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

D. Bresolin$^1$, E. Muñoz-Velasco$^2$, and G. Sciavicco$^3$

$^1$ 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 no 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, a number of relevant application domains, such as planning and synthesis of controllers, are characterized by advanced features that are neglected or dealt with in an unsatisfactory way by point-based formalisms. 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

* 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).
sufficient to achieve decidability. For example, when formulas are interpreted over strongly discrete linear orders, only 44 decidable fragments exists [3].

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 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 these fragments presents 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 AP of proposition letters, the boolean connectives ∨ and ¬, plus the six modalities to capture the existence of an interval in a particular Allen’s relation with the current one:

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \varphi \mid \langle B \rangle \varphi \mid \langle \overline{A} \rangle \varphi \mid \langle \overline{B} \rangle \varphi \mid \langle E \rangle \varphi \mid \langle \overline{E} \rangle \varphi \]

The other boolean connectives, the box modalities, and the temporal modalities corresponding to other Allen’s relations (such as during, (D), later, (L), and overlaps, (O)) are definable in the language.

Let \( D = \langle D, \prec \rangle \) be a discrete linearly ordered set. An interval over \( D \) 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 \( M = \langle I(D), V \rangle \), where \( I(D) \) is the set of all intervals over \( D \) and \( V : AP \mapsto 2^{I(D)} \) is a valuation function that assigns to every \( p \in AP \) the set of intervals \( V(p) \) over which \( p \) holds. The truth relation \( \models \) of a formula over a given interval \( [x, y] \) in an interval model \( M \) is defined by structural induction on formulas:

- \( M, [x, y] \models p \iff [x, y] \in V(p) \);
- boolean connectives are dealt with in the standard way;
- \( M, [x, y] \models \langle A \rangle \varphi \iff \text{there exists } z > y \text{ such that } M, [y, z] \models \varphi \);
- \( M, [x, y] \models \langle B \rangle \varphi \iff \text{there exists } z < x \text{ such that } M, [z, x] \models \varphi \);
- \( M, [x, y] \models \langle \overline{A} \rangle \varphi \iff \text{there exists } x < z < y \text{ such that } M, [x, z] \models \varphi \);
- \( M, [x, y] \models \langle \overline{B} \rangle \varphi \iff \text{there exists } z > y \text{ such that } M, [z, x] \models \varphi \);
- \( M, [x, y] \models \langle E \rangle \varphi \iff \text{there exists } x < z < y \text{ such that } M, [z, y] \models \varphi \);
- \( M, [x, y] \models \langle \overline{E} \rangle \varphi \iff \text{there exists } z < x \text{ such that } M, [z, y] \models \varphi \).

A HS-formula \( \varphi \) is satisfiable if and only if there exists a model \( M \) and an interval \([x, y]\) such that \( M, [x, y] \models \varphi \). 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 $\land$ and $\lor$ 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 primitive 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 $\varphi$ is called a Horn HS-formula (or, simply, Horn) if and only if

(i) $\varphi$ is a proposition;
(ii) $\varphi$ is a negative formula;
(iii) $\varphi = [X]\psi$, $\varphi = (X)\psi$, or $\varphi = \psi \land \rho$, where $\psi$ and $\rho$ are Horn and $(X)$ is any HS modality; (iv) $\varphi = \psi \rightarrow \rho$, where $\psi$ is positive and $\rho$ is Horn; (v) $\varphi$ 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, 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 former.

Notice that the (definable) universal modality $[U]$ can be considered as part of the language of Horn HS, because it is defined as a conjunction of boxes (for example, over $\mathbb{Z}$, we have that $[U]\varphi = [A][A][A]\varphi \land [A][A]\varphi \land [A][A][A]\varphi$; the same holds for the so-called difference modality $[\neq]$, meaning for every interval, except the current one.

### 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 $A = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$, where $Q$ is the set of states, $q_0$ (resp., $q_f$) is the initial (resp., final) state, $\Sigma$ is the machine’s alphabet that does not contain $\sqcup$ (blank), $\Gamma = \Sigma \cup \{\sqcup\}$ (without loss of generality) is the tape alphabet, and $\delta : Q \times \Gamma \to Q \times \Sigma \times \{L, R\}$ is the transition function ($L, R$ represent the possible moves on the machine’s tape: left, right). Even under very restrictive assumptions, that is, that $\Sigma = \{0, 1\}$ and that the input is empty, the halting problem on a empty input for a deterministic Turing Machine is undecidable [8]. In the rest of this section, we reduce the halting problem of a Turing Machine to the satisfiability problem for Horn HS over $\mathbb{Z}$. While our construction does not contain any idea essentially different from other similar reductions, it is not obvious at all that it can be done in the considered sub-propositional fragment of HS, which strongly limits the use of disjunctions.

The underlying idea is to represent the computation history of a given Turing Machine using the propositional symbols $*$ 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 shall also use the propositional symbols Fr to correctly encode the initial configuration, u to represent the unit intervals, Co to represent a generic configuration, and Cr to connect successive configurations. When the machine is in the final state qf the computation immediately halts. For this reason we can discard the symbol under the head and use a unique propositional letter qf.

We denote by L the set \( \{0, 1, \underline{\bot}, *\} \cup \{q^c | q \in Q \setminus \{q_f\} \land c \in \{0, 1, \underline{\bot}\} \} \cup \{q_f\} \). A configuration represents the (always finite) content of the tape.

\[
\langle A \rangle u \land [U](u \rightarrow ((\langle A \rangle u \land [B] \bot)) \text{ unit intervals structure}
\]

\[
[U] \bigwedge_{l \in L}(l \rightarrow u) \text{ tape/state pro positions and * are units}
\]

\[
[U] \bigwedge_{l \neq l'}(l \rightarrow \lnot l') \text{ tape/state propositions and * are unique}
\]

\[
\langle A \rangle \ast \land \langle A \rangle Fr \land \langle A \rangle Co \land [U]([Fr \rightarrow \lnot \ast = Fr]) \text{ initial configuration}
\]

\[
[U]([Fr \rightarrow ([B] q^c_1 \land [E] \ast \land ([D](u \rightarrow \bot)))]) \text{ Fr structure}
\]

\[
[U]([Co \rightarrow ([A] Co \land [B] \ast \land [E] \ast \land [D] \lnot \ast]) \text{ configuration sequence}
\]

\[
[U]([Co \rightarrow ([D] \lnot Co \land [B] \lnot Co \land [E] \lnot Co)) \text{ configuration structure}
\]

Intuitively, the conjunction of the above formulas guarantees that configurations (denoted by Co) are built of units, each one of them contains either * or a tape/head elements, which are unique, and that there is an infinite and unique sequence of configurations, starting with the first one: a single *, followed by q0 reading blank, and a number of unit intervals containing blank, followed by a *; the proposition Fr, holding over the maximal ending interval of the the first Co, is used to set the structure of the first configuration.

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.

\[
[U]([u \rightarrow ((\langle A \rangle Cr \land \lnot(A) Cr)) \text{ each unit interval starts and ends a “Cr”}
\]

\[
[U]([Cr \rightarrow ([B] \lnot Cr \land [D] \lnot Cr \land [E] \lnot Cr)) \text{ “Cr” structure}
\]

\[
[U]([Co \rightarrow (\lnot Cr \land \lnot(D) Cr \land \lnot(D) \lnot Cr \land \lnot(E) Cr) \text{ “Cr/Co” relation}
\]

It remains to ensure that the machine A behaves as imposed by \( \delta \). 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, \( c, c', c'' \in \{0, 1, \underline{\bot}\}, d \in \{0, 1, \underline{\bot}, *\}, \) and \( q, q' \in Q \) (by a little abuse of notation, we assume that all symbols \( q^c \) are equal to \( q_f \)).

\[
\bigwedge_{d, \delta(q,c)=(q',c',R),c''} [U]((q^c \land \langle A \rangle c'' \land \lnot(A) d) \rightarrow \text{ to the right, not the last cell}
\]

\[
[A][Cr \rightarrow \langle A \rangle(c' \land \langle A \rangle q^{c''} \land \lnot(A) d))
\]

\[
\bigwedge_{d, \delta(q,c)=(q',c',R),c''} [U]((q^c \land \langle A \rangle \ast) \rightarrow \bot \text{ to the right, last cell: forbidden}
\]

\[
[A][Cr \rightarrow \langle A \rangle(c' \land \lnot(A) q^{c''} \land \langle A \rangle d))
\]

\[
\bigwedge_{d, \delta(q,c)=(q',c',L),c''} [U]((q^c \land \lnot(A) c'' \land \langle A \rangle d) \rightarrow \text{ to the left, not the first cell}
\]

\[
[A][Cr \rightarrow \langle A \rangle(c' \land \lnot(A) q^{c''} \land \langle A \rangle d))
\]

\[
\bigwedge_{d, \delta(q,c)=(q',c',L)} [U]((q^c \land \lnot(A) \ast \land \langle A \rangle d) \rightarrow \text{ to the left, first cell}
\]

\[
[A][Cr \rightarrow \langle A \rangle(q^{c''} \land \lnot(A) \ast \land \langle A \rangle d))
\]
Finally, we ensure that cells located away from the head are kept between configurations.

\[ \bigwedge_{c,c',c''} \left[ U \left( (c \land (\neg A)c' \land (A)c'') \rightarrow [A](Cr \rightarrow ((A)(c \land (A)c'') \land (\neg A)c')) \right) \right] \]

Let us call \( \varphi_A \) the conjunction of all of the above. The following result holds.

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

As a corollary, we deduce that the satisfiability problem for Horn HS interpreted over \( \mathbb{Z} \) is undecidable. The above construction can be slightly rephrased to deal with the cases where Horn HS are 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-based temporal logics justifies a certain interest in exploring the Horn fragment of them in search of languages that present a better computational behaviour, and yet are, expressiveness-wise, suitable for some applications. 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, for example, \( AA \) and \( ABBL \), to understand whether or not their satisfiability problem present a better computational behaviour.

### References