ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

A highly restricted temporal logic with a tractable decision procedure

Ian Pratt

Department of Computer Science
University of Manchester

Technical Report Series
UMCS-95-11-1

A highly restricted temporal logic with a tractable decision procedure*

Ian Pratt

Department of Computer Science
University of Manchester
Oxford Road, Manchester, UK.
ipratt@cs.man.ac.uk

29-11-95

*Refereed by: Bill Mitchell

Abstract

This paper demonstrates the tractability of a highly restricted temporal logic, into which English sentences containing a variety of temporal prepositional constructions can be translated. This paper is intended as a preliminary study for a larger project of identifying tractable temporal logics corresponding to temporal constructions in natural language. This larger project also includes current work exploring the automatic translation of English sentences involving temporal prepositions (and related adverbials) into temporal logics.

1 Background and objectives

This paper is intended as a preliminary study for the project of identifying tractable temporal logics corresponding to temporal constructions in natural language. This project relates to current work exploring the automatic translation of English sentences involving temporal prepositions (and related adverbials) into temporal logics (Pratt and Brée [1]). On this translation scheme, sentences such as

> The ferry had departed by the time John telephoned
>
> Mary will arrive sometime between Monday and Tuesday
>
> David worked on the paper every day for four months
>
> Susan will not finish the report until 2 months after Mark returns

are mapped to a modal temporal logic in which the sentential operators quantify over subintervals of the interval at which they are evaluated (see below for examples). This translation from English to temporal logic has already been implemented in a program running under LPA Prolog 4.5.[1]

In this paper we shall not discuss any details of the translation process; nor do we attempt to analyse the properties of the full logic into which that translation process maps English. Rather, we concentrate on establishing the tractability of a highly restricted subset of that logic.

2 The syntax

We have an infinite stock P of proposition letters $\{p_1, p_2, \ldots\}$ representing event-types. The formulae of the language are constructed as follows:

1. If p is a proposition letter, $\diamond p$ and $\neg \diamond p$ are formulae.
2. If p is a proposition letter and ψ is a formula, then $\bullet\!\!\rightarrow(p, \psi)$ and $\leftarrow\!\!\bullet(p, \psi)$ are formulae.

3 The semantics

Propositions $p \in P$ receive truth-values with respect to time-intervals. We allow open, closed and half-open intervals; however, we insist that, if I is an interval, $\text{end}(I) > \text{start}(I)$, where $\text{start}(I)$ and $\text{end}(I)$ are the greatest lower bounds and least upper bounds of I, respectively. This restriction is motivated by the assumption that the events described in this logic never take zero time.

Formally, we take a structure to be a set $M \subseteq P \times \Xi$ where Ξ is the set of intervals of the real line and P is the set of proposition letters. Intuitively, the pair $\langle p, J \rangle \in M$ just in case an event of the type represented by p occurs over the interval J, starting at the beginning of J and finishing at the end of J.

A structure M assigns truth-values to formulae with respect to a closed interval I, according to the following rules:

[1] Available upon request from the author.

S1: $M \models_I \diamond(p)$ iff $\langle p, J\rangle \in M$ for some subinterval $J \subseteq I$

S2: $M \models_I \neg\diamond(p)$ iff $\langle p, J\rangle \notin M$ for all subintervals $J \subseteq I$

S3: $M \models_I \bullet\!\!\rightarrow(p, \psi)$ iff $\langle p, J\rangle \in M$ for some subinterval $J \subseteq I$, such that $\langle p, K\rangle \notin M$ for all subintervals $K \subseteq I$ beginning earlier than J, and $M \models_G \psi$, where G is the closed interval [start(I), start(J)].

S4: $M \models_I \leftarrow\!\!\bullet(p, \psi)$ iff $\langle p, J\rangle \in M$ for some subinterval $J \subseteq I$, such that $\langle p, K\rangle \notin M$ for all subintervals $K \subseteq I$ ending later than J, and $M \models_G \psi$, where G is the closed interval [end(J), end(I)].

Clearly, this logic is very limited. However, it allows us to express many temporal preposition-constructs in English:

The ferry had departed by the time John telephoned
$\bullet\!\!\rightarrow$([John telephone], $\diamond$[the ferry depart])

Mary will arrive sometime between Monday and Tuesday
$\bullet\!\!\rightarrow$([Tuesday], $\leftarrow\!\!\bullet$[Monday], $\diamond$[Mary arrive])

Note that [Monday] and [Tuesday] are taken to be events which occur over intervals here. However, the logic is unable to express the uniqueness of such intervals within a period such as a week.

That the symbol $\neg$ cannot attach to whole formulae corresponds approximately to the way *not* interacts with temporal preposition phrases in English. Thus, for example, the sentence

Mary will not leave until John telephones

presupposes that John will telephone, and states that Mary will not leave until (the first time) an event of that type occurs. Taking the liberty of merging the presupposition and truth-conditions, we might represent the meaning of this sentence by means of a formula with the negation on the inside:

$\bullet\!\!\rightarrow$([John telephone], $\neg\diamond$[Mary leave]).

4 The decision procedure

We present a procedure for taking a finite set of sentences Φ in the above language and determining whether a model can be constructed for them.

The basic data-structures in this decision procedure are so-called left and right proto-models. We say that a set Φ of formulae is *left-handed* if its members are of the forms $\bullet\!\!\rightarrow(p_1, \psi)$, $\diamond p$ and $\neg\diamond p$; and we say that Φ is *right-handed* if its members are of the forms $\leftarrow\!\!\bullet(p_1, \psi)$, $\diamond p$ and $\neg\diamond p$. A *left* proto-model is a partial model of a left-handed set of formulae, and a *right* proto-model is a partial model of a right-handed set of formulae. Proto-models are *partial* in that the intervals with respect to which proposition letters are made true are only partially ordered. Left proto-models in general contain right proto-models as constituents; right proto-models in general contain left proto-models as constituents. Since left and right proto-models are mirror images, we describe only left proto-models below.

Think of a left proto-model L as attaching to a closed interval I, within which all of the events mentioned in L have to occur. L has four components with the following intuitive interpretations:

- required(L): a list of propositions which must be instantiated at some subinterval of I
- forbidden(L): a list of propositions which may not be instantiated at any subinterval of I
- events(L): a list of events, where an event is defined below
- order(L): a partial ordering on the set of end points $\{\text{start}(E) | E \in \text{events(L)}\} \cup \{\text{end}(E) | E \in \text{events(L)}\}$.

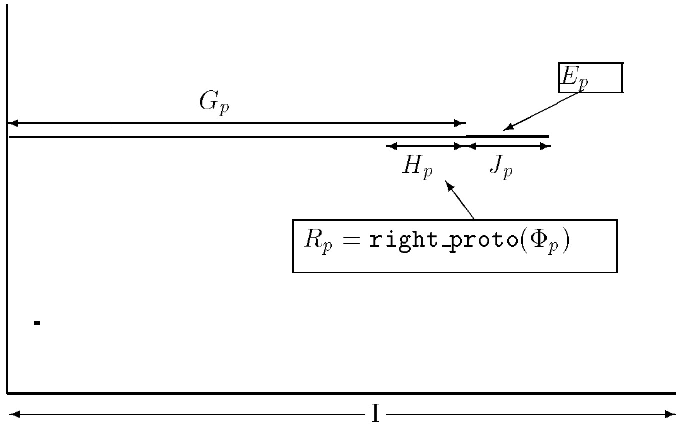

Figure 1: Schematic representation of a left proto-model

Think of an event E in a left proto-model L as an occasion within the basic interval I of L at which some atomic proposition p is true. Such an event will typically be generated by a formula of the form $\bullet\!\!\rightarrow(p, \psi)$ or of the form $\diamond p$ and has the following components:

- proposition(E): the proposition which is true in the event
- right_proto(E): a right proto-model encoding events that must occur between the start of the basic interval I and the start of the event E.

For every proposition letter p, a proto-model L contains at most one event whose proposition is p; we denote this event by E_p. Fig. 1 depicts an event E_p in a left proto-model L. We take the event E_p to occur over an interval J_p, and the right proto-model R_p belonging to E_p to attach to a closed interval H_p just prior to J_p. G_p denotes the closed interval from the start of I to the start of J_p.

The function **left_proto_model**(Φ) takes a left-handed set of formulae Φ and builds a left proto-model. The function uses the following notation.
If Φ is left-handed we write:

$$\begin{aligned}
&\text{blob_arrows}(\Phi) = \{p | \bullet\!\!\rightarrow(r_1, \bullet\!\!\rightarrow(r_2, \ldots \bullet\!\!\rightarrow(p, \psi)\ldots)) \in \Phi \text{ some } r_1,\, r_2,\, \ldots,\, \psi\} \\
&\Phi_p = \{\psi | \bullet\!\!\rightarrow(r_1, \bullet\!\!\rightarrow(r_2, \ldots \bullet\!\!\rightarrow(p, \psi)\ldots)) \in \Phi \text{ some } r_1,\, r_2,\, \ldots, \text{ s.t. } \psi \text{ not of the form } \bullet\!\!\rightarrow(q, \pi)\}\ .
\end{aligned}$$

If Φ is right-handed we write:

$$\begin{aligned}
&\text{blob_arrows}(\Phi) = \{p | \leftarrow\!\!\bullet(r_1, \leftarrow\!\!\bullet(r_2, \ldots \leftarrow\!\!\bullet(p, \psi)\ldots)) \in \Phi \text{ some } r_1,\, r_2,\, \ldots,\, \psi\} \\
&\Phi_p = \{\psi | \leftarrow\!\!\bullet(r_1, \leftarrow\!\!\bullet(r_2, \ldots \leftarrow\!\!\bullet(p, \psi)\ldots)) \in \Phi \text{ some } r_1,\, r_2,\, \ldots, \text{ s.t. } \psi \text{ not of the form } \leftarrow\!\!\bullet(q, \pi)\}\ .
\end{aligned}$$

Finally, for any Φ, we write:

$$\begin{aligned}
\text{diamonds}(\Phi) &= \{p | \diamond p \in \Phi\} \\
\text{neg_diamonds}(\Phi) &= \{p | \neg \diamond p \in \Phi\}
\end{aligned}$$

The function is defined as follows.

function `left_proto_model`(Φ)
create the new proto-model L with:
events(L) = banned(L) = required(L) = order(L) = $\emptyset$
A 1 **for** r **in** blob_arrows(Φ) **do**
2 create a new right proto-model R_r = `right_proto_model`(Φ_r)
3 **set** banned(R_r) = banned(R_r) $\cup \{r\}$
4 create the new event E_r with:
5 proposition(E_r) = r
6 right_proto(E_r) = R_r
7 **set** events(L) = events(L) $\cup \{E_r\}$
8 **set** required(L) = required(L) $\cup$ required(R_r) $\cup \{r\}$
9 **endfor**
B 1 **for** r **in** diamonds(Φ) **do**
2 **if** there is no $E \in$ events(L) with proposition(E) = r **then**
3 create a new right proto-model R_r with:
4 events(L) = banned(L) = required(L) = order(L) = $\emptyset$
5 create the new event E_r with:
6 proposition(E_r) = r
7 right_proto(E_r) = R_r
8 **set** events(L) = events(L) $\cup \{E_r\}$
9 **set** required(L) = required(L) $\cup \{r\}$
10 **endif**
11 **endfor**
C 1 **for** r **in** neg_diamonds(Φ) **do**
2 **set** banned(L) = banned(L) $\cup \{r\}$
3 **endfor**
D 1 **for** each formula of the form $\bullet\!\!\rightarrow(r_1, \bullet\!\!\rightarrow(r_2, \ldots \bullet\!\!\rightarrow(r_n, \psi) \ldots))$ **in** Φ **do**
2 **for** $i = 1 \ldots n$ **do**
3 **set** order(L) = order(L) $\cup \{\text{start}(E_{r_i}) \geq \text{end}(E_{r_{i+1}})\}$
4 **endfor**
5 **endfor**
E 1 **for** r **in** blob_arrows(Φ) $\cup$ diamonds(Φ) **do**
2 **for** s **in** blob_arrows(Φ) $\setminus \{r\}$ **do**
3 **if** required(R_r) $\cap$ banned(R_s) $\neq \emptyset$ **then**
4 **set** order(L) = order(L) $\cup \{\text{start}(E_r) > \text{start}(E_s)\}$
5 **endif**
6 **if** $r \in$ banned(R_s) **then**
7 **set** order(L) = order(L) $\cup \{\text{end}(E_r) > \text{start}(E_s)\}$
8 **endif**
9 **endfor**
10 **endfor**
return L
end `left_proto_model`

The function `right_proto_model`(Φ) is identical, except that the inequalities are reversed, the pairs of words `start` and `end` and `left` and `right` are switched, and line D1 contains the mirror-image schema $\leftarrow\!\!\bullet(r_1, \leftarrow\!\!\bullet(r_2, \ldots \leftarrow\!\!\bullet(r_n, \psi) \ldots))$.

The following definition identifies those proto-models that can be extended to models.

Definition 1 *Let P be a (left or right) proto-model. We say that P is* well-behaved *if the following conditions hold.*

1. $\mathrm{required}(P) \cap \mathrm{banned}(P) = \emptyset$
2. $\mathrm{order}(P)$ *is consistent (contains no cycles)*
3. *All the sub-proto-models belonging to the events of P are well-behaved*

We can determine the satisfiability of a left-handed set of formulae Φ as follows. First, we construct `left_proto_model`(Φ). If the result is a well-behaved proto-model, then Φ has a model; otherwise, Φ has no model. That this procedure really does test for consistency is proved in section 6. Similarly, to determine the satisfiability of a right-handed set of formulae Φ, we construct `right_proto_model`(Φ) and determine whether the result is a well-behaved proto-model. If Φ is neither left- nor right-handed, we split into sets which are. Specifically, we define:

$$\begin{aligned}
\mathrm{exists}(\Phi) &= \{\phi \in \Phi | \phi \text{ is of the form } \diamond p\} \\
\mathrm{neg_exists}(\Phi) &= \{\phi \in \Phi | \phi \text{ is of the form } \neg \diamond p\} \\
\mathrm{lefts}(\Phi) &= \{\phi \in \Phi | \phi \text{ is of the form } \bullet\!\!\rightarrow(p, \psi)\} \\
\mathrm{rights}(\Phi) &= \{\phi \in \Phi | \phi \text{ is of the form } \leftarrow\!\!\bullet(p, \psi)\}
\end{aligned}$$

Then if we let

$$\begin{aligned}
\Phi_{\mathrm{left}} &= \mathrm{exists}(\Phi) \cup \mathrm{neg_exists}(\Phi) \cup \mathrm{lefts}(\Phi) \\
\Phi_{\mathrm{right}} &= \mathrm{neg_exists}(\Phi) \cup \mathrm{rights}(\Phi)
\end{aligned}$$

It is easy to show that Φ is satisfiable if and only if both Φ_{left} and Φ_{right} are, so we can test Φ_{left} and Φ_{right} separately.

To summarize, the function for determining the satisfiability of a set of formulae Φ is as follows:

function `consis_check`(Φ)
  Split Φ into Φ_{left} and Φ_{right}
  Compute $L =$ `left_proto_model`(Φ_{left}), $R =$ `right_proto_model`(Φ_{right})
  Determine whether L and R are well-behaved
  if yes in both cases **then**
    return "Φ has a model"
  else
    return "Φ has no model"
  endif
end `consis_check`

5 Complexity

The following result establishes the tractability of determining satisfiability in the logic.

Theorem 1 (Tractability) *The time-complexity of the procedure described in section 4 is cubic in the number of connectives in Φ.*

Proof: Consider first the construction of the left proto-model as described in `left_proto_model`(Φ). Each pass through the loops A, B, C and D 'uses up' at least one connective in Φ. Loop E has cubic complexity, since, for each pair of proto-models, the required- and banned-lists have to be searched for a common entry. (We assume constant-time table look-up to determine whether $t \in \text{required}(R_r)$ and $t \in \text{banned}(R_s)$ for any r, s, t.)
The total number of events in the proto-model generated from Φ (including all its sub-proto-models) is bounded by the size of Φ. Moreover, testing the well-behavedness of a proto-model involves only checking for the disjointness of required(L) and banned(L), and for the absence of cycles in order(L). This step is clearly quadratic in the size of events(L). $\square$

6 Correctness and completeness

The following two theorems establish that the procedure of section 4 actually works. We state here only the left-hand versions. The right-hand analogues have corresponding proofs.

Theorem 2 (Correctness) *Let Φ be a left-handed set of formulae. If Φ has a model, then* `left_proto_model`(Φ) *is well-behaved.*

Proof: Suppose $M \models_I \Phi$ and $L =$ `left_proto_model`(Φ). We show by induction on the size of L that L is well-behaved. If L is empty, then L is trivially well-behaved.

First, we map the events in events(L) to intervals in M at which the corresponding proposition letters are made true. The only events in events(L) will be those created by lines A4 and B5, corresponding to the proposition letters in blob_arrows(Φ) and in diamonds(Φ), respectively.
For each $p \in \text{diamonds}(\Phi)$, let J_p be any interval such that $\langle p, J_p \rangle \in M$; the semantic rule S1 guarantees the existence of J_p. Map the event E_p in L to the interval J_p.
For each $p \in \text{blob\_arrows}(\Phi)$, let J_p be a first interval (ordered by starting point) such that $\langle p, J_p \rangle \in M$; the semantic rule S3 guarantees the existence of J_p. Map the event E_p in L to the interval J_p.
Let G_p be the interval $[\text{start}(I), \text{start}(J_p)]$. Note that, by the semantic rule S3, $M \models_{G_p} \Phi_p$.
To show that L is well-behaved, it suffices to show that

1. R_p is well-behaved for every $E_p \in \text{events}(L)$
2. $\text{required}(L) \cap \text{banned}(L) = \emptyset$
3. order(L) contains no cycles.

These conditions are then established as follows

1. If $E_p \in \text{events}(L)$ then $p \in \text{blob\_arrows}(\Phi)$ or $p \in \text{diamonds}(\Phi)$. If $p \in \text{blob\_arrows}(\Phi)$, then, as we have observed, $M \models_{G_p} \Phi_p$. But since $R_p = \texttt{right\_proto\_model}(\Phi_p)$, then, by the inductive hypothesis, R_p is well-behaved. If $p \notin \text{blob\_arrows}(\Phi)$ but $p \in \text{diamonds}(\Phi)$, then R_p is an empty proto-model and so is trivially well-behaved.

2. Let t be a proposition letter. By lemma 1 below, if $t \in \text{required}(L)$, then, since $M \models_I \Phi$, we have $M \models_I \diamond t$. Moreover, the only way in which t could have been added to banned(L) is by line C2, and this line is executed only if $\neg \diamond t \in \Phi$, contradicting $M \models_I \Phi$ and $M \models_I \diamond t$. So $\text{required}(P) \cap \text{banned}(P) = \emptyset$.

3. We need only check that the order imposed in lines D3, E4 and E7 is respected in M under the mapping given above. The absence of cycles then follows immediately. We illustrate with the case of line E4.

   Line E4 will add $\text{start}(E_r) > \text{start}(E_s)$ to order(L) only if $\text{required}(R_r) \cap \text{banned}(R_s) \neq \emptyset$.

   Suppose, then that $t \in \text{required}(R_r) \cap \text{banned}(R_s)$. $M \models_{G_r} \Phi_r$, by construction of G_r, and $R_r = \texttt{right\_proto\_model}(\Phi_r)$, so, by lemma 1, $M \models_{G_r} \diamond t$ and so, by semantic rule S3, $M \models_I \bullet\!\!\rightarrow(r, \diamond t)$. Moreover, t could only have been inserted in banned(R_s) by line A3 (in which case $t = s$) or by line C2, in a recursive call (in which case $\neg \diamond t \in \Phi_s$). In the former case, we have $M \models_I \bullet\!\!\rightarrow(r, \diamond s)$ so that $\text{start}(J_r) > \text{start}(J_s)$. In the latter case, we have $\bullet\!\!\rightarrow(s, \neg \diamond t) \in \Phi$ which, given that $M \models_I \bullet\!\!\rightarrow(r, \diamond t)$, again forces $\text{start}(J_r) > \text{start}(J_s)$, by the semantic rules S1, S2 and S3. Hence the order $\text{start}(E_r) > \text{start}(E_s)$ is respected by the order $\text{start}(J_r) > \text{start}(J_s)$ on the corresponding intervals.

   The orderings added by lines D3 and E7 can be similarly shown to correspond to orderings on the intervals in M. □

Theorem 3 (Completeness) *Let Φ be a left-handed set of formulae. If* $\texttt{left\_proto\_model}(\Phi)$ *is well-behaved, then Φ has a model.*

Proof: Let L be a well-behaved left proto-model. We define the notion of an *expansion*, M_L, of L into a closed interval I. Intuitively, M_L is a structure which assigns propositions to intervals according to the events in L, in such a way that the ordering imposed by L is respected.
The definition proceeds recursively. If L is an empty proto-model, the only expansion of L into a closed interval I is the empty structure $\emptyset$.
Otherwise, let M be any set of pairs

$$M = \{\langle p, J_p\rangle | E_p \in \text{events}(L),\ J_p \subseteq I \text{ an interval}\} \tag{1}$$

such that the orderings on the start- and end-points of the J_p's respect the orderings on the corresponding E_p's imposed in order(L). If L is well-behaved, then order(L) contains no cycles, so that there exist M. Since order(L) can never impose the conditions $\text{start}(J_p) = \text{start}(J_q)$ or $\text{start}(J_p) = \text{start}(I)$ for $p \neq q$, we may assume that $\text{start}(J_p) \neq \text{start}(I)$ and $\text{start}(J_p) \neq \text{start}(J_q)$, for all $p \neq q$.
As before, for each $E_p \in \text{events}(L)$,we define $G_p = [\text{start}(I), \text{start}(J_p)]$; in addition, let H_p be a small closed interval such that

$$\begin{aligned} \text{end}(H_p) &= \text{start}(J_p) \text{ and} \\ \text{start}(H_p) &> \text{start}(H_q) \text{ for all } q \text{ such that } \text{start}(J_p) > \text{start}(J_q)\ . \end{aligned} \tag{2}$$

The idea is that H_p is a small interval wedged in between the start of J_p and the start of anything that starts before J_p (see figure 1).
For each $E_p \in \text{events}(L)$, let R_p be the right proto-model belonging to E_p, and let M_{R_p} be an expansion of R_p into H_p. Then we say that

$$M_L = M \cup \bigcup_{E_p \in \text{events}(L)} M_{R_p} \tag{3}$$

is an expansion of L into I.
To prove theorem 3, we establish that if L is the well-behaved left proto-model generated from Φ, and M_L is an expansion of L into I, then, for all $\phi \in \Phi$,

$$M_L \models_I \phi \ . \tag{4}$$

The proof proceeds by induction on the structure of ϕ. Let ϕ be any formula in Φ. We must show that $M_L \models_I \phi$. There are four cases:

Case 1: ϕ is $\diamond p$.

The loop in B1–B11 will ensure that L contains an event E_p. The construction of M_L in (1) and (3) will ensure that there exists a $J_p \subseteq I$ such that $\langle p, J_p \rangle \in M_L$. It follows that $M_L \models_I \phi$.

Case 2: ϕ is $\neg \diamond p$. The loop in C1–C3 will ensure that $p \in \text{banned}(L)$. Now, by lemma 2, if there exists a $J_p \subseteq I$ such that $\langle p, J_p \rangle \in M_L$, then $p \in \text{required}(L)$. Since, by the well-behavedness of L, $\text{required}(L) \cap \text{banned}(L) = \emptyset$, there is no $J_p \subseteq I$ such that $\langle p, J_p \rangle \in M_L$. It follows that $M_L \models_I \phi$.

Case 3: ϕ is $\bullet\!\!\rightarrow(p_1, \bullet\!\!\rightarrow(p_2, \ldots \bullet\!\!\rightarrow(p_n, \psi) \ldots))$ where ψ is not of the form $\bullet\!\!\rightarrow(q, \pi)$.

We show

a) For $i = 1, \ldots, n$, J_{p_i} is the first (ordered by starting point) sub-interval $J \subseteq I$ such that $\langle p_i, J \rangle \in M_L$

b) For $i = 1, \ldots, n-1$, $J_{p_{i+1}} \subseteq G_{p_i}$

c) $M_L \models_{G_{p_n}} \psi$

Regarding a), the loop in A1–A9 ensures that, for each i, L contains an event E_{p_i}. The construction of M_L in (1) and (3) will ensure that there exists a $J_{p_i} \subseteq I$ such that $\langle p_i, J_{p_i} \rangle \in M_L$. We need to show that, if $\langle p_i, J \rangle \in M_L$, then $\text{start}(J) \geq \text{start}(J_{p_i})$.

Suppose, then, that $\langle p_i, J \rangle \in M_L$.

By lemma 2, $p_i \in \text{required}(L)$.

Since there is only one event $E \in \text{events}(L)$ with $\text{proposition}(E) = p_i$,

by the construction of M_L in (1) and (3), $\langle p_i, J \rangle \in M_{R_q}$ for some $E_q \in \text{events}(L)$. In that case, $J \subseteq H_q$, and $p_i \in \text{required}(R_q)$ by lemma 2. Now, line A3 ensures that $p_i \in \text{banned}(R_{p_i})$, whence lines E3–E5 ensure that $\text{order}(L)$ contains $\text{start}(E_q) > \text{start}(E_{p_i})$. Since M must respect this ordering, we have $\text{start}(J_q) > \text{start}(J_{p_i})$. But then $\text{start}(H_q) > \text{start}(J_{p_i})$, by construction of H_q in (2), and so $\text{start}(J) > \text{start}(J_{p_i})$, as required.

Regarding b), the loop in D1–D5 ensures that, for each $i = 1, \ldots, n-1$, $\text{start}(J_{p_i}) \leq \text{end}(J_{p_{i+1}})$. Hence, $J_{p_{i+1}} \subseteq G_{p_i}$.

Regarding c), if L is well-behaved, then so is R_{p_n}. Since $M_{R_{p_n}}$ in equation (3) is an expansion of R_{p_n} into H_{p_n}, and $\psi \in \Phi_{p_n}$, the inductive hypothesis yields:

$$M_{R_{p_n}} \models_{H_{p_n}} \psi \ . \tag{5}$$

In fact, we can strengthen (5) to

$$M_L \models_{H_{p_n}} \psi \ , \tag{6}$$

since, if $\langle r, K \rangle \in M_L \setminus M_{R_{p_n}}$ then $K \not\subseteq H_{p_n}$. (In other words, the additional facts in M_L cannot affect what is true in H_{p_n}.) To see this latter fact, it is sufficient to note that $H_p \cap H_q = \emptyset$ and $J_p \not\subseteq H_q$ for all distinct E_p, $E_q \in \text{events}(L)$.

Finally, we strengthen (6) to

$$M_L \models_{G_{p_n}} \psi \ . \tag{7}$$

If ψ is of the form $\diamond p$ or $\leftarrow\!\!\bullet(p, \pi)$, then (6) immediately implies (7) by the semantic rules S1 and S4, respectively. If, on the other hand, ψ is of the form $\neg \diamond p$, then, when R_{p_n} is built from Φ_{p_n}, the loop in lines C1–C3 will force $p \in$ banned(R_{p_n}). Suppose, then that, for some $J \subseteq G_{p_n}$, $\langle p, J\rangle \in M_L$. By (3), and lemma 2, we have some $E_{p_i} \in$ events(L) with either (i) $p = p_i$ or (ii) $p \in$ required(R_{p_i}). If (i), then the condition in line E7 will be satisfied, and we will have end$(J) >$ start(J_{p_n}) (by setting $r = p$ and $s = p_n$). Hence $J \not\subseteq G_{p_n}$. If (ii), then the condition in line E3 will be satisfied, and we will have start$(J_i) >$ start(J_{p_n}) (by setting $r = p_i$ and $s = p_n$). Hence, start$(H_{p_i}) >$ start(J_{p_n}) and so start$(J) >$ start(J_{p_n}), whence $J \not\subseteq G_{p_n}$. Thus, there is no $J \subseteq G_{p_n}$ with $\langle p, J\rangle \in M_L$, and (7) is established.

Thus we have established a), b) and c) and hence that $M_L \models_I \phi$.

This completes the examination of all the possible cases for ϕ, so that we have established (4). Hence Φ has a model. □

Corollary 1 (Finite models) *If a set of formulae Φ has a model, then it has a finite model.*

Proof: If Φ is left-handed, the result follows from theorems 2 and 3, noting that the model M_L in the proof of theorem 3 is finite. The extension to the general case is straightforward. □

Finally, we present the proof of the lemmas used above

Lemma 1 *Let $L =$ `left_proto_model`(Φ), and let t be a proposition letter. If $t \in$ required(L) and $M \models_I \Phi$, then $M \models_I \diamond t$.*
Let $R =$ `right_proto_model`(Φ), and let t be a proposition letter. If $t \in$ required(R) and $M \models_I \Phi$, then $M \models_I \diamond t$.

We give the proof for the left-hand case: the right-hand case is similar.
Proof: The proof proceeds by induction on the size of L. The case $L = \emptyset$ is trivial. A proposition letter t can be inserted into required(L) only by lines A8 and B9. Therefore, either (i) Φ must contain the formula $\diamond t$ or (ii) Φ contains a formula of the form $\bullet\!\!\rightarrow(r_1, \bullet\!\!\rightarrow(r_2, \ldots \bullet\!\!\rightarrow(t, \psi)\ldots))$ or (iii) Φ contains a formula of the form $\bullet\!\!\rightarrow(r_1, \bullet\!\!\rightarrow(r_2, \ldots \bullet\!\!\rightarrow(q, \psi)\ldots))$ with $p \in$ required(Φ_q). In cases (i) and (ii), we have immediately that, if $M \models_I \Phi$, then $M \models_I \diamond t$. In case (iii), if $M \models_I \Phi$, then $M \models_{G_q} \Phi_q$. But R_q is smaller than L, so that, by the inductive hypothesis, $M \models_{G_q} \diamond t$. Since $G_q \subseteq I$, $M \models_I \diamond t$ by semantic rule S1. □

Lemma 2 *Let L be a proto-model constructed by the above algorithm, and let M_L be an expansion of L into a closed interval I. If there exists an interval $J \subseteq I$ and a proposition letter p such that $\langle p, J\rangle \in M_L$, then $p \in$ required(L).*
(Similarly for right proto-models.)

We give the proof for the left-hand case: the right-hand case is similar.
Proof: The proof proceeds by induction on the size of L. The case $L = \emptyset$ is trivial. By the construction of M_L in (1) and (3), $\langle p, J\rangle \in M_L$ implies either (i) there is an $E_p \in$ events(L) or (ii) $\langle p, J\rangle \in M_{R_q}$ for some $E_q \in$ events(L). If (i), then, since events(L) is modified only by lines A7 and B8, the lines A8 and A9 immediately following will ensure that $p \in$ required(L). If (ii), then, since R_q is a proto-model constructed by the above algorithm, M_{R_q} is an expansion of R_q into a closed interval, and R_q is smaller than L, by the inductive hypothesis, $p \in$ required(R_q). But line A8 will ensure that required$(R_q) \subseteq$ required(L). □

7 Conclusions

This paper demonstrates the tractability of a highly restricted temporal logic, into which English sentences containing a variety of temporal prepositional constructions can be translated.

References

[1] Pratt, Ian and D.S. Brée: "An approach to the Semantics of some English Temporal Constructions", *Proceedings, Seventeenth Annual Conference of the Cognitive Science Society* Mahwah, NJ: Lawrence Erlbaum, pp. 118–123.