

ADVANCED TENSE LOGIC

1 INTRODUCTION

In this chapter we consider the tense (or temporal) logic with until and since connectives over general linear time. We will call this logic *US/LT*. This logic is an extension of Prior's original temporal logic of *F* and *P* over linear time [Prior, 1957], via the introduction of the more expressive connectives of Kamp's *U* for "until" and *S* for "since" [Kamp, 1968b]. *U* closely mimics the natural language construct "until" with $U(A, B)$ holding when *A* is constantly true from now up until a future time at which *B* holds. *S* is similar with respect to the past. We will see that *U* and *S* do indeed extend the expressiveness of the temporal language.

In the chapter we will also be looking at other related temporal logics. The logics differ from each other in two respects. Logics may differ in the kinds of structures which they are used to describe. Structures vary in terms of their underlying model of time (or frame): this can be like the natural numbers, or like the rationals or like the reals or some other linear order or some non-linear branching or multi-dimensional shape. Logics are defined with respect to a class of structures. Considering a logic defined by the class of all linear structures is a good base from which to begin our exploration. Temporal logics also vary in their language. For various purposes, until and since may be not expressive enough. For example, if we want to be able to reason about alternative avenues of development then we may want to allow branches in the flow of time and, in order to represent directly the fact of alternative possibilities, we may need to add appropriate branching connectives. Equally, until and since may be too strong: for simple reasoning about the forward development of a mechanical system, using since may not only be unnecessary, but may require additional axioms and complexity of a decision procedure.

In this chapter we will not be looking at temporal logics based on branching. See the handbook chapter by Thomason for these matters. We will also avoid consideration of temporal logics incorporating quantification. Instead, see the handbook chapter by Garson for a discussion of predicate temporal and modal logics and see the reference [Gabbay *et al.*, 1994] for a discussion of temporal logics incorporating quantification over propositional atoms.

So we will begin with a tour of the many interesting results concerning *US/LT* including axiom systems, related logics, decidability and complexity. In section 3 we sketch a proof of the expressive completeness of the logic.

Then, in section 4 we investigate combinations of logics with a temporal element. In section 5, we develop the proof theory for temporal logic within the framework of labelled deductive systems. In section 6, we show how temporal reasoning can be handled within logic programming. In section 7, we survey the much studied temporal logic of the natural numbers and consider the powerful automata technique for reasoning about it. Finally, in section 8, we consider the possibility of treating temporal logic in an imperative way.

2 U, S LOGIC OVER GENERAL LINEAR TIME

Here we have a close look at the US logic over arbitrary linear orders.

2.1 *The logic*

Frames for our logic are linear. Thus we have a non-empty set T and a binary relation $< \subseteq T \times T$ which is:

1. *irreflexive*, i.e. $\forall t \in T$, we do not have $t < t$;
2. *total*, i.e. $\forall s, t \in T$, either $s < t$, $s = t$ or $t < s$;
3. *transitive*, i.e. $\forall s, t, u \in T$, if $s < t$ and $t < u$ then $s < u$.

The underlying model of time for a temporal logic is captured by the frame $(T, <)$.

Any use of a temporal logic will involve something happening over time. The simplest method of trying to capture this formally is to use a propositional temporal logic. So we fix a countable set $\mathcal{L}$ of atoms. The truth of a particular atom will vary in time. For example, points of time (i.e. $t \in T$) may correspond to days and the truth of the atom r on a particular day may correspond to the event of rain on that day.

A structure is a particular history of the truth of all the atoms over the full extent of time. Structures $(T, <, h)$ are linear so we have a linear frame $(T, <)$ and we have a valuation h for the atoms, i.e. for each atom $p \in \mathcal{L}$, $h(p) \subseteq T$. The set $h(p)$ is the set of all time points at which p is true.

The language $L(U, S)$ is generated by the 2-place connectives U, S along with classical $\neg$ and $\wedge$. That is, we define the set of formulas recursively to contain the atoms and $\top$ (i.e. truth) and for formulas A and B we include $\neg A$, $A \wedge B$, $U(A, B)$ and $S(A, B)$. We read $U(A, B)$ as “until A, B ” corresponding to B being true until A is. Similarly S is read as “since”.

Formulas are evaluated at points in structures. We write $\mathcal{T}, x \models A$ when A is true at the point $x \in T$. This is defined recursively as follows. Suppose that we have defined the truth of formulas A and B at all points of $\mathcal{T}$. Then for all points x :

$\mathcal{T}, x \models p$	iff	$x \in h(p)$, for p atomic;
$\mathcal{T}, x \models \top$;		
$\mathcal{T}, x \models \neg A$	iff	$\mathcal{T}, x \not\models A$;
$\mathcal{T}, x \models A \wedge B$	iff	both $\mathcal{T}, x \models A$ and $\mathcal{T}, x \models B$;
$\mathcal{T}, x \models U(A, B)$	iff	there is a point $y > x$ in T such that $\mathcal{T}, y \models A$ and for all $z \in T$ such that $x < z < y$ we have $\mathcal{T}, z \models B$;
$\mathcal{T}, x \models S(A, B)$	iff	there is a point $y < x$ in T such that $\mathcal{T}, y \models A$ and for all $z \in T$ such that $y < z < x$ we have $\mathcal{T}, z \models B$;

Often definitions and results involving S can be given by simply exchanging U and S and swapping $<$ and $>$. In that situation we just mention that a *mirror image* case exists and do not go into details.

There are many abbreviations that are commonly used in the language. As well as the classical $\perp$ (i.e. $\neg\top$ for falsity), $\vee$, $\rightarrow$ and $\leftrightarrow$, we have the following temporal abbreviations:

FA	$= U(A, \top)$	A will happen (sometime);
GA	$= \neg F\neg A$	A will always hold;
PA	$= S(A, \top)$	A was true (sometime);
HA	$= \neg P\neg A$	A was always true;
$K^+(A)$	$= \neg U(\top, \neg A)$	A will be true arbitrarily soon;
$K^-(A)$	$= \neg S(\top, \neg A)$	A was true arbitrarily recently.

Notice that Prior's original connectives F and P appear as abbreviations in this logic. The reader should check that their original semantics (see [Burgess, 2001]) are not compromised.

A formula ϕ is *satisfiable* if it has a model: i.e. there is a structure $\mathcal{T} = (T, <, h)$ and $x \in T$ such that $\mathcal{T}, x \models \phi$. A formula is *valid* iff it is true at all points of all structures. We write $\models A$ iff A is a validity. Of course, a formula is valid iff its negation is not satisfiable.

We can also define (semantic) consequence in the logic. Suppose that Γ is a set of formulas and A a formula. We say that A is a consequence of Γ and write $\Gamma \models A$ iff whenever we have $\mathcal{T}, t \models C$ for all $C \in \Gamma$, for some point t from some structure $\mathcal{T}$, then we also have $\mathcal{T}, t \models A$.

First-Order Monadic Logic of Order

For many purposes such as assessing the expressiveness of temporal languages or establishing their decidability, it is useful to be able to move from the internal tensed view of the world to an external untensed view. In doing

so we can also make use of logics with more familiar syntax. In the case of our linear temporal logics we find it convenient to move to the first-order monadic logic of linear order which is a sub-logic of the full second-order monadic logic of linear order.

The language of the full second-order monadic logic of linear order has formulas built from $<$, $=$, quantification over individual variable symbols and quantification over monadic (i.e. 1-ary) predicate symbols. To be more formal, suppose that $X = \{x_0, x_1, \dots\}$ is our set of individual variable symbols and $Q = \{P_0, P_1, \dots\}$ is our set of monadic predicates. The formulas of the language are $x_i < x_j$, $x_i = x_j$, $P_i(x_j)$, $\neg\alpha$, $\alpha \wedge \beta$, $\exists x_i\alpha$, and $\exists P_j\alpha$ for any $i, j < \omega$ and any formula α . We use the usual abbreviations $x_i > x_j$, $x_i \leq x_j$, $x_i < x_j < x_k$, $\forall x_i\alpha$ and $\forall P_i\alpha$ etc.

As usual we define the concept of a free individual variable symbol in a formula. We similarly define the set of free monadic variables of a formula. Write $\phi(x_1, \dots, x_m, P_1, \dots, P_n)$ to indicate that all the free variables (of both sorts) in the formula ϕ are contained in the lists $x_1, \dots, x_m$ and $P_1, \dots, P_n$.

The language is used to describe linear orders. Suppose that $(T, <)$ is a linear order. As individual variable symbols we will often use t, s, r, u etc, instead of $x_1, x_2, \dots$.

An individual variable assignment V is a mapping from X into T . A predicate variable assignment W is a mapping from Q into $\wp(T)$ (the set of subsets of T). For an individual variable assignment V , an individual variable symbol $x \in X$ and an element $t \in T$, we define the individual variable assignment $V[x \mapsto t]$ by:

$$V[x \mapsto t](y) = \begin{cases} V(y) & y \neq x \\ t & y = x. \end{cases}$$

Similarly for predicate variable assignments and subsets of T .

For a formula ϕ , variable assignments V (individual) and W (predicate), we define whether (or not resp.) ϕ under V and W is true in $(T, <)$, written $(T, <), V, W \models \phi$ by induction on the quantifier depths of ϕ .

Given some ϕ , suppose that for all its subformulas ψ , for all variable assignments V and W , we have defined whether or not $(T, <), V, W \models \psi$. For variable assignments V and W we define:

$$\begin{aligned}
 (T, <), V, W \models x_i < x_j & \text{ iff } V(x_i) < V(x_j); \\
 (T, <), V, W \models x_i = x_j & \text{ iff } V(x_i) = V(x_j); \\
 (T, <), V, W \models P_i(x_j) & \text{ iff } V(x_j) \in W(P_i); \\
 (T, <), V, W \models \neg \alpha & \text{ iff } (T, <), V, W \not\models \alpha; \\
 (T, <), V, W \models \alpha \wedge \beta & \text{ iff } (T, <), V, W \models \alpha \text{ and } (T, <), V, W \models \beta; \\
 (T, <), V, W \models \exists x_i \alpha & \text{ iff there exists some } t \in T \text{ such that} \\
 & (T, <), V[x_i \mapsto t], W \models \alpha. \\
 (T, <), V, W \models \exists P_j \alpha & \text{ iff there is some } S \subseteq T \text{ such that} \\
 & (T, <), V, W[P_j \mapsto S] \models \alpha.
 \end{aligned}$$

This is standard second-order semantics. Note that it is easy to show that the truth of a formula does not depend on the assignment to variables which do not appear free in the formula.

Mostly we will be interested in fragments of the full second-order monadic logic. In particular, we refer to the first-order monadic logic of linear order which contains just those formulas with no quantification of predicate variables. We will also mention the universal second-order monadic logic of linear order which contains just those formulas which consist of a first-order monadic formula nested under zero or more universal quantifications of predicate variables.

The important correspondence for us is that between temporal logics such as *US/LT* and the first-order monadic logic. Most of the temporal logics which we will consider allow a certain equivalence between their formulas and first-order monadic formulas. To define this we need to use a fixed one-to-one correspondence between the propositional atoms of the temporal language and monadic predicate variables. Let us suppose that $p_i \in \mathcal{L}$ corresponds to $P_i \in \mathcal{Q}$.

The translation will propagate upwards through the full temporal language provided that each of the connectives have a first-order translation. In particular we require for any n -ary temporal connective C some first-order monadic formula $\phi_C(t, P_1, \dots, P_n)$ which corresponds to $C(p_1, \dots, p_n)$. We say that ϕ_C is the (*first-order*) *table* of C iff for every linear order $(T, <)$, for all $h : \mathcal{L} \rightarrow \wp(T)$, for all $t_0 \in T$, for all variable assignments V and W ,

$$\begin{aligned}
 (T, <, h), t_0 \models C(p_1, \dots, p_n) \\
 \text{iff } (T, <), V[t \mapsto t_0], W[P_1 \mapsto h(p_1), \dots, P_n \mapsto h(p_n)] \models \phi_C.
 \end{aligned}$$

U and S have first-order tables as follows: the table of U is $\phi_U = \exists s((t < s) \wedge P_1(s) \wedge \forall r((t < r \wedge r < s) \rightarrow P_2(r)))$. The table of S is the mirror image.

If we have a temporal logic with first-order tables for its connectives then it is straightforward to define a meaning-preserving translation (to the first-order monadic language) of all formulas in the language. The translation is

Handbook of Philosophical Logic

Gabbay, D.; Guenther, F. (Eds.)

2002, XIII, 350 p. 1 illus., Hardcover

ISBN: 978-1-4020-0599-2