Sharpening the undecidability of interval temporal logic

Kamal Lodaya

Institute of Mathematical Sciences,
C.I.T. Campus, Taramani, Chennai 600 113, India
Email: kamal@imsc.ernet.in

1 Motivation and results

Interval temporal logics (ITLs) were introduced in the philosophy of time (see [Ben95] for a survey) but have proved useful in artificial intelligence and computer science [All83,HMM83,HS91,ZHR91]. They provide a rich specification language for systems working with dense time (for example, [RRM93]). By now, there is a whole menagerie of ITLs. In this paper, we work with the simplest (propositional) ITLs and discuss their decidability.

Let T be a set linearly ordered by $\geq$. An interval $[a, b]$ is defined as usual for $a, b \in T,\ \ b \geq a$.

Formulas of the basic propositional Interval Temporal Logic C build on a countable collection of propositions by closing under the logical connectives $\neg, \vee$ and $\frown$ ("chop"). The other Boolean connectives are defined as usual. We identify a special proposition ℓ_0 which we will use to mark intervals $[a, a]$ consisting of one point [Ven91]. To be more precise, the logic can be called C_0 [LR00] but in this paper we use the simpler notation.

Given a proposition p, a valuation $\mathcal{V}$ assigns intervals on which it is true, subject to the condition that the special proposition ℓ_0 is made true exactly on point intervals of the form $[a, a]$. A model is a pair $\mathcal{M} = ((T, \geq), \mathcal{V})$.

A formula is assigned truth value inductively, the only nontrivial case being for the "chop" operator:

- $\mathcal{M}, [a, b] \models p$ iff $[a, b] \in \mathcal{V}(p)$,

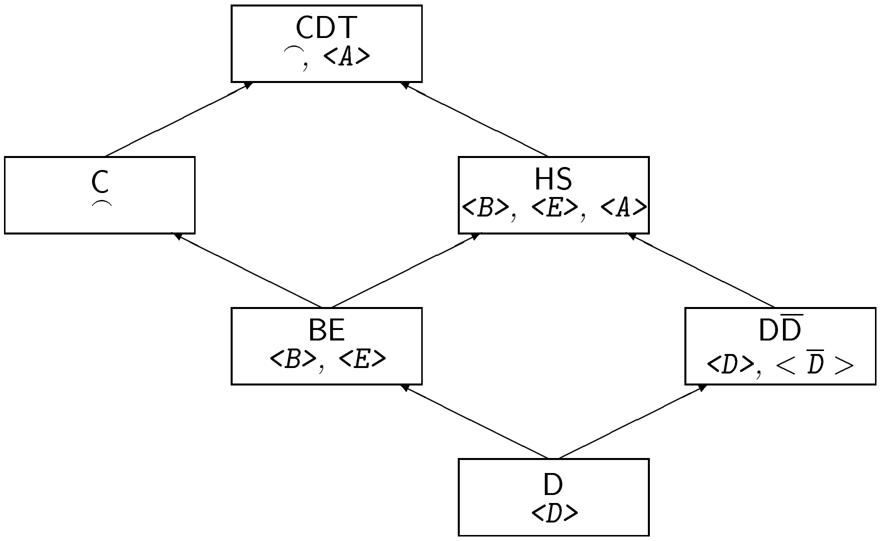
– $\mathcal{M}, [a, b] \models \phi \vee \psi$ iff $\mathcal{M}, [a, b] \models \phi$ or $\mathcal{M}, [a, b] \models \psi$,
– $\mathcal{M}, [a, b] \models \phi{}^\frown\psi$ iff $\mathcal{M}, [a, m] \models \phi$ and $\mathcal{M}, [m, b] \models \psi$, for some m such that $b \geq m \geq a$.

An example formula is $\neg\ell_0{}^\frown\phi{}^\frown\neg\ell_0$, which we denote <D>$\phi$. It says that ϕ holds somewhere within an interval. Its dual [D]ϕ is $\neg$<D>$\neg\phi$. Similarly <B>ϕ and <E>ϕ are defined to be $\phi{}^\frown\neg\ell_0$ and $\neg\ell_0{}^\frown\phi$ respectively, and they specify assertions holding in a "beginning" or "ending" interval. As an aside, note that the proposition ℓ_0 can be defined as [B]false.

As an example of a formula not definable in C, we mention <A>ϕ, which holds in an interval $[a, b]$ if ϕ holds in a "neighbouring" interval $[b, c]$ for some c.

These formulas come from a pioneering paper by Halpern and Shoham [HS91] which studied decidability questions of ITLs. This paper sharpens their results.

Instead of the binary "chop" modality, Halpern and Shoham consider the logic HS with the three unary modalities <B>, <E> and <A>. Observe that <D>ϕ is definable in HS as <B><E>ϕ. Venema's paper [Ven90] shows that chop is not definable in HS. The logic having the chop modality as well as the three unary modalities is called CDT [Ven91].



We will also be interested in two sublogics BE and $\mathsf{D\overline{D}}$ of HS. BE only has the modalities <B> and <E> and is also a sublogic of C. $\mathsf{D\overline{D}}$ has the two modalities <D> and $< \overline{\mathtt{D}} >$, where $< \overline{\mathtt{D}} >\phi$ holds if ϕ is true in an interval *of which* the current interval is a subinterval (a kind of "mirror image" of <D>). We also define D to have the single modality <D>.

Halpern and Shoham [HS91] showed that the logic HS (and therefore CDT) is undecidable on many reasonable kinds of ordered structures (including the natural numbers and the reals), and raised the question of decidability of BE and $\mathsf{D\overline{D}}$. The decidability of the logic C has also remained open.

In this paper, we show that BE (and therefore C) is undecidable. Specifically, validity is r.e.-hard. We expect that $\mathsf{D\overline{D}}$ can also be shown undecidable by a refinement of the Halpern-Shoham argument, similar to the one below. On the other hand, the logic D should be decidable.

We do not claim a great deal of originality for our proof, which closely follows Halpern and Shoham [HS91]. They reduce the non-halting problem of a Turing machine on a blank tape to the satisfiability problem of HS. Looking at their proof, we see that most of it is carried out in the BE sublogic. Our aim is to convert the remaining part to BE.

Our proof shows that satisfiability is undecidable over the ordinal $\omega+1$ — equivalently, over the natural numbers when intervals of the form $[a,\infty)$ are allowed. This is in contrast to monadic second-order logic MSO[<], which Büchi showed to be decidable (in fact, over all countable ordinals). Our results show that a translation of formulas from the logic into MSO *à la* [Pan95,Rab98] will not work for the entire class of valuations.

The logic HS was shown to have a complete axiomatization by Venema [Ven90]. In [Ven91], Venema showed that CDT is also complete. In a forthcoming paper, we report on the axiomatization of BE and C [LR00].

Related work

Many papers provide decidability results for ITL and Duration Calculus (or DC, an extension of ITL). The reason they are able to do this is because they work with models where the valuation function is restricted from being so wild as to assign an arbitrary set of intervals for a proposition.

For instance, much of the DC literature [ZHR91,ZHS93] interprets a proposition's being true in an interval as holding *almost everywhere* [Bur82], that is, "most" subintervals of this interval must also make it true. We call models with such valuations ae-models. Zhou, Hansen and Sestoft encoded a proposition as a string of formulas and then used regular language techniques to prove decidability of DC on ae-models [ZHS93].

Pandya [Pan95] works with an interesting variation where the valuation for propositions only needs to be provided for point intervals. We call these pi-models. Pandya shows decidability of DC (and extensions of it) on pi-models [Pan95] by embedding the logic into the monadic second-order logic of order MSO[<].

Another class considered is that of *finitely varying* models, where a proposition can only change value finitely many times during any finite interval [ZHR91]. We call these fv-models. (For instance, all models over a discrete time line fall into this class.) Rabinovich shows decidability on fv-models [Rab98], again by translation into MSO[<].

Like Halpern and Shoham, we are working on models where the valuation is unrestricted, hence none of these decidability results apply.

Acknowledgements

Paritosh Pandya suggested to us that the logic C may be undecidable over dense time. Suman Roy worked on some of the ideas in this paper. I thank both of them for discussion.

2 The undecidability proof

Idea

Halpern and Shoham represent a Turing machine computation as an infinite sequence of IDs (instantaneous descriptions, which are sometimes also called configurations of the machine). In specifying this sequence, they use the <A> modality.

Each ID is a finite sequence of tape cells containing a unique tape symbol, and one of the cells has additional information representing the head position and state of the machine.

The most ingenious part of Halpern and Shoham's construction lies in using a proposition *corr* ("corresponds") which makes it possible to talk about consecutive IDs. For instance, *corr* would be true of an interval which begins with the 1936th tape cell of an ID and ends with the 1936th tape cell of the next ID. Again, this requires using an <A> modality.

Once this is done, the transition function δ of the Turing machine can be respected by examining a group of three cells in an ID and determining the value of the same three cells in the next ID.

Our aim is to do the coding within BE, i.e. to eliminate the use of the <A> modality. We achieve this by treating the entire infinite computation as being inside a dense interval $[a, b]$. Now the <D> modality can be used to talk about consecutive IDs, sequences, etc.

Details

Our claim is that the formula $computation \wedge comp\text{-}properties$ below, parameterized by a Turing machine, is satisfiable if and only if it does not halt on a blank tape. We closely follow the proof in [HS91, 8.2]. In particular, analogues of Lemmas 8.9, 8.12 and 8.14 of that paper hold for our reduction.

We assume the TM only writes the symbols 0 and 1. Let Q be its set of states. The propositions of our language include $L = \{0, 1, (q, 0), (q, 1), (q, B) \mid q \in Q\}$. There are also other propositions like *cell*, *ID* etc, which we describe below.

The formula *comp-properties* enforces properties that an interval $[a, b]$ where the proposition *computation* holds must satisfy. Assume such an $[a, b] \models$ *computation*.

$$
\begin{aligned}
\text{comp-properties} = {} & \text{computation} \supset \\
& \text{not-contains(computation)} \wedge \\
& \text{seq(ID)} \wedge \text{seq(ID)-properties} \wedge \\
& \texttt{<B>}\text{init-ID} \wedge \text{ID-properties} \wedge \\
& \text{cell-properties} \wedge \text{seq(cell)-properties} \wedge \\
& \text{corr-properties} \wedge \text{obeys-}\delta
\end{aligned}
$$

First, we have a generic specification that an interval does not contain a subinterval satisfying a particular proposition. For the formula above, this will show that no proper subinterval of $[a, b]$ will have *computation* true.

$$\text{not-contains(x)} = \neg\texttt{<B>}x \wedge \neg\texttt{<D>}x \wedge \neg\texttt{<E>}x$$

Now we encode generic sequences using the chop operator. Hence we are outside BE, but we will show how to eliminate the chops later.

$$
\begin{aligned}
\text{seq(x)-properties} = {} & \texttt{[D]}(\text{seq(x)} \supset \text{x} \frown \text{seq(x)}) \wedge \\
& \texttt{[D]}(\text{x} \supset \neg\ \text{seq(x)})
\end{aligned}
$$

comp-properties now shows that $[a, b] \models ID \frown seq(ID)$, and by repeated application, $[a, b] \models ID \frown ID \frown \ldots$ (that is, an infinite sequence of IDs). The same trick works for IDs, showing it to be an infinite sequence of cells, additionally requiring that it contains a unique cell containing a state.

$$
\begin{aligned}
&\text{ID-properties} = \texttt{[D]}(\text{ID} \supset \text{not-contains(ID)} \wedge \text{seq(cell)} \wedge \text{one-state}) \\
&\text{one-state} = \texttt{<D>}\text{state} \wedge \neg\ \texttt{<D>}(\texttt{<B>}\text{state} \wedge \texttt{<E>}\text{state})
\end{aligned}
$$

Some special IDs are abbreviated:

$$\text{init-ID} = \text{ID} \wedge \texttt{[D]}((\text{cell} \supset \text{blank}) \wedge (\text{state} \supset \text{init-state}))$$

The properties of a cell are described below. We also abbreviate some special cells.

$$
\begin{aligned}
&\text{cell-properties} = \texttt{[D]}(\text{cell} \supset \text{not-contains(cell)} \wedge \text{unique-val}) \\
&\text{unique-val} = \bigvee_{l \in L} l \wedge \bigwedge_{l \neq m} (l \supset \neg m) \\
&\text{cell}(l) = \text{cell} \wedge l \\
&\text{state} = \bigvee_{l=(q,i) \in L} \text{cell}(l)
\end{aligned}
$$

$$\text{init-state} = \bigvee_{l=(q_0,i)\in L} \text{cell}(l)$$
$$\text{blank} = \text{cell}(B) \vee \bigvee_{l=(q,B)\in L} \text{cell}(l)$$

Now we come to Halpern and Shoham's ingenious *corr* proposition. It is true of an interval if and only if it starts and ends with a cell, and these cells are corresponding cells in consecutive IDs.

$$\begin{aligned}\text{corr-properties} = \texttt{[D]}(&\text{ cell-rule} \wedge \text{ID-rule} \wedge \text{corr-starts} \wedge \text{corr-ends}\\ &\wedge (\text{corr} \supset \text{not-contains(corr)}))\end{aligned}$$
$$\text{cell-rule} = \text{corr} \supset (\texttt{<B>}\text{cell} \wedge \texttt{<E>}\text{cell})$$
$$\text{corr-starts} = (\texttt{<B>}\text{cell} \wedge \texttt{<E>}\text{corr}) \supset \texttt{<B>}\text{corr}$$
$$\text{corr-ends} = (\texttt{<E>}\text{cell} \wedge \texttt{<B>}\text{corr}) \supset \texttt{<E>}\text{corr}$$
$$\text{ID-rule} = \text{ID} \frown \text{cell} \supset \text{corr}$$

The last two formulas above are simpler than in [HS91] because we are only concerned with an infinite computation.

Finally, the transition function is respected by examining a group of three cells and determining the value of the middle state in the next ID. Here is our formulation of Halpern and Shoham:

$$\text{3-cell(x,y,z)} = \text{cell(x)} \frown \text{cell(y)} \frown \text{cell(z)}$$
$$\begin{aligned}\text{obeys-}\delta = \bigwedge_{i,j,k\in L} \texttt{[D]}(&\texttt{<B>}\text{corr} \wedge \texttt{<B>}\text{3-cell(i,j,k)} \supset\\ &\texttt{<B>}(\text{corr} \frown \text{cell}(\delta\text{(i,j,k)})))\end{aligned}$$

As stated earlier, the formula *computation*$\wedge$*comp-properties* will encode a Turing machine.

Undecidability of BE

Now we have undecidability of C, but to prove that of BE we have to replace occurrences of chop by operators definable within BE. First of all, we can rewrite our sequencing operation:

$$\begin{aligned}\text{seq(x)-properties} = \texttt{[D]}(&\text{seq(x)} \supset \texttt{<B>}\text{x} \wedge \texttt{[E]}(\text{x} \supset \text{seq(x)}) \wedge\\ &\texttt{[D]}(\text{x} \supset \neg\, \text{seq(x)})\end{aligned}$$

The 3-*cell* formula was already written by Halpern and Shoham in BE, as also a 2-*cell* formula which we will require later:

$$\begin{aligned}
\text{3-cell(x,y,z)} &= \texttt{<B>}\text{cell(x)} \wedge \texttt{<D>}\text{cell(y)} \wedge \texttt{<E>}\text{cell(z)} \\
&\qquad \wedge\ \texttt{[D]}(\text{cell} \supset \text{cell(y)})
\end{aligned}$$

$$\text{3-cell} = \bigvee_{x,y,z} \text{3-cell(x,y,z)}$$

$$\text{2-cell} = \texttt{<B>}\text{cell} \wedge \texttt{<E>}\text{cell} \wedge \texttt{[D]}\neg\ \text{cell}$$

We modify the interpretation of *corr* slightly: it holds if and only if it starts and ends with a cell, and the ending cell is the successor of the cell corresponding to the starting cell in the next ID. Now we can define the transition relation within BE. Our trick requires a modification of *ID-rule* as well:

$$\text{ID-rule} = (\texttt{<B>}\text{ID} \wedge \texttt{<E>}\text{2-cell} \wedge \texttt{[E]}\neg\ \text{3-cell}) \supset \text{corr}$$

$$\begin{aligned}
\text{obeys-}\delta = \bigwedge_{i,j,k \in L} \texttt{[D]}(\ &\texttt{<B>}\text{corr} \wedge \texttt{<B>}\text{3-cell(i,j,k)} \supset \\
&\qquad \texttt{<E>}(\text{cell}(\delta(\text{i,j,k}))))
\end{aligned}$$

Undecidability over $\omega + 1$

We have used a simpler argument than Halpern and Shoham's, representing each ID as an interval containing an infinite sequence of tape cells, and the computation as an infinite sequence of IDs. So our proof shows that satisfiability is undecidable over the ordinal ω^2.

On the other hand, Halpern and Shoham use *markers* of two kinds: cell-markers separating consecutive cells and ID-markers separating consecutive IDs. This means that the entire computation can be represented as a single infinite sequence with each cell/ID delimited by the appropriate kind of marker.

It is not difficult to add this level of detail to our proof. Notice that our initial interval $[a, b]$, which satisfies the proposition *computation*, has to have inside it the entire computation, hence the point b must be *after* the computation. Hence we obtain undecidability of BE over $\omega + 1$. The same effect may be achieved if we work over the natural numbers by allowing intervals of the form $[a, \infty)$.

References

[All83] J.F. Allen. Maintaining knowledge about temporal intervals, *Commun. ACM* 26,11 (Nov 1983) 832–843.

[Ben95] J. van Benthem. Temporal logic, in D.M. Gabbay, C. Hogger and J.A. Robinson (eds), *Handbook of logic in artificial intelligence and logic programming* IV (Oxford, 1995) 241–350.

[Bur82] J.P. Burgess. Axioms for tense logic II: time periods, *Notre Dame J.FL* 23,4 (Oct 1982) 375–383.

[Dut95] B. Dutertre. On first order interval temporal logic, Tech Report, Royal Holloway College, University of London (1995), `http://www.csl.sri.com/~bruno/publis/TR-CSD-94-3.ps.gz`. [Also in *Proc. 10th LICS* (1995).]

[HMM83] J.Y. Halpern, Z. Manna and B. Moszkowski. A hardware semantics based on temporal intervals, *Proc. ICALP*, *LNCS* 154 (1983) 278–291.

[HS91] J.Y. Halpern and Y. Shoham. A propositional modal logic of time intervals, *J. ACM* 38,4 (Oct 1991) 935–962.

[LR00] K. Lodaya and S. Roy. Lines, a while and intervals, presented at the *Workshop on Many-dimensional logical systems*, *ESSLLI '00*, Birmingham (2000).

[Pan95] P.K. Pandya. Some extensions to propositional mean-value calculus: expressiveness and decidability, *Proc. CSL*, *LNCS* 1092 (1995) 434–451.

[Rab98] A. Rabinovich. On the decidability of continuous time specification formalisms, *J. Logic Comput.* 8,5 (1998) 669–678.

[RP86] R. Rosner and A. Pnueli. A choppy logic, *Proc. 1st LICS* (1986) 306–313.

[RRM93] A.P. Ravn, H. Rischel and K.M. Hansen. Specifying and verifying requirements of real-time systems, *IEEE Trans. Softw. Engg.* 19,1 (Jan 1993) 41–55.

[Ven90] Y. Venema. Expressiveness and completeness of an interval tense logic, *Notre Dame J.FL* 31,4 (1990) 529–547.

[Ven91] Y. Venema. A modal logic for chopping intervals, *J. Logic Comput.* 1,4 (1991) 453–476.

[ZHR91] Zhou C., C.A.R. Hoare and A. Ravn. A calculus of durations, *IPL* 40,5 (Dec 1991) 269–276.

[ZHS93] Zhou C., M.R. Hansen and P. Sestoft. Decidability and undecidability results for duration calculus, *Proc. STACS*, *LNCS* 665 (1993) 58–68.