,i-1} [\mathsf{U}](\overline{\mathsf{cell}}_k^{i,(q,x)} \wedge \overline{\mathsf{cell}}_k^{j,y} \rightarrow \mathsf{cell}_k^{j,y}). \tag{22}$$

Lemma 2. *Let M be a deterministic Turing machine, which uses a polynomial number of tape cells when starting computation with empty input. Then, the following conditions are equivalent:*

1. $\varphi_{unit1,2} \wedge \varphi_{hornM}$ *is* HS*-satisfiable under (*$\leq$*,S,Dis);*
2. *M diverges with empty input.*

Proof sketch. Intuitively, our aim is to represent that $C_n(i) = x$ (in the n'th step of computation i'th cell contains x) with $\mathsf{cell}_k^{i,x}$ being satisfied in $\langle u_n, u_{n+1}\rangle$, where $k = 1$ if n is even and $k = 2$ if n is odd. Formula (10) encodes the content of M in the first step of computation, (11)–(13) introduce auxiliary variables which pass an information about the content of a cell in a previous step of computation, and by (14) the halting state is never reached. Then, (15)–(16) encode the transition function of a Turing machine in the case when a new symbol is to be written on the tape, (17)–(19) in the case when the head is to be moved one cell to the right, and (20)–(22) when the head is to be moved one cell to the left. It is straight forward to check that φ_{hornM} is satisfiable if an only if the Turing machine diverges. □

Hence, PSpace-Bound BlankNon-Halting reduces polynomially to $\mathsf{HS}^{\Box}_{horn}$-satisfiability under ($\leq$,S,Dis), so we obtain the following complexity result.

Theorem 1. $\mathsf{HS}^{\Box}_{horn}$*-satisfiability under (*$\leq$*,S,Dis) is* PSpace*-hard.*

3.2 Core fragment

Next, we will show that HS_{core}-satisfiability is undecidable under ($\leq$,S,Dis). We will use the well known result that the problem of checking whether a Turing machine diverges on an empty input, denoted by BlankNon-Halting, is undecidable [12]. Given a Turing machine M we will construct an HS_{core}-formula which is satisfiable under ($\leq$,S,Dis) if and only if M diverges with empty input, which implies undecidability of HS_{core} under ($\leq$,S,Dis).

We will extensively use the alternating sequence of unit_1 and unit_2 encoded by the formula $\varphi_{unit1,2}$ (see the previous sub-section). To make a reduction from BlankNon-Halting we will use a technique known from the literature [5, Theorem 4.4] which was used to prove undecidability of HS_{core} under irreflexive semantics. The approach is based on (i) encoding a Cantor-like enumeration of contents of Turing machine cells in consequent steps of computation (see Figure 3) by means of 'up-pointers' whose intended placement is depicted in Figure 4 and then (ii) using the obtained placement of 'up-pointers' to transfer information according to the transition function of a Turing machine. Under the reflexive we will obtain (ii) by a simple adaptation of the known technique but (i) will require non-trivial modifications, which make the proof relatively complex.

Recall, that by $C_n(m) \in \Gamma^+$ we denote the content of the m'th cell in the n'th step of M's computation when starting with empty input. In order to refer to a particular cell in a given step of computation we introduce horizontal and vertical axes containing ordered positive natural numbers – as depicted in Figure 3. Then, a pair of natural numbers, namely x-coordinate and y-coordinate, enables us to refer to any cell of M in any step of computation, e.g., $(1, 1)$ refers to $C_1(1)$.

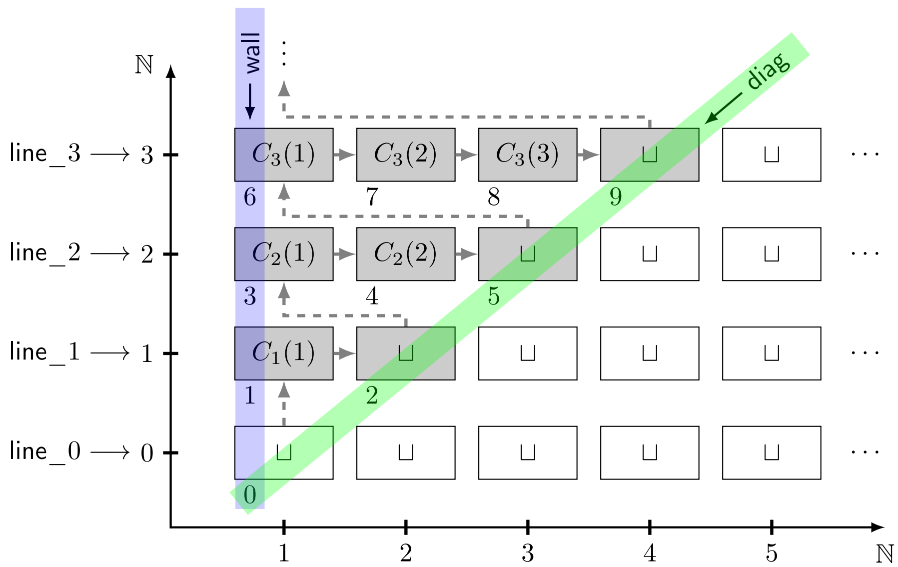


Figure 3: Horizontal lines depict the subsequent contents of M's tape. It suffices to consider cell's (x, y) for $x \leq y$, which are enumerated and marked with grey background.

To encode the computation of M with empty input it suffices to consider cells denoted by pairs $(x, y) \in \mathbb{N}_+ \times \mathbb{N}_+$ such that $x \leq y$, where $\mathbb{N}_+$ is the set of all positive natural numbers, i.e., without 0. Let us denote the set of coordinates referring to these cells by S:

$$S := \{(x, y) \mid x, y \in \mathbb{N}_+, \text{ and } x \leq y\} \cup (1, 0),$$

where (1,0) is an auxiliary element which makes the encoding more convenient. Let *enum* be the enumeration of S as depicted with dashed arrows in Figure 3, i.e., $enum : \mathbb{N} \longrightarrow S$ is a bijective function defined recursively as follows:

($enum1$) $enum(0) = (1, 0)$;

($enum2$) If $enum(n) = (x, y)$ for $x < y$, then $enum(n + 1) = (x + 1, y)$;

($enum3$) If $enum(n) = (x, x)$ for some $x \in \mathbb{N}$, then $enum(n + 1) = (0, x + 1)$;

where n is any natural number. We introduce the relations *wall*, *diag*, *up*, and $line_k$'s for $k \in \mathbb{N}$ in a similar way as it was done in [5], namely:

- $wall \subseteq \mathbb{N} \times \{0, 1\}$ intends to determine numbers lying in the first column of the enumeration *enum*. Hence, the intended definition of *wall* is that for any $n \in \mathbb{N}$ it holds that:

$$wall(n) = 1 \text{ whenever } enum(n) = (0, y) \text{ for some } y \in \mathbb{N}; \tag{23}$$

- $\mathit{diag} \subseteq \mathbb{N} \times \{0,1\}$ intends to determine numbers lying on the diagonal in the enumeration *enum*. The intended definition of *diag* is that for any $n \in \mathbb{N}$:

$$\mathit{diag}(n) = 1 \text{ whenever } \mathit{enum}(n) = (x, x) \text{ for some } x \in \mathbb{N}; \tag{24}$$

- $\mathit{up} \subseteq \mathbb{N} \times \mathbb{N}$ intends to assign to each natural number its direct neighbour from above in the enumeration *enum*. The intended definition of *up* is such that that for any $m, n \in \mathbb{N}$:

$$\begin{gathered}\mathit{up}(m) = n \text{ whenever } \mathit{enum}(m) = (x, y) \text{ and } \mathit{enum}(n) = (x, y+1) \\ \text{for any } x, y \in \mathbb{N};\end{gathered} \tag{25}$$

- $\mathit{line}_k \subseteq \mathbb{N} \times \{0,1\}$ for any $k \in \mathbb{N}$ intends to determine numbers lying in the k-th row in the enumeration *enum*. The intended meaning of line_k is such that for any $n \in \mathbb{N}$:

$$\begin{gathered}\mathit{line}_k(n) = 1 \text{ whenever } \mathit{enum}(n) = (x, k) \text{ for some } x \in \mathbb{N} \\ \text{such that } x \leq k.\end{gathered} \tag{26}$$

Next, we show conditions similar to the ones introduced in [5, Theorem 4.4], which imposed on the relations *wall*, *diag*, *up*, and line_k's force them to obtain the intended meaning. Define:

(c1) $\mathit{line}_0(0) = 1$ and $\mathit{up}(0) = 1$;

(c2) For each n there is exactly one k such that $\mathit{line}_k(n) = 1$;

(c3) If $m < n$, $\mathit{line}_k(m) = 1$, and $\mathit{line}_k(n) = 1$, then for all o such that $m < o < n$ it holds that $\mathit{line}_k(o) = 1$;

(c4) $\mathit{wall}(n) = 1$ whenever for some k we have $\mathit{line}_k(n) = 1$ and $\mathit{line}_k(n-1) = 0$. $\mathit{diag}(n) = 1$ whenever for some k we have $\mathit{line}_k(n) = 1$ and $\mathit{line}_k(n+1) = 0$. Moreover, if $\mathit{line}_k(n) = 1$ and $\mathit{diag}(n) = 1$, then $\mathit{line}_{k+1}(n+1) = 1$;

(c5) If $\mathit{diag}(n) = 0$, then there is m such that $\mathit{up}(m) = n$;

(c6) $\mathit{up} : \mathbb{N} \longrightarrow \mathbb{N}$ is an injective function;

(c7) If $m < n$ then $\mathit{up}(m) < \mathit{up}(n)$;

(c8) If $\mathit{line}_k(m) = 1$ and $\mathit{up}(m) = n$, then $\mathit{line}_{k+1}(n) = 1$.

where m, n, o, k are any natural numbers. The approach used in [5] can be used to show that (c1)–(c8) force (23)–(26):

Lemma 3. *Let* $\mathit{wall} \subseteq \mathbb{N} \times \{0,1\}$, $\mathit{diag} \subseteq \mathbb{N} \times \{0,1\}$, $\mathit{up} \subseteq \mathbb{N} \times \mathbb{N}$, *and* $\mathit{line}_k \subseteq \mathbb{N} \times \mathbb{N}$ *for any* $k \in \mathbb{N}$ *be any relations. If they satisfy conditions (c1)–(c8), then it holds that (23)–(26).*

In what follows we will use the *implication tricks* introduced in [5], which occur to work under ($\leq$,S,Dis) semantics as well. For any HS positive temporal literals λ_1, λ_2, and λ_3, define:

$$\begin{aligned}[\lambda_1 \wedge \lambda_2 \Rightarrow_H \lambda_3] :=\ & [\mathsf{U}](\lambda_1 \to \langle \mathsf{A} \rangle p_1) \wedge [\mathsf{U}](\lambda_2 \to \langle \mathsf{A} \rangle p_2) \wedge [\mathsf{U}](p_2 \to \neg \langle \overline{\mathsf{B}} \rangle p_1) \\ & \wedge [\mathsf{U}](p_1 \to p_3) \wedge [\mathsf{U}](p_1 \to [\overline{\mathsf{B}}] p_3) \wedge [\mathsf{U}](p_2 \to p_3) \wedge [\mathsf{U}](p_2 \to [\mathsf{B}] p_3) \wedge [\mathsf{U}]([\mathsf{A}] p_3 \to \lambda_3); \\ [\lambda_1 \wedge \lambda_2 \Rightarrow_V \lambda_3] :=\ & [\mathsf{U}](\lambda_1 \to \langle \overline{\mathsf{A}} \rangle p_1) \wedge [\mathsf{U}](\lambda_2 \to \langle \overline{\mathsf{A}} \rangle p_2) \wedge [\mathsf{U}](p_2 \to \neg \langle \overline{\mathsf{E}} \rangle p_1) \\ & \wedge [\mathsf{U}](p_1 \to p_3) \wedge [\mathsf{U}](p_1 \to [\overline{\mathsf{E}}] p_3) \wedge [\mathsf{U}](p_2 \to p_3) \wedge [\mathsf{U}](p_2 \to [\mathsf{E}] p_3) \wedge [\mathsf{U}]([\overline{\mathsf{A}}] p_3 \to \lambda_3).\end{aligned}$$

Lemma 4 ([5, Claim 4.1])**.** *Let $\mathcal{M}$ be any* HS*-model and let λ_1, λ_2, λ_3 be any* HS *positive temporal literals. The following hold:*

- *If $\mathcal{M} \models [\lambda_1 \wedge \lambda_2 \Rightarrow_H \lambda_3]$, $\mathcal{M}, \langle x_1, y\rangle \models \lambda_1$, and $\mathcal{M}, \langle x_2, y\rangle \models \lambda_2$ for any x_1, x_2, y, then for all x we have $\mathcal{M}, \langle x, y\rangle \models \lambda_3$.*
- *If $\mathcal{M} \models [\lambda_1 \wedge \lambda_2 \Rightarrow_V \lambda_3]$, $\mathcal{M}, \langle x, y_1\rangle \models \lambda_1$, and $\mathcal{M}, \langle x, y_2\rangle \models \lambda_2$ for any x, y_1, y_2, then for all y we have $\mathcal{M}, \langle x, y\rangle \models \lambda_3$.*

Notice that if $[\lambda_1 \wedge \lambda_2 \Rightarrow_H \lambda_3]$ forces λ_3 to be satisfied in some interval $\langle x, y\rangle$, then it forces λ_3 to be satisfied in all intervals $\langle x', y\rangle$ for any x' (i.e., λ_3 is *horizontally stable* [5]). Similarly, $[\lambda_1 \wedge \lambda_2 \Rightarrow_V \lambda_3]$ forces λ_3 to be *vertically stable.*

The most important and innovative part of the proof is to construct an HS_{core}-formula which encodes *enum*, i.e., conditions (c1)–(c8). The formula will force a specific placement of propositional variables up_1 and up_2, as depicted in Figure 4. Intuitively, $\mathsf{up}_1 \vee \mathsf{up}_2$ being satisfied in $\langle u_m, u_n\rangle$ represents that $up(m) = n-1$, i.e., in the enumeration *enum* the direct up neighbor of m is $n-1$.

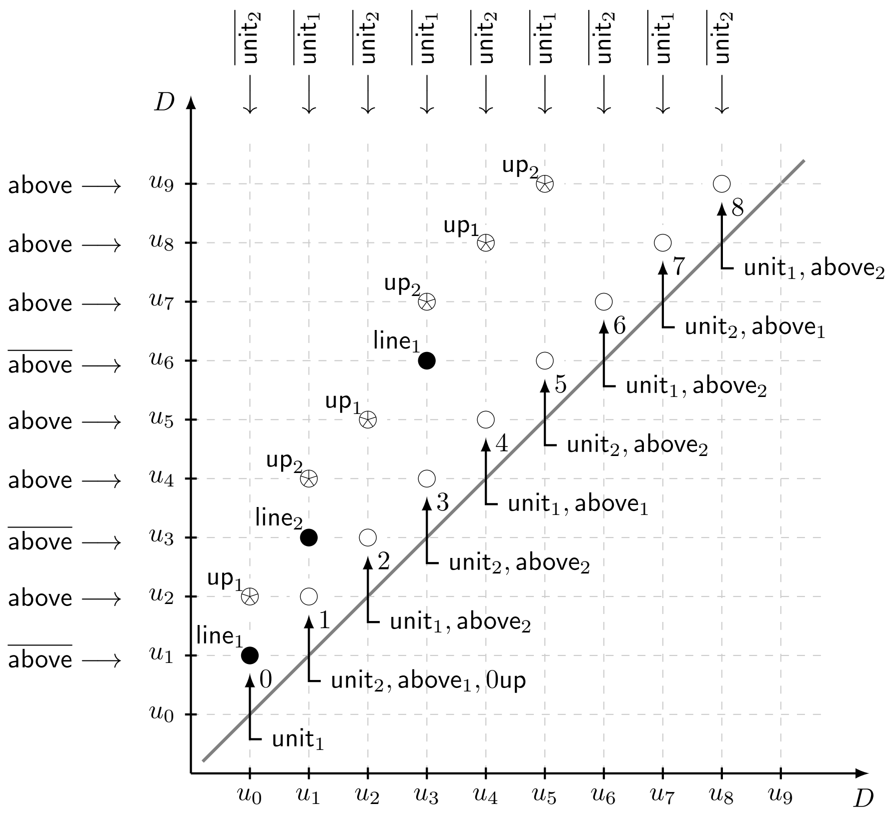


Figure 4: Intended placement of propositional variables, where '○' denotes an interval in which unit is satisfied, '●' an interval in which line_1 or line_2 is satisfied, '⊛' an interval in which up_1 or up_2 is satisfied, whereas 'above ⟶', '$\overline{\mathsf{above}}$ ⟶', '⟵ up_1', and '⟵ up_2' denote lines in which above, $\overline{\mathsf{above}}$, up_1, and up_2 are true, respectively.

We start encoding *enum* by forcing placement of alternating propositional variables unit_1 and

unit_2 in the consequent unit intervals with $\varphi_{\mathsf{unit1,2}}$ (see Lemma 1). Then, we introduce a formula φ_{unit} which forces unit to be satisfied exactly in all intervals in which unit_1 or unit_2 hold:

$$\begin{aligned}
\varphi_{unit} :=& [\mathsf{U}](\mathsf{unit}_k \to \mathsf{unit})\wedge && (27)\\
& [\mathsf{U}](\mathsf{unit} \wedge \overline{\mathsf{unit}} \to \bot)\wedge && (28)\\
& [\mathsf{U}](\mathsf{unit} \to [\mathsf{D}]\mathsf{unit})\wedge && (29)\\
& [\mathsf{unit} \wedge \langle \mathsf{B}\rangle \mathsf{unit}_1 \Rightarrow_H [\mathsf{E}]\overline{\mathsf{unit}_2}] \wedge [\mathsf{U}](\mathsf{unit}_2 \wedge \overline{\mathsf{unit}}_2 \to \bot)\wedge && (30)\\
& [\mathsf{unit} \wedge \langle \mathsf{B}\rangle \mathsf{unit}_2 \Rightarrow_H [\mathsf{E}]\overline{\mathsf{unit}_1}] \wedge [\mathsf{U}](\mathsf{unit}_1 \wedge \overline{\mathsf{unit}}_1 \to \bot), && (31)
\end{aligned}$$

where $k \in \{1, 2\}$, and unit is a propositional variable. The formula (27) forces $\mathsf{unit}_1 \vee \mathsf{unit}_2$ to imply unit, whereas (28)–(31) force unit to imply $\mathsf{unit}_1 \vee \mathsf{unit}_2$. Hence, we obtain the intended placement of unit, namely:

Lemma 5. *Let $\mathcal{M}$ be any HS-model under ($\leq$,S,Dis) semantics. Assume that $\mathcal{M}, \langle u_0, u_1\rangle \models \varphi_{unit1,2} \wedge \varphi_{unit}$ for some interval $\langle u_0, u_1\rangle$ and let $u_1 < u_2 < \dots$ be the infinite sequence of immediate $<$-successors. Then, for any interval $\langle x, y\rangle$ the following conditions are equivalent:*

1. *$\mathcal{M}, \langle x, y\rangle \models \mathsf{unit}$;*
2. *$x = u_i$ and $y = u_{i+1}$ for some $i \in \mathbb{N}$.*

Proof sketch. By (27) we have that $\mathsf{unit}_1 \vee \mathsf{unit}_2$ implies unit. It remains to show the reverse implication. By (9) and (28) unit cannot be satisfied in any $\langle x, y\rangle$ such that $x < u_0$. Then, (29)–(31) disallows unit to be satisfied in any $\langle u_i, u_j\rangle$ such that $i + 1 < j$. It follows that unit implies $\mathsf{unit}_1 \vee \mathsf{unit}_2$. □

Now, we introduce propositional variables line_1 and line_2 which will enable us to distinguish horizontal 'lines' depicted in Figure 3. The intended placement of these variables is such that $\mathsf{line}_1 \vee \mathsf{line}_2$ is satisfied in an interval $\langle u_m, u_n\rangle$ if all numbers from $\{m, m+1, \dots, n\}$ belong to the same line_k from Figure 3. Such a placement is forced by the following formula:

$$\begin{aligned}
\varphi_{line} :=& [\mathsf{U}](\mathsf{line}_1 \to \langle \mathsf{A}\rangle \mathsf{line}_2) \wedge [\mathsf{U}](\mathsf{line}_2 \to \langle \mathsf{A}\rangle \mathsf{line}_1)\wedge && (32)\\
& [\overline{\mathsf{A}}][\overline{\mathsf{B}}][\mathsf{B}]\overline{\mathsf{line}} \wedge [\mathsf{U}](\mathsf{line}_k \wedge \overline{\mathsf{line}} \to \bot)\wedge && (33)\\
& \mathsf{line}_1\wedge && (34)\\
& [\mathsf{U}](\mathsf{line}_k \wedge \langle \overline{\mathsf{A}}\rangle\langle \overline{\mathsf{B}}\rangle \mathsf{line}_k \to \bot) \wedge [\mathsf{U}](\mathsf{line}_k \wedge \langle \mathsf{A}\rangle\langle \overline{\mathsf{E}}\rangle \mathsf{line}_k \to \bot)\wedge && (35)\\
& [\mathsf{U}](\mathsf{line}_1 \wedge \langle \mathsf{B}\rangle\langle \overline{\mathsf{E}}\rangle\langle \mathsf{E}\rangle \mathsf{line}_2 \to \bot)\wedge && (36)\\
& [\mathsf{U}](\mathsf{line}_2 \wedge \langle \mathsf{B}\rangle\langle \overline{\mathsf{E}}\rangle\langle \mathsf{E}\rangle \mathsf{line}_1 \to \bot), && (37)
\end{aligned}$$

where $k \in \{1, 2\}$, and line_1, line_2, and $\overline{\mathsf{line}}$ are propositional variables.

Lemma 6. *Let $\mathcal{M}$ be any HS-model under ($\leq$,S,Dis) semantics. Assume that $\mathcal{M}, \langle u_0, u_1\rangle \models \varphi_{unit1,2} \wedge \varphi_{unit} \wedge \varphi_{line}$ for some interval $\langle u_0, u_1\rangle$. Then, for any interval $\langle x, y\rangle$ the following conditions are satisfied:*

1) *If $\mathcal{M}, \langle x, y\rangle \models \mathsf{line}_1$, then there is z such that $\mathcal{M}, \langle y, z\rangle \models \mathsf{line}_2$;*
2) *If $\mathcal{M}, \langle x, y\rangle \models \mathsf{line}_2$, then there is z such that $\mathcal{M}, \langle y, z\rangle \models \mathsf{line}_1$;*
3) *If $\mathcal{M}, \langle x, y\rangle \models \mathsf{unit}$, then there is exactly one interval $\langle u, w\rangle$ such that $u \leq x$, $w \geq y$, and $\mathcal{M}, \langle u, w\rangle \models \mathsf{line}_1 \vee \mathsf{line}_2$. Moreover, $\langle u, w\rangle$ is such that either $\mathcal{M}, \langle u, w\rangle \models \mathsf{line}_1$ or $\mathcal{M}, \langle u, w\rangle \models \mathsf{line}_2$.*

Proof sketch. The conditions 1) and 2) follow directly from (32). Hence, there is a sequence $\langle u_{k0}, u_{k1}\rangle, \langle u_{k1}, u_{k2}\rangle, \langle u_{k2}, u_{k3}\rangle, \ldots$ with alternating line_1 and line_2. By (34) line_1 is satisfied in $\langle u_0, u_1\rangle$ and by (33) line_1 and line_2 are not satisfied in any interval $\langle x, y\rangle$ such that $x < u_0$. Then, by (35)–(37) we obtain that line_1 and line_2 cannot be satisfied anywhere accept this alternating sequence, which implies the condition 3). □

Finally, we force the intended placement of up_1 and up_2, i.e., we want to force $\mathsf{up}_1 \vee \mathsf{up}_2$ to hold in $\langle u_m, u_n\rangle$ whenever $up(m) = n-1$, i.e., $n-1$ is the direct up neighbor of m in the enumeration *enum*. The particularly important part is to ensure that these 'up-pointers' are functional and injective in a sense that for each u_m there is exactly one u_n pointed by up_1 or up_2, and each u_n is pointed by up_1 or up_2 from at most one u_m. We encode this condition by a quite complex interplay between various propositional variables – mainly between up_1, up_2, and line_1, line_2. We achieve it by means of the following formula:

$$
\begin{aligned}
\varphi_{up} :=\, & \langle \mathsf{A}\rangle \mathsf{0up} \wedge [\mathsf{U}](\mathsf{0up} \to \mathsf{unit}) \wedge && (38)\\
& [\mathsf{U}](\mathsf{0up} \to \mathsf{above}_1) \wedge [\mathsf{U}](\mathsf{0up} \to [\overline{\mathsf{E}}]\mathsf{above}) \wedge && (39)\\
& [\mathsf{above}_k \wedge \mathsf{above} \Rightarrow_H \langle \overline{\mathsf{E}}\rangle\langle \mathsf{E}\rangle \mathsf{up}_k] \wedge && (40)\\
& [\mathsf{U}](\mathsf{0up} \to \overline{\mathsf{up}}_1) \wedge [\overline{\mathsf{A}}][\overline{\mathsf{B}}][\mathsf{B}]\overline{\mathsf{up}}_k \wedge [\mathsf{U}](\mathsf{up}_k \wedge \overline{\mathsf{up}}_k \to \bot) \wedge && (41)\\
& [\overline{\mathsf{E}}][\mathsf{B}]\overline{\mathsf{above}} \wedge && (42)\\
& [\mathsf{U}](\langle \overline{\mathsf{E}}\rangle\langle \mathsf{E}\rangle \mathsf{line}_k \to \overline{\mathsf{above}}) \wedge && (43)\\
& [\mathsf{U}](\mathsf{above} \wedge \overline{\mathsf{above}} \to \bot) \wedge && (44)\\
& [\langle \overline{\mathsf{D}}\rangle \mathsf{line}_k \wedge \langle \mathsf{A}\rangle\langle \overline{\mathsf{E}}\rangle \mathsf{line}_k \Rightarrow_H \mathsf{above}] \wedge && (45)\\
& [\mathsf{U}](\mathsf{up}_1 \to \langle \mathsf{A}\rangle \mathsf{above}_2) \wedge [\mathsf{U}](\mathsf{up}_2 \to \langle \mathsf{A}\rangle \mathsf{above}_1) \wedge && (46)\\
& [\mathsf{U}](\mathsf{above}_k \to \mathsf{unit}) \wedge && (47)\\
& [\mathsf{above}_k \wedge \overline{\mathsf{above}} \Rightarrow_H \langle \mathsf{A}\rangle \mathsf{above}_k] \wedge && (48)\\
& [\mathsf{U}](\mathsf{unit}_k \to \langle \overline{\mathsf{B}}\rangle \mathsf{up}_k) \wedge && (49)\\
& [\mathsf{U}](\mathsf{unit}_1 \to [\overline{\mathsf{B}}]\overline{\mathsf{up}_2}) \wedge [\mathsf{U}](\mathsf{unit}_2 \to [\overline{\mathsf{B}}]\overline{\mathsf{up}_1}) \wedge && (50)\\
& [\mathsf{U}](\overline{\mathsf{above}} \wedge \mathsf{up}_k \to \bot) \wedge && (51)\\
& [\mathsf{U}](\mathsf{up}_1 \wedge \langle \mathsf{D}\rangle \mathsf{up}_2 \to \bot) \wedge [\mathsf{U}](\mathsf{up}_2 \wedge \langle \mathsf{D}\rangle \mathsf{up}_1 \to \bot) \wedge && (52)\\
& [\mathsf{U}](\mathsf{up}_k \wedge \langle \overline{\mathsf{D}}\rangle \mathsf{line}_l \to \bot) \wedge && (53)\\
& [\mathsf{U}](\mathsf{line}_k \to \langle \mathsf{A}\rangle r) \wedge [\mathsf{U}](r \to \mathsf{unit}) \wedge [\mathsf{U}](r \to [\overline{\mathsf{E}}]\mathsf{above}) \wedge && (54)\\
& [\mathsf{up}_k \wedge \langle \mathsf{D}\rangle \mathsf{line}_l \Rightarrow_V \langle \overline{\mathsf{B}}\rangle\langle \mathsf{B}\rangle \mathsf{line}_l], && (55)
\end{aligned}
$$

where $k, l \in \{1, 2\}$, and up_1, up_2, $\overline{\mathsf{up}}_1$, $\overline{\mathsf{up}}_2$, above_1, above_2, above, and $\overline{\mathsf{above}}$ are propositional variables.

Lemma 7. *Let $\mathcal{M} = (\mathbb{D}, I(\mathbb{D}), \{\mathsf{rel_R}\}_{\mathsf{R}\in\mathsf{HS_{rel}}}, V)$ be an* HS*-model under ($\leq$,S,Dis) and $u_0 < u_1 < u_2 < \ldots$ an infinite sequence of immediate $<$-successors in D. Assume that $\mathcal{M}, \langle u_0, u_1\rangle \models \varphi_{unit1,2} \wedge \varphi_{unit} \wedge \varphi_{line} \wedge \varphi_{up}$. Then, for all $\langle x, y\rangle \in I(\mathbb{D})$ the following conditions are equivalent:*

1. $\mathcal{M}, \langle x, y\rangle \models \mathsf{up}_1 \vee \mathsf{up}_2$;
2. *$x = u_i$ and $y = u_j$ for $i, j \in \mathbb{N}$ such that we have $enum(i) = (u, w)$ and $enum(j-1) = (u, w+1)$ for some $u, w \in \mathbb{N}$ (in other words $up(i) = j$ for up defined by (25)).*

Proof sketch. As already mentioned, the main part of the proof is to show that up_1 and up_2 are functional and injective. By (49) each u_m points with an 'up-pointer' to some u_n and by (51) each u_n can be pointed by at most one 'up-pointer' . Hence, it remains to show that no u_m can 'up-point' two u_n's.

First, for each n which is not on a diagonal in enumeration *enum* we mark by means of (45) the interval $\langle u_n, u_{n+1}\rangle$ with a propositional variable above. Then, by (40) and a proper placement of above_1 and above_2 we force that each non diagonal n is 'up-pointed'. The condition that no u_m can 'up-point' two u_n's is forced by (53) and (55), which encode the interplay between line_k's and up_k's. □

We have shown the hard part of the proof, i.e., forcing the placement of propositional variables as presented in Figure 4. Using this placement we can adapt slightly modified formulas from [5, Theorem 4.4] to encode a computation of a Turing machine with empty input. First, we represent transition function δ by means of *triples to cells* function τ. Let:

$$\Sigma := \Gamma - \{\triangleright, \sqcup\}; \qquad Q^- = Q - \{q_{halt}\}; \qquad LEnd = \{\triangleright\} \cup \{Q^- \times \{\triangleright\}\}.$$

Then, define:

$$\begin{aligned} W := &\{\{Q^- \times \Sigma\} \times \Sigma \times \Sigma\} \cup \{\Sigma \times \{Q^- \times \Sigma\} \times \Sigma\} \cup \{\Sigma \times \Sigma \times \{Q^- \times \Sigma\}\} \cup \\ &\{LEnd \times \Sigma \times \Sigma\} \cup \{\{\sqcup\} \times LEnd \times \Sigma\} \cup \{\Sigma \times \{\sqcup\} \times LEnd\} \cup \\ &\{((q_{\mathsf{start}}, \triangleright), \sqcup, \triangleright)\}. \end{aligned}$$

$\tau : W \longrightarrow \Gamma^+$ is such that for any $(x, y, z) \in W$ we have:

$$\tau(x,y,z) = \begin{cases} (q', y) & \text{if either } x \in \{Q - \{q_{\mathsf{halt}}\}\} \times \{\Sigma \cup \{\triangleright\}\} \text{ and } \delta(x) = (q', \mathsf{R}), \\ & \text{or } z \in \{Q - \{q_{\mathsf{halt}}\}\} \times \{\Sigma \cup \{\sqcup\}\} \text{ and } \delta(z) = (q', \mathsf{L}); \\ (q', y') & \text{if } y \in \{Q - \{q_{\mathsf{halt}}\}\} \times \Gamma \text{ and } \delta(y) = (q', y'); \\ y' & \text{if } y = (q, y') \text{ and } \delta(y) = (q', \mathsf{L}) \text{ or } \delta(y) = (q', \mathsf{R}); \\ y & \text{otherwise.} \end{cases}$$

τ determines computation of M in a sense that for any $n, m \in \mathbb{N}_+$ such that $m \leq n$, we have:

$$C_n(m) = \begin{cases} \tau(\sqcup, C_{n-1}(1), C_{n-1}(2)) & \text{if } m = 1; \\ \tau(C_{n-1}(m-1), C_{n-1}(m), C_{n-1}(m+1)) & \text{if } 1 < m < n; \\ \tau(C_{n-1}(n-1), \sqcup, C_n(1)) & \text{if } m = n. \end{cases}$$

For a given Turing machine $M = (\Gamma, Q, q_{start}, q_{halt}, \delta)$ we encode M's computation with empty input as follows:

$$
\begin{aligned}
\varphi_M := & \langle \mathsf{A} \rangle (q_{start}, \rhd) \wedge & (56)\\
& [\mathsf{U}](\mathsf{line}_k \to \langle \mathsf{E} \rangle \sqcup) \wedge & (57)\\
& \bigwedge_{x \in \Gamma} [\mathsf{U}]((q_{halt}, x) \to \bot) \wedge & (58)\\
& \bigwedge_{x \in \Gamma^+} [\mathsf{U}](x \to \mathsf{unit}) \wedge & (59)\\
& \bigwedge_{x, y \in \Gamma^+ | x \neq y} [\mathsf{U}](x \wedge y \to \bot) \wedge & (60)\\
& \bigwedge_{(x,y,z) \in W} \Big([y \wedge \langle \mathsf{A} \rangle z \Rightarrow_H \langle \mathsf{E} \rangle (y, z)] \wedge & (61)\\
& [\mathsf{U}]((y, z) \to \mathsf{unit}) \wedge & (62)\\
& [(y, z) \wedge \langle \overline{\mathsf{A}} \rangle x \Rightarrow_H \langle \mathsf{E} \rangle (x, y, z)] \wedge & (63)\\
& [\mathsf{U}]((x, y, z) \to \mathsf{unit}) \wedge & (64)\\
& [(x, y, z) \wedge \mathsf{unit}_k \Rightarrow_V \langle \overline{\mathsf{B}} \rangle \langle \mathsf{B} \rangle (x, y, z)_k] \wedge & (65)\\
& [\mathsf{U}]((x, y, z)_k \to \mathsf{up}_k) \wedge & (66)\\
& [\mathsf{U}]((x, y, z)_k \to \langle \mathsf{E} \rangle \tau(x, y, z)) \Big), & (67)
\end{aligned}
$$

where $k \in \{1, 2\}$. Since the above encoding is a quite straight forward adaptation of the one in [5, Theorem 4.4] we leave the following lemma without a proof and direct the reader to [5].

Lemma 8. *The following conditions are equivalent for any deterministic Turing machine M:*

1. $\varphi_{unit1,2} \wedge \varphi_{unit} \wedge \varphi_{line} \wedge \varphi_{up} \wedge \varphi_M$ *is* HS*-satisfiable under* ($\leq$,S,Dis)*;*
2. *M diverges with empty input.*

Notice that $\varphi_{unit1,2} \wedge \varphi_{unit} \wedge \varphi_{line} \wedge \varphi_{up} \wedge \varphi_{hornM}$ is an HS_{core}-formula, so the undecidability of HS_{core}-satisfiability under ($\leq$,S,Dis) follows.

Theorem 2. HS_{core}*-satisfiability under* ($\leq$,S,Dis) *is undecidable.*

4 Conclusions

In the paper we have studied computational complexity of HS-fragments under under reflexive semantics in which punctual intervals are forbidden, and a time-line is discrete, denoted by $(\leq, \mathsf{S}, \mathsf{Dis})$. As we have showed, disallowing punctual intervals in reflexive semantics allows us to retain some expressive power which was lost by weakening the semantics (i.e., choosing reflexive rather than irreflexive semantics). The additional expressive power allows us to show new complexity results, namely we have proved that under $(\leq, \mathsf{S}, \mathsf{Dis})$ the satisfiability problem in $\mathsf{HS}^{\Box}_{horn}$ is PSPACE-hard and in HS_{core} it is undecidable. In contrast, we recall that if punctual intervals are allowed or the time-line is dense, then under reflexive semantics the satisfiability problem in $\mathsf{HS}^{\Box}_{horn}$ is P-complete, and its decidability in HS_{core} is an open problem.

One of the important properties of the semantics $(\leq, \mathsf{S}, \mathsf{Dis})$ is that the accessibility relations $\mathsf{rel}_{\mathsf{A}}$ and $\mathsf{rel}_{\overline{\mathsf{A}}}$ is irreflexive. As a result, we were able to introduce (quite sophisticated) formulas simulating a computation of a Turing machine. Among the technical aspects of the proofs a particularly important idea we have developed is to introduce propositional variables with two indices and formulas which force these variables to be satisfied alternately (for an example see Figure 4). This method turned out to be especially important and allowed us to overcome serious problems caused by low expressive power of reflexive semantics.

Among the interesting open problems concerning complexity of HS-fragments under the semantics $(\leq, \mathsf{S}, \mathsf{Dis})$ we plan to study the following:

- Is $\mathsf{HS}^{\Box}_{horn}$ under $(\leq, \mathsf{S}, \mathsf{Dis})$ decidable? If so, what is its computational complexity?
- What is the computational complexity of $\mathsf{HS}^{\Box}_{core}$ under $(\leq, \mathsf{S}, \mathsf{Dis})$? It does not seem that our proofs may be adapted for $\mathsf{HS}^{\Box}_{core}$.

Acknowledgements

This work was supported by the NCN grant 2016/23/N/HS1/02168 and by the Foundation for Polish Science (FNP).

References

[1] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Tractable interval temporal propositional and description logics. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, pages 1417–1423. 2015.

[3] Davide Bresolin, Dario Della Monica, Angelo Montanari, Pietro Sala, and Guido Sciavicco. Interval temporal logics over finite linear orders: the complete picture. In *ECAI*, pages 199–204, 2012.

[4] Davide Bresolin, Dario Della Monica, Angelo Montanari, and Guido Sciavicco. The light side of interval temporal logic: the Bernays-Schönfinkel fragment of CDT. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):11–39, 2014.

[5] Davide Bresolin, Agi Kurucz, Emilio Muñoz-Velasco, Vladislav Ryzhikov, Guido Sciavicco, and Michael Zakharyaschev. Horn fragments of the Halpern-Shoham interval temporal logic. *ACM Transactions on Computational Logic (TOCL)*, 18(3):22:1–22:39, 2017.

[6] Dario Della Monica, Valentin Goranko, Angelo Montanari, and Guido Sciavicco. Interval temporal logics: a journey. *Bulletin of EATCS*, 3(105), 2013.

[7] Michael Fisher. A resolution method for temporal logic. In *IJCAI*, volume 91, pages 99–104, 1991.

[8] Valentin Goranko, Angelo Montanari, and Guido Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

[9] Joseph Y Halpern and Yoav Shoham. A propositional modal logic of time intervals. *Journal of the ACM (JACM)*, 38(4):935–962, 1991.

[10] Angelo Montanari, Ian Pratt-Hartmann, and Pietro Sala. Decidability of the logics of the reflexive sub-interval and super-interval relations over finite linear orders. In *Temporal Representation and Reasoning (TIME), 2010 17th International Symposium on*, pages 27–34. IEEE, 2010.

[11] Angelo Montanari, Gabriele Puppis, and Pietro Sala. Maximal decidable fragments of Halpern and Shoham's modal logic of intervals. In *Automata, Languages and Programming*, pages 345–356. Springer, 2010.

[12] Bernard M. Moret. *The Theory of Computation*. Addison-Wesley, 1998.

[13] Yde Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.