
A First Study of the Horn Fragment of the Modal Logic of Time Intervals*

D. Bresolin¹, E. Muñoz-Velasco², and G. Sciavicco³

¹ Department of Computer Science and Engineering
University of Bologna (Italy) (davide.bresolin@unibo.it)

² Department of Applied Mathematics
University of Malaga (Spain) (emilio@ctima.uma.es)

³ Department of Information, Engineering and Communications
University of Murcia (Spain) (guido@um.es)

Abstract. Interval temporal logics provide a natural framework for temporal reasoning about interval structures over linearly ordered domains, where intervals are taken as the primitive ontological entities. The most influential propositional interval-based logic is probably Halpern’s and Shoham Modal Logic of Time Intervals, a.k.a. HS. While most studies focused on the computational properties of the syntactic fragments that arise by considering only a subset of the set of modalities, the fragments that are obtained by weakening the propositional side of HS have received much less attention. Here, we approach this problem by considering the Horn fragment of HS and proving that the satisfiability problem remains undecidable, at least for discrete linear orders.

1 Introduction

Most temporal logics proposed in the literature assume a point-based model of time, and they have been successfully applied in a variety of fields. However, there are examples of relevant application domains, such as planning and temporal databases, where interesting problems are dealt with in an unsatisfactory way by point-based formalisms [9, 10]. Interval temporal logics provide a natural framework for temporal reasoning about interval structures over linearly (or partially) ordered domains. They take time intervals as the primitive ontological entities and define truth of formulas relative to time intervals, rather than time points; their modalities correspond to various relations between pairs of intervals. In particular, the well-known logic HS [6] features a set of modalities that make it possible to express all Allen’s interval relations [1]. Unfortunately, in interval temporal logic undecidability is the rule and decidability the exception: HS is undecidable when interpreted on most meaningful classes of linearly ordered sets, and limiting the set of temporal modalities of the logic is not always sufficient to achieve decidability. For example, when formulas are interpreted over strongly discrete linear orders, only 44 decidable fragments exist [3].

* The authors acknowledge the support from the Italian GNCS Project “Automata, games and temporal logics for verification and synthesis of safety-critical systems” (D. Bresolin), the Spanish Project TIN12-39353-C04-01 (E. Muñoz-Velasco), and the Spanish fellowship program ‘Ramon y Cajal’ RYC-2011-07821 (G. Sciavicco).

In this context, it makes sense to study sub-propositional fragments of HS, such as the *Horn* fragment. Horn fragments of modal and temporal logics have been studied, for example, in [2, 4, 5, 7]. Being sub-propositional, the obvious question is whether or not the satisfiability problem of Horn HS remains undecidable, and, if so, on which classes of linearly ordered sets. The results presented in this preliminary study proves that, unfortunately, at least in the class of models built over \mathbb{Z} and other strongly discrete linear orders, this is the case. While discouraging, these results should be seen as the first attempt of studying sub-propositional fragments of interval temporal logics. Horn HS is still undecidable, but we might find out that the decidability frontier is different from the one for the syntactical fragments of full HS, and/or that some of the decidable fragments present a better computational behaviour.

2 The Logic HS and its Horn Fragment

Halpern and Shoham's logic HS is a multi-modal logic with formulas built on a set \mathcal{AP} of proposition letters, the Boolean connectives \vee and \neg , plus the six modalities to capture the existence of an interval in a particular Allen's relation with the current one:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle A \rangle \varphi \mid \langle \bar{A} \rangle \varphi \mid \langle B \rangle \varphi \mid \langle \bar{B} \rangle \varphi \mid \langle E \rangle \varphi \mid \langle \bar{E} \rangle \varphi$$

The other Boolean connectives, the box modalities, and the temporal modalities corresponding to other Allen's relations (such as *during*, $\langle D \rangle$, *later*, $\langle L \rangle$, and *overlaps*, $\langle O \rangle$) are definable in the language (e.g. $[A]\varphi \equiv \neg\langle A \rangle\neg\varphi$, $\langle D \rangle\varphi = \langle E \rangle\langle B \rangle\varphi$, and $\langle L \rangle\varphi = \langle A \rangle\langle A \rangle\varphi$).

Let $\mathbb{D} = \langle D, < \rangle$ be a discrete linearly ordered set. An *interval* over \mathbb{D} is an ordered pair $[x, y]$, where $x, y \in D$ and $x < y$ (*strict semantics*). An interval of the type $[x, x+1]$ is called *unit*. The semantics of HS is given in terms of *interval models* $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $\mathbb{I}(\mathbb{D})$ is the set of all intervals over \mathbb{D} and $V : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})}$ is a *valuation function* that assigns to every $p \in \mathcal{AP}$ the set of intervals $V(p)$ over which p holds. The *truth relation* \Vdash of a formula over a given interval $[x, y]$ in an interval model M is defined by structural induction on formulas (Boolean connectives are dealt with in the standard way):

- $M, [x, y] \Vdash p$ iff $[x, y] \in V(p)$;
- $M, [x, y] \Vdash \langle A \rangle \varphi$ iff there exists $z > y$ such that $M, [y, z] \Vdash \varphi$;
- $M, [x, y] \Vdash \langle \bar{A} \rangle \varphi$ iff there exists $z < x$ such that $M, [z, x] \Vdash \varphi$;
- $M, [x, y] \Vdash \langle B \rangle \varphi$ iff there exists $x < z < y$ such that $M, [x, z] \Vdash \varphi$;
- $M, [x, y] \Vdash \langle \bar{B} \rangle \varphi$ iff there exists $z > y$ such that $M, [x, z] \Vdash \varphi$;
- $M, [x, y] \Vdash \langle E \rangle \varphi$ iff there exists $x < z < y$ such that $M, [z, y] \Vdash \varphi$;
- $M, [x, y] \Vdash \langle \bar{E} \rangle \varphi$ iff there exists $z < x$ such that $M, [z, y] \Vdash \varphi$.

A HS-formula φ is *satisfiable* if and only if there exists a model M and an interval $[x, y]$ such that $M, [x, y] \Vdash \varphi$. The *satisfiability problem* for HS is the problem of finding a model and an interval that satisfies a formula.

Following [7], we define the Horn fragment of HS. First, consider only formulas in *negative normal form* (nnf) (i.e., with only \wedge and \vee as Boolean connectives and such that \neg occurs only in front of propositions). A formula in nnf is called *negative* if and

only if every proposition is prefixed by negation; it is *non-negative* if it is not negative, and *positive* if its negation is a negative formula. An HS-formula φ is called a *Horn HS-formula* (or, simply, *Horn*) if and only if (i) φ is a proposition; (ii) φ is a negative formula; (iii) $\varphi = [X]\psi$, $\varphi = \langle X \rangle \psi$, or $\varphi = \psi \wedge \rho$, where ψ and ρ are Horn and $\langle X \rangle$ is any HS modality; (iv) $\varphi = \psi \rightarrow \rho$, where ψ is positive and ρ is Horn; (v) φ is a disjunction of a negative formula and a Horn formula. Alternative definitions of modal Horn clauses can be found, for example, in [2, 4, 5]. However, these definitions are equivalent to Nguyen's one [7], in the sense that every given set of Horn formulas (or clauses) in the latter can be translated into set of equi-satisfiable Horn clauses of any of the formers.

In the following we will make use of the *universal* modality $[U]$, that can be defined in Horn HS as a conjunction of boxes ($[U]\varphi = [\bar{A}][\bar{A}][A]\varphi \wedge [\bar{A}][A]\varphi \wedge [\bar{A}][A][A]\varphi$), and of the *difference* modality $[\neq]$, meaning *for every interval, except the current one*, that can be defined in a similar way.

3 Undecidability of Horn HS

In this section, we assume that Horn HS is interpreted over \mathbb{Z} . Notice that this assumption can be immediately relaxed, to include Horn HS interpreted over \mathbb{N} , over the class of all strongly discrete linear orders, and over the class of finite linear orders. Our construction, that closely follows both the original undecidability proof for full HS [6] and more recent undecidability proofs for fragments of HS [3], is based on a reduction from the *halting problem* of a deterministic Turing Machine on empty input [8].

A *Turing Machine* is defined as a tuple $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$, where Q is the set of states, q_0 (resp., q_f) is the initial (resp., final) state, Σ is the machine's alphabet that does not contain \sqcup (blank), $\Gamma = \Sigma \cup \{\sqcup\}$ is the tape alphabet, and $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function (L, R represent the possible moves on the machine's tape: left, right). Even under the assumption that $\Sigma = \{0, 1\}$ and that the input is empty, the halting problem for a deterministic Turing Machine is undecidable [8].

In the following, we reduce the halting problem of a Turing Machine to the satisfiability problem for Horn HS over \mathbb{Z} . The underlying idea is to represent the *computation history* of the machine using the propositional symbol $*$ to separate successive configurations, the propositional symbols $0, 1, \sqcup$ to represent tape cells *not* under the machine's head, and propositional symbols q^c , with $q \in Q \setminus \{q_f\}$ and $c \in \{0, 1, \sqcup\}$, to represent the tape cell under the head and the current (non-final) state of the machine. We will use the propositional symbols Fr to encode the initial configuration, u to represent the units, Co to represent a generic configuration, and Cr to connect successive configurations. When the machine is in the final state q_f the computation immediately halts. For this reason we can discard the symbol under the head and use a unique propositional letter q_f . We denote by \mathcal{L} the set $\{0, 1, \sqcup, *\} \cup \{q^c \mid q \in Q \setminus \{q_f\} \wedge c \in \{0, 1, \sqcup\}\} \cup \{q_f\}$. Any terminating run of the Turing Machine will make use of only a finite portion of the tape. Hence, we can assume w.l.o.g. that all configurations have the same length and that they are long enough to contain the relevant part of the tape.

$$\langle A \rangle u \wedge [U](u \rightarrow (\langle A \rangle u \wedge [B] \perp)) \quad \text{units' structure}$$

$$\begin{array}{ll}
[U] \bigwedge_{l \in \mathcal{L}} (l \rightarrow u) & \text{tape/state propositions and } * \text{ are units} \\
[U] \bigwedge_{l, l' \in \mathcal{L}, l \neq l'} (l \rightarrow \neg l') & \text{tape/state propositions and } * \text{ are unique} \\
\langle A \rangle (* \wedge \langle A \rangle Fr) \wedge \langle A \rangle Co \wedge [U] (Fr \rightarrow [\neq] \neg Fr) & \text{initial configuration} \\
[U] (Fr \rightarrow (\langle B \rangle q_0^\sqcup \wedge \langle E \rangle * \wedge ([D] (u \rightarrow \sqcup)))) & Fr \text{ structure} \\
[U] (Co \rightarrow (\langle A \rangle Co \wedge \langle B \rangle * \wedge \langle E \rangle * \wedge [D] \neg *)) & \text{configuration sequence} \\
[U] (Co \rightarrow ([D] \neg Co \wedge [B] \neg Co \wedge [E] \neg Co)) & \text{configuration structure}
\end{array}$$

Intuitively, the above formulas guarantee that *configurations* (denoted by Co) are built of units, each one of them contains either $*$ or a tape/head element and that there is an infinite and unique sequence of configurations. The proposition Fr sets the structure of the first configuration: a single $*$, followed by q_0 reading blank, and a number of blanks, followed by a $*$.

The relation between successive configurations is maintained by the proposition Cr (*corresponds*), constrained by the following formulas. These are also used to guarantee that all configurations have the same length.

$$\begin{array}{ll}
[U] (u \rightarrow (\langle A \rangle Cr \wedge \langle \bar{A} \rangle Cr)) & \text{each unit starts and ends a "Cr"} \\
[U] (Cr \rightarrow ([B] \neg Cr \wedge [D] \neg Cr \wedge [E] \neg Cr)) & \text{"Cr" structure} \\
[U] (Co \rightarrow (\neg Cr \wedge [D] \neg Cr \wedge [\bar{D}] \neg Cr \wedge [\bar{E}] \neg Cr)) & \text{"Cr/Co" relation}
\end{array}$$

It remains to ensure that the machine \mathcal{A} behaves as imposed by δ . In the encoding of the transition function, we treat as special cases the situations in which (i) the head is at the last cell of the segment of the tape currently shown and the head must be moved to the right and, (ii) the head is at the first cell of the tape and the head must be moved to the left. In the following formulas, $c, c', c'' \in \{0, 1, \sqcup\}$, $d \in \{0, 1, \sqcup, *\}$, and $q, q' \in Q$ (by a little abuse of notation, we assume that all symbols q_f^c are equal to q_f).

$$\begin{array}{ll}
\bigwedge_{d, \delta(q,c)=(q',c',R), c''} [U] ((q^c \wedge \langle A \rangle c'' \wedge \langle \bar{A} \rangle d) \rightarrow & \text{to the right, not the last cell} \\
[A] (Cr \rightarrow \langle A \rangle (c' \wedge \langle A \rangle q^{c''} \wedge \langle \bar{A} \rangle d)) & \\
\bigwedge_{\delta(q,c)=(q',c',R)} [U] ((q^c \wedge \langle A \rangle *) \rightarrow \perp) & \text{to the right, last cell: forbidden} \\
\bigwedge_{d, \delta(q,c)=(q',c',L), c''} [U] ((q^c \wedge \langle \bar{A} \rangle c'' \wedge \langle A \rangle d) \rightarrow & \text{to the left, not the first cell} \\
[A] (Cr \rightarrow \langle A \rangle (c' \wedge \langle \bar{A} \rangle q^{c''} \wedge \langle A \rangle d)) & \\
\bigwedge_{d, \delta(q,c)=(q',c',L)} [U] ((q^c \wedge \langle \bar{A} \rangle *) \wedge \langle A \rangle d) \rightarrow & \text{to the left, first cell} \\
[A] (Cr \rightarrow \langle A \rangle (q^{c'} \wedge \langle \bar{A} \rangle * \wedge \langle A \rangle d)) &
\end{array}$$

Finally, we force cells located away from the head to remain unchanged.

$$\bigwedge_{c, c', c''} [U] ((c \wedge \langle \bar{A} \rangle c' \wedge \langle A \rangle c'') \rightarrow [A] (Cr \rightarrow \langle A \rangle c))$$

The conjunction $\varphi_{\mathcal{A}}$ of all formulas above encodes a computation of the Turing Machine.

Theorem 1. *Let \mathcal{A} be a deterministic Turing Machine. Then, \mathcal{A} halts on an empty input if and only if the Horn HS-formula $\varphi_{\mathcal{A}} \wedge \langle L \rangle q_f$ is satisfiable over \mathbb{Z} .*

This proves that the satisfiability problem for Horn HS interpreted over \mathbb{Z} is undecidable. The construction can be rephrased to deal with the cases where Horn HS is interpreted over \mathbb{N} , or in the class of all strongly discrete linear orders, or over finite linear orders.

4 Conclusions

Sub-propositional fragments, such as the Horn fragment, of classical and modal logics have been extensively studied. The generally high complexity of the (few) decidable interval temporal logics justifies the study of sub-propositional fragments of interval logics in search of expressive languages that present a better computational behaviour. In this paper we proved a first negative result in this sense, by showing that the well-known interval temporal logic HS is still undecidable when its Horn fragment is considered, at least in the discrete case. Nevertheless, we believe that syntactical fragments of Horn HS should be studied in the same way in which syntactical fragments of full HS have been. In the long run, we plan to consider the Horn fragments of decidable interval logics like $A\bar{A}$ and $AB\bar{B}L$, to understand whether or not their satisfiability problem present a better computational behaviour.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyashev. The complexity of clausal fragments of LTL. In *Proc. of the 19th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 8312 of *LNCS*, pages 35–52. Springer, 2013.
3. D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Interval temporal logics over strongly discrete linear orders: the complete picture. In *Proc. of the 4th International Symposium on Games, Automata, Logics, and Formal Verification (GANDALF)*, volume 96 of *EPTCS*, pages 155–169, 2012.
4. C. Chen and I. Lin. The computational complexity of the satisfiability of modal Horn clauses for modal propositional logics. *Theoretical Computer Science*, 129(1):95–121, 1994.
5. L. Fariñas Del Cerro and M. Penttonen. A note on the complexity of the satisfiability of modal Horn clauses. *Journal of Logic Programming*, 4(1):1–10, 1987.
6. J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
7. L. Nguyen. On the complexity of fragments of modal logics. *Advances in Modal Logic*, 5:318–330, 2004.
8. M. Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.
9. D. E. Smith. The case for durative actions: a commentary on PDDL2.1. *Journal of Artificial Intelligence Reasoning*, 20(1):149–154, Dec. 2003.
10. D. Toman. Point vs. interval-based query languages for temporal databases. In *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems*, pages 58–67. ACM, 1996.