Adding a temporal dimension to a logic system

MARCELO FINGER and DOV M. GABBAY
Imperial College, Department of Computing

January 11, 1993

Abstract. We introduce a methodology whereby an arbitrary logic system **L** can be enriched with temporal features to create a new system **T(L)**. The new system is constructed by combining **L** with a pure propositional temporal logic **T** (such as linear temporal logic with "Since" and "Until") in a special way. We refer to this method as "adding a temporal dimension to **L**" or just "temporalising **L**".

We show that the logic system **T(L)** preserves several properties of the original temporal logic like soundness, completeness, decidability, conservativeness and separation over linear flows of time.

We then focus on the temporalisation of first-order logic, and a comparison is made with other first-order approaches to the handling of time.

1. Introduction

We are interested in describing the way that a system $\mathcal{S}$, specified in a logic **L**, changes over time. There are two main methods for doing so. In the *external* method, snapshots of $\mathcal{S}$ are taken at different moments of time as describing the state of $\mathcal{S}$ at those times. We can write $\mathcal{S}_t$ for the way $\mathcal{S}$ is at time t, and use **L** to describe $\mathcal{S}_t$. We then externally add a temporal system that allows us to relate different $\mathcal{S}_t$ at different times t.

In the *internal* method, instead of considering $\mathcal{S}$ as a whole, we observe how $\mathcal{S}$ is built up from internal components and we transform these components into time dependent building blocks. The internal temporal description of each component will give us the temporal description of the whole system $\mathcal{S}$. We can assume that $\mathcal{S}$ can be completely described through its components and that the way the components are put together to make $\mathcal{S}$ into a whole is also a (possibly time varying) component.

Both the external and the internal methods have their counterpart in logic as well. A temporal logical systems with temporal connectives such as "Since" and "Until" is the result of externally turning classical logic into a temporal (time varying) system. The use of a two-sorted predicate logic with one time variable in which atoms are of the form $A(t, x)$, with t time and x an element of a domain, is an internal way of making classical logic into a temporal system.

The purpose of this paper is to investigate the external way of temporalising a logic system. In the external approach, we do not need to have detailed knowledge about the components of the system $\mathcal{S}$ or about the logical components of its description in **L**. We introduce a methodology whereby an arbitrary logic system **L** can be enriched with temporal features to create a new system **T(L)**. The new system is constructed by combining **L** with a pure propositional temporal logic **T** (e.g. linear-time temporal logic with "Since" and "Until") in a special way. We refer to this method as "adding a temporal dimension to **L**" or just "temporalising **L**". The method we use is not confined to temporal features only, but is a methodology of combining two logics by substituting one in another. Thus in the general case we can combine any two logic systems $\mathbf{L_1}$ and $\mathbf{L_2}$ to form $\mathbf{L_1}(\mathbf{L_2})$.

In classical propositional temporal logic we add to the language of classical propositional logic the connectives P and F and we are able to express statements like "*in the future* a certain proposition a will hold" by constructing sentences of the form Fa. The idea we develop here is to apply temporal operators not only to propositions but also to sentences from an arbitrary logic system **L**.

Our aim can be viewed as describing both the "statics" and the "dynamics" of a logic system, while still remaining in a logical framework. The "statics" is given by the properties of the underlying logic system **L**; in propositional temporal logic **T**, we already have the ability to describe the "dynamics", i.e. changes in time of a set of atomic propositions. This point of view leads us to combine the upper-level temporal **T** system with an underlying logic system **L** so as to describe the evolution in time of a theory in **L** and its models.

Another more general point of view comes from the work in (Gabbay 1991d) about networks of logic databases. A database is considered to be a model of a theory in some logic system $\mathbf{L}_2$ and the interaction between databases is modelled by another logic system $\mathbf{L}_1$; therefore, two basic logic levels can be identified, namely the local logic $\mathbf{L}_2$ and the global logic $\mathbf{L}_1$. The two systems are illustrated in Figure 1 with a temporal upper-level system **T** in the place of $\mathbf{L}_1$ and an arbitrary underlying logic system **L** in the place of $\mathbf{L}_2$.

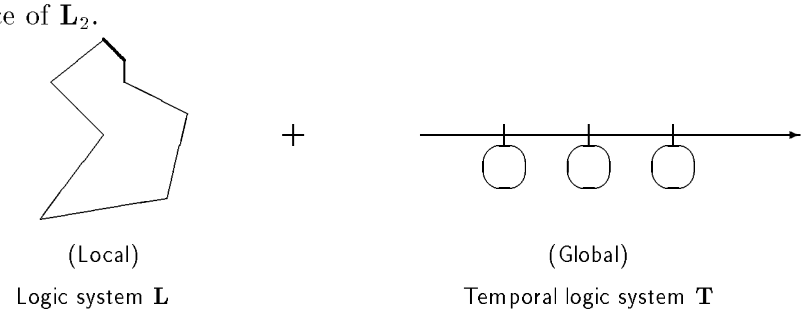


Figure 1: Two logic levels in a database network

We consider a network of databases distributed in time, as an extension of the more usual idea of a network of databases distributed in space. The underlying logic system **L** characterises the local behaviour of a database, i.e. the way queries are answered by a single element of the network. The upper-level logic system describes how one local system (at some moment in time) relates to another local system (at some other moment in time). We combine those two logic systems to be able to reason about the "temporal network" as a whole, creating a logic system **T(L)**. The result of this combination is the addition of a temporal dimension to system **L**, as illustrated in Figure 2.

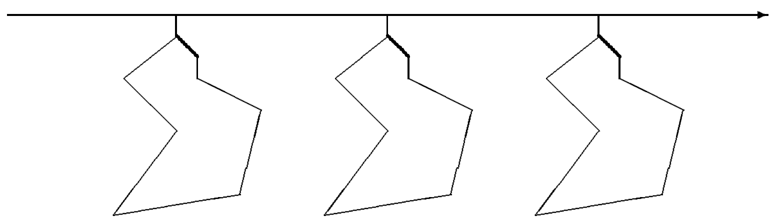


Figure 2: The logic system **T(L)**

The above point of view is not yet the most general setting for our operations. One may ask a general question: given two logics $\mathbf{L}_1$ and $\mathbf{L}_2$, can we combine them into one logic? Suppose we take a disjoint union of the two systems, for example a modal logic system **K**, with modality $\Box_1$, and a modal logic system **S4**, with modality $\Box_2$. Here $\mathbf{L}_1 = \mathbf{K}$ and $\mathbf{L}_2 = \mathbf{S4}$. Form a language with $\{\Box_1, \Box_2\}$ and the separate axioms on $\Box_1$ (**K** axioms) and on $\Box_2$ (**S4** axioms). What do we know about the union? What is the semantics? These questions have been recently investigated by Fine and Schurz (1992) and by Kracht and Wolter (1991), in a framework in which several independently axiomatisable monomodal systems were syntactically combined. The temporal case, however, differs from those since temporal logic is a bimodal system where the two modalities, one for the past and one for the future, always interact. The methods in (Kracht and Wolter 1991) do not immediately apply. This paper differs from the above papers in two respects. First we are dealing with binary connectives *Since* (S) and *Until* (U). Secondly and most importantly, we are not arbitrarily combining two logics but rather embedding one logic inside the other. If we were to embed one modality within another in the framework above we would syntactically combine them ruling out the formulae containing $\Box_1$ within the scope of $\Box_2$. This yields what we call $\mathbf{L_1}(\mathbf{L_2})$ ($\Box_1$ is *externally* applied to $\mathbf{L_2}$). The special case where $\mathbf{L}_1$ is a temporal logic **T** and $\mathbf{L}_2$ is an arbitrary logic **L**, gives us **T(L)**, that we study in this paper.

General combinations of logics have been addressed in the literature in various forms. Combinations of tense and modality were discussed in (Thomason 1984), without explicitly providing a general methodology for doing so. A methodology for constructing logics of belief based on existing deductive systems was proposed by Konolige (1986); in this case, the language of the original system was the base for the construction of a new modal language, and the modal logic system thus generated had its semantics defined in terms of the inferences of the original system. The model theory used by Konolige, called a *deductive model*, was the connection between the original system and the modal one. Here we present a quite different methodology, in which the language, inference system and semantics of $\mathbf{T(L)}$ are based on, respectively, the language, the inference system and the semantics of $\mathbf{T}$ and $\mathbf{L}$. Recently we have developed a general methodology for combining any two logics through fibring their semantics (Gabbay 1991a; Gabbay 1992); the assumptions on the semantics of the candidate logics are very general and yield many known results.

Extensions of temporal logic are also found in the literature. In (Casanova and Furtado 1982) a family of formal languages was generated by means of certain mechanisms to define temporal modalities; the approach there was based on grammars and the resulting family of languages was claimed to be useful in expressing transition constraints for databases. Gabbay (1991b) mixes two predicate languages $\mathbf{G}$ and $\mathbf{L}$, generating the language $\mathbf{L}_k^*(\mathbf{G})$, a two-sorted predicate language in which one sort comes from terms originated in $\mathbf{G}$ and the other sort comes from terms originated in $\mathbf{L}$; in the case that the original language $\mathbf{G}$ is supposed to describe an order relation $<$, the resulting system $\mathbf{L}_k^*(\mathbf{G})$ can be seen as a predicate logic like approach to temporal logic. Such a construction corresponds to an *internal* way of adding a temporal dimension to a logic system. We propose in this work a different approach, in which temporal modalities are applied to an existing logic system and thence a temporal dimension is added. Eventually, we are going to informally compare the internal and external approaches in Section 7.

The rest of the paper is organised as follows. In Section 2 we formalize the idea of temporalising a logic system $\mathbf{L}$ in terms of the S, U-temporal logic and we show the soundness and completeness of the resulting system $\mathbf{T(L)}$ over linear time. Section 3 shows that $\mathbf{T(L)}$ preserves the decidability property of system $\mathbf{L}$ over linear time, and the complexity of the decision procedure is estimated. Section 4 shows that $\mathbf{T(L)}$ is a conservative extension of $\mathbf{L}$. Section 5 shows that $\mathbf{T(L)}$ has the separation property, which is useful to specify how the past states of a database influence its future states. In Section 6 we discuss the temporalisation of first-order logic as a particularly interesting application; two different temporalisations of first-order logic are shown, yielding two expressively different logics. Finally, in Section 7 we show how the added temporal dimension can be internalised in first-order logic and we compare the temporalised approach with the internalised first-order one.

2. Temporalising an Existing Logic

This section will construct $\mathbf{T(L)}$ out of $\mathbf{T}$ and $\mathbf{L}$. Our $\mathbf{T}$ is the temporal system with "Since" and "Until", described below. Our $\mathbf{L}$ is in general any logic and in particular it can be classical predicate logic. We construct $\mathbf{T(L)}$ by allowing substitution of formulae of $\mathbf{L}$ for the atoms of formulae of $\mathbf{T}$. We are *not* allowing the substitution of formulae of $\mathbf{T}$ or even formulae of $\mathbf{T(L)}$ for atoms of $\mathbf{L}$. Thus the temporal connectives of $\mathbf{T}$ are never within the scope of connectives of $\mathbf{L}$.

Next we first define $\mathbf{T}$, both syntactically and semantically. Then we define $\mathbf{T(L)}$ syntactically and semantically and we prove soundness and completeness for $\mathbf{T(L)}$.

2.1. Propositional Temporal Logics

We present here several propositional temporal logics of "Since" and "Until"; these logics are defined over the same language but vary in the nature of the flow of time they describe. So the language is defined starting from a set of propositional letters $\mathcal{P}$ and then formulas are built up from the propositional letters using the boolean operators $\neg$ (negation) and $\wedge$ (conjunction) and the two-place temporal operators S (since) and U (until). Other boolean connectives such as $\vee$ (disjunction), $\rightarrow$ (material implication) and $\leftrightarrow$ (material biconditional), as well as the abbreviations $\top$ (constant true) and $\bot$ (constant false), can be defined in terms of $\neg$ and $\wedge$; similarly for other temporal operators like P (sometime in the past), F (sometime in the future), H (always in the past) and G (always in the future) with respect to U and S.

In the following, propositional letters are represented by p, q, r and s, and temporal formulae are represented by upper case letter A, B, C and D.

DEFINITION 2.1. **Syntax of propositional temporal logics**

Let $\mathcal{P}$ be a denumerably infinite set of propositional letters. The set $\mathcal{L}_{\mathbf{S},\mathbf{U}}$ of temporal propositional formulas is the smallest set such that:

- $\mathcal{P} \subset \mathcal{L}_{\mathbf{S},\mathbf{U}}$;
- If A and B are in $\mathcal{L}_{\mathbf{S},\mathbf{U}}$, then $\neg A$ and $(A \wedge B)$ are in $\mathcal{L}_{\mathbf{S},\mathbf{U}}$;
- If A and B are in $\mathcal{L}_{\mathbf{S},\mathbf{U}}$, then $S(A, B)$ and $U(A, B)$ are in $\mathcal{L}_{\mathbf{S},\mathbf{U}}$.

The *mirror image* of a formula is obtained by changing U by S and vice-versa. □

The outermost pair of brackets of a formulas are sometimes omitted when no ambiguity is implied. Boolean connectives are defined in the standard way, while temporal operators can be defined by:

$$
\begin{aligned}
FA &=_{def} U(A, \top)\\
PA &=_{def} S(A, \top)\\
GA &=_{def} \neg F \neg A\\
HA &=_{def} \neg P \neg A
\end{aligned}
$$

A flow of time is an ordered pair $\mathcal{F} = (T, <)$, where T is a nonempty set of time points and $<$ is a binary relation over T. A valuation g is a function assigning to every time point t in T a set of propositional letters $g(t) \subseteq \mathcal{P}$, namely the set of proposition letters that are true at the time point t. A *model* $\mathcal{M}$ is a 3-tuple $(T, <, g)$, where $(T, <)$ is the underlying flow of time and g is a valuation. $\mathcal{M}, t \models A$ reads the formula A holds over model $\mathcal{M}$ at time point t and is defined recursively as follows.

DEFINITION 2.2. **Semantics of propositional temporal logic**

$\mathcal{M}, t \models p, p \in \mathcal{P}$	iff $p \in g(t)$.
$\mathcal{M}, t \models \neg A$	iff it is not the case that $\mathcal{M}, t \models A$.
$\mathcal{M}, t \models A \wedge B$	iff $\mathcal{M}, t \models A$ and $\mathcal{M}, t \models B$.
$\mathcal{M}, t \models S(A, B)$	iff there exists an $s \in T$ with $s < t$ and $\mathcal{M}, s \models A$ and for every $u \in T$, if $s < u < t$ then $\mathcal{M}, u \models B$.
$\mathcal{M}, t \models U(A, B)$	iff there exists an $s \in T$ with $t < s$ and $\mathcal{M}, s \models A$ and for every $u \in T$, if $t < u < s$ then $\mathcal{M}, u \models B$.

□

A formula A is *valid* over a class $\mathcal{K}$ of flows of time, indicated by $\mathcal{K} \models A$, if for every $\mathcal{M}$ whose underlying flow of time is in $\mathcal{K}$ and for every time point $t \in T$, $\mathcal{M}, t \models A$. If Σ is a set of formulae, we write $\mathcal{K} \models \Sigma$ to indicate that $\mathcal{K} \models A$ for every $A \in \Sigma$. Therefore, for different classes $\mathcal{K}$ we have different sets of valid formulae.

A minimal axiomatic system for the S, U-temporal logic over a class $\mathcal{K}$,$\vdash_{\mathbf{S,U}}$, contains the following axioms:

A0 all classical tautologies
A1a $G(p \to q) \to (U(p, r) \to U(q, r))$
A1b $H(p \to q) \to (S(p, r) \to S(q, r))$
A2a $G(p \to q) \to (U(r, p) \to U(r, q))$
A2b $H(p \to q) \to (S(r, p) \to S(r, q))$
A3a $(p \wedge U(q, r)) \to U(q \wedge S(p, r), r)$
A3b $(p \wedge S(q, r)) \to S(q \wedge U(p, r), r)$

Note that the axioms above come in pairs, represented by **a** and **b**, such that one is the mirror image of the other. The inference rules are:

Subst Uniform Substitution, i.e. let $A(q)$ be an axiom containing the propositional letter q and let B be any formula, then from $\vdash A(q)$ infer $\vdash A(q\backslash B)$ by substituting all appearances of q in A by B.

MP Modus ponens: from $\vdash A$ and $\vdash A \to B$ infer $\vdash B$.

***T*G** Temporal Generalisation: from $\vdash A$ infer $\vdash HA$ and $\vdash GA$.

A *deduction* is a finite string of formulae each of which is either an axiom or follows from earlier formulae by a rule of inference. A *theorem* is any formula A appearing as a last element of a deduction, and we indicate by $\vdash_{\mathbf{S},\mathbf{U}} A$. The axioms of $\vdash_{\mathbf{S},\mathbf{U}}$ can be extended by a set of axioms Σ so as to impose restrictions on the flow of time, therefore generating the inference system $\vdash_{\mathbf{S},\mathbf{U}(\mathbf{\Sigma})}$. When Σ is the empty set we have $\vdash_{\mathbf{S},\mathbf{U}} = \vdash_{\mathbf{S},\mathbf{U}(\varnothing)}$. A set of formulae is consistent if we cannot deduce falsity ($\perp$) from it.

We say that an inference system is *sound* and *complete* with respect to a class $\mathcal{K}$ of flows of time if

$$\mathcal{K} \models A \text{ iff } \vdash A,$$

or equivalently,

$$A \text{ is consistent iff } A \text{ has a model over } \mathcal{K},$$

soundness corresponding to the *if* part and completeness [1] to the *only if* part. We write **S,U**/$\mathcal{K}$ to indicate that fact.

If we consider $\mathcal{K}_0$, the class of all flows of time, we have the following well known result.

THEOREM 2.1. **(Soundness and Completeness of S,U/$\mathcal{K}_0$)**
The inference system $\vdash_{\mathbf{S},\mathbf{U}}$ is sound and complete with respect to the class $\mathcal{K}_0$.

An elegant proof of the above is given by Xu (1988). A proof of completeness for the class of transitive linear flows of time, $\mathcal{K}_{lin}$, is given by Burgess (1982) adding the following set Σ of axioms together with their mirror images (**b** axioms).

A4a $U(p,q) \to U(p, q \wedge U(p,q))$

A5a $U(q \wedge U(p,q), q) \to U(p,q)$

A6a $(U(p,q) \wedge U(r,s)) \to$
$$(U(p \wedge r, q \wedge s) \vee U(p \wedge s, q \wedge s) \vee U(q \wedge r, q \wedge s))$$

[1] This is sometimes called *weak completeness*; strong completeness says that for any (possibly infinite) set of formula Γ, if Γ is consistent then Γ has a model. Strong completeness implies weak completeness but the converse is not true.

Burgess actually used an extra axiom, but Xu (1988) proved the same result omitting it and axiom **A5b**. Axioms **A4ab** and **A5a** are responsible for restricting the class of flows of time to a transitive one. The pair of axioms **A6ab** are responsible for restricting the class of flows of time to a linear one. Adding the axiom

A7a $(p \wedge H\,p) \rightarrow F\,H\,p$

and its mirror image restricts the flow of time to a discrete one. Extending original proofs of completeness to include new axioms over a more restricted flow of time is discussed by Burgess (1984). With axioms **A0–A7** we have soundness and completeness results for a class of linear, discrete and transitive flows of time. There are also complete axiomatisations **S,U/R** over the reals (Gabbay and Hodkinson 1990; Reynolds 1992) and **S,U/Z** over the integers (Reynolds 1992).

2.2. Logic Systems and Their Temporalised Form

Having defined a family of S, U-temporal logics, we now externally apply such logic systems to any other logic system $\mathbf{L}$, i.e. we "temporalise" $\mathbf{L}$.

A logical system is a pair $\mathrm{L} = \langle \mathcal{L}_\mathbf{L}, \vdash_\mathbf{L} \rangle$, where $\mathcal{L}_\mathbf{L}$ is its language and $\vdash_\mathbf{L}$ is its inference system; the set $\mathcal{L}_\mathbf{L}$ must be countable. A model for the logic system $\mathbf{L}$ is a structure $\mathcal{M}_\mathbf{L}$ and we denote $\mathcal{M}_\mathbf{L} \models \alpha$ when a formula $\alpha \in \mathcal{L}_\mathbf{L}$ is true under the model $\mathcal{M}_\mathbf{L}$. The class of all models of $\mathbf{L}$ is denoted by $\mathcal{K}_\mathbf{L}$ and a formula α is said to be valid if $\mathcal{M}_\mathbf{L} \models \alpha$ for all $\mathcal{M}_\mathbf{L} \in \mathcal{K}_\mathbf{L}$.

A logical system $\mathbf{L}$ is said to be sound if, whenever $\vdash_\mathbf{L} \alpha$, we have $\mathcal{M}_\mathbf{L} \models \alpha$ for all $\mathcal{M}_\mathbf{L} \in \mathcal{K}_\mathbf{L}$. The logical system $\mathbf{L}$ is said to be complete if, whenever $\mathcal{M}_\mathbf{L} \models \alpha$ for all $\mathcal{M}_\mathbf{L} \in \mathcal{K}_\mathbf{L}$, we have that $\vdash_\mathbf{L} \alpha$.

We constrain the logic system $\mathbf{L}$ to be an extension of classical logic, i.e. all propositional tautologies must be valid in it. This constraint is due to the fact that all S, U-temporal logics presented above are extensions of classical logic and any of them can be taken as the logic $\mathbf{T}$ in which we base the temporalisation. We discuss later in this section what should be the case if $\mathbf{L}$ is not an extension of classical logic.

DEFINITION 2.3. **Boolean combinations and monolithic formulae**

The set $\mathcal{L}_\mathbf{L}$ is partitioned in two sets, $BC_\mathbf{L}$ and $ML_\mathbf{L}$. A formula $A \in \mathcal{L}_\mathbf{L}$ belongs to the set of *boolean combinations*, $BC_\mathbf{L}$, iff it is built up from other formulae by the use of one of the boolean connectives $\neg$ or $\wedge$ or any other connective defined only in terms of those; it belongs to the set of *monolithic formula* $ML_\mathbf{L}$ otherwise. □

We can proceed then to the definition of the temporalised language. In the following we will use α, β, γ, ... , to range over formulae of $\mathbf{T(L)}$.

The result of temporalising the logic system $\mathbf{L}$ is the logic system $\mathbf{T(L)}$ $= \langle \mathcal{L}_{\mathbf{T(L)}}, \vdash_{\mathbf{T(L)}} \rangle$ and its models by $\mathcal{M}_{\mathbf{T(L)}}$. The alphabet of the temporalised language uses the alphabet of $\mathbf{L}$ plus the two-place operators S and U, if they are not part of the alphabet of $\mathbf{L}$; otherwise, we use S_2 and U_2 or any other proper renaming.

DEFINITION 2.4. **Temporalised formulae**

The set $\mathcal{L}_{\mathbf{T(L)}}$ of formulae of the logic system $\mathbf{L}$ is the smallest set such that:
1. If $\alpha \in ML_{\mathbf{L}}$, then $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$;
2. If $\alpha, \beta \in \mathcal{L}_{\mathbf{T(L)}}$ then $\neg\alpha \in \mathcal{L}_{\mathbf{T(L)}}$ and $(\alpha \wedge \beta) \in \mathcal{L}_{\mathbf{T(L)}}$;
3. If $\alpha, \beta \in \mathcal{L}_{\mathbf{T(L)}}$ then $S(\alpha, \beta) \in \mathcal{L}_{\mathbf{T(L)}}$ and $U(\alpha, \beta) \in \mathcal{L}_{\mathbf{T(L)}}$.

The set of *maximal monolithic subformulae of* α, $Mon(\alpha)$, is the set of all monolithic subformulae of α that are used to build α up by the rules above. $\square$

It is obvious from the definition above that the set $\mathcal{L}_{\mathbf{T(L)}}$ is denumerably infinite. Note that from item 1 and 2 of the definition above, it follows that $\mathcal{L}_{\mathbf{L}} \subset \mathcal{L}_{\mathbf{T(L)}}$. The reason to define the base case in item 1 in terms of monolithic formulae of $\mathbf{L}$ instead of simply defining it in terms of any formula in $\mathcal{L}_{\mathbf{L}}$ is that we would have a double parsing problem. In fact, suppose an item $1'$ that would state that:

$1'$. If $\alpha \in \mathcal{L}_{\mathbf{L}}$, then $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$.

Suppose we want to define a function over the set of formulae, e.g. the depth of the parsing tree of a formula. Consider the formula $(\alpha \wedge \beta) \in \mathcal{L}_{\mathbf{L}}$; it would belong to $\mathcal{L}_{\mathbf{T(L)}}$ both by items $1'$ and 2. If we parse it by $1'$, then its depth will be 0, but if we parse it by 2, its depth will be 1, i.e. depth is not a well defined function. To avoid such problem we introduce the restriction to monolithic formulae in item 1. We also note that, for instance, if $\Box$ is an operator of the alphabet of $\mathbf{L}$ and α and β are two formulae in $\mathcal{L}_{\mathbf{L}}$, the formula $\Box U(\alpha, \beta)$ is *not* in $\mathcal{L}_{\mathbf{T(L)}}$.

There is nothing to prevent us from defining the temporalisation in terms of some F, P-temporal language, but since the language with S and U is more expressive it received our preference.

If $\mathbf{L}$ is an extension of classical logic, we must pay attention to some details before being able to describe the semantics of $\mathbf{T(L)}$. First, if $\mathcal{M}_{\mathbf{L}}$ is a model in the class of models of $\mathbf{L}$, for every formula $\alpha \in \mathcal{L}_{\mathbf{L}}$ we must have either $\mathcal{M}_{\mathbf{L}} \models \alpha$ or $\mathcal{M}_{\mathbf{L}} \models \neg\alpha$. For example, if $\mathbf{L}$ is a modal logic system, e.g. **S4**, we must consider a "current world" o as part of its model to achieve that condition. Second, we must be careful about the semantics of boolean connectives in the temporalised system. The construction of temporalised formulae based on monolithic formulae of $\mathcal{L}_{\mathbf{L}}$ guarantees that the semantics of the boolean connectives is the same in both the upper-level temporal logic system $\mathbf{T}$ and in the temporalised system $\mathbf{T(L)}$.

The language of $\mathbf{T(L)}$ is independent of the underlying flow of time, but not its semantics and inference system, so we must fix a class $\mathcal{K}$ of flows of time over which the temporalisation is defined; this is equivalent to fixing one logic $\mathbf{T}$ among the family of temporal logics presented above. We are then in a position to define the semantics of the temporalised logic system $\mathbf{T(L)}$.

DEFINITION 2.5. **Semantics of the temporalised logic**

Consider a flow of time $(T, <) \in \mathcal{K}$ and a function $g : T \to \mathcal{K}_{\mathbf{L}}$, mapping every time point in T to a model in the class of models of $\mathbf{L}$. A model of $\mathbf{T(L)}$ is a triple $\mathcal{M}_{\mathbf{T(L)}} = (T, <, g)$ and the fact that α is true in the model $\mathcal{M}_{\mathbf{T(L)}}$ at time point t is represented by $\mathcal{M}_{\mathbf{T(L)}}, t \models \alpha$. The semantics of $\mathbf{T(L)}$ is given by:

$$
\begin{array}{ll}
\mathcal{M}_{\mathbf{T(L)}}, t \models \alpha,\ \alpha \in ML_{\mathbf{L}} & \text{iff } g(t) = \mathcal{M}_{\mathbf{L}} \text{ and } \mathcal{M}_{\mathbf{L}} \models \alpha. \\
\mathcal{M}_{\mathbf{T(L)}}, t \models \neg\alpha & \text{iff it is not the case that } \mathcal{M}_{\mathbf{T(L)}}, t \models \alpha. \\
\mathcal{M}_{\mathbf{T(L)}}, t \models (\alpha \wedge \beta) & \text{iff } \mathcal{M}_{\mathbf{T(L)}}, t \models \alpha \text{ and } \mathcal{M}_{\mathbf{T(L)}}, t \models \beta. \\
\mathcal{M}_{\mathbf{T(L)}}, t \models S(\alpha, \beta) & \text{iff there exists } s \in T \text{ such that } s < t \text{ and} \\
& \quad \mathcal{M}_{\mathbf{T(L)}}, s \models \alpha \text{ and for every } u \in T, \text{ if} \\
& \quad s < u < t \text{ then } \mathcal{M}_{\mathbf{T(L)}}, u \models \beta. \\
\mathcal{M}_{\mathbf{T(L)}}, t \models U(\alpha, \beta) & \text{iff there exists } s \in T \text{ such that } t < s \text{ and} \\
& \quad \mathcal{M}_{\mathbf{T(L)}}, s \models \alpha \text{ and for every } u \in T, \text{ if} \\
& \quad t < u < s \text{ then } \mathcal{M}_{\mathbf{T(L)}}, u \models \beta.
\end{array}
$$

□

We write $\mathbf{T(L)} \models \alpha$ if, for every model $\mathcal{M}_{\mathbf{T(L)}}$ whose underlying flow of time $(T, <) \in \mathcal{K}$ and for every time point $t \in T$, it is the case that $\mathcal{M}_{\mathbf{T(L)}}, t \models \alpha$.

The inference system of $\mathbf{T(L)}/\mathcal{K}$ is given by the following:

DEFINITION 2.6. **Axiomatisation for $\mathbf{T(L)}$**

- The axioms of $\mathbf{T}/\mathcal{K}$;
- The inference rules of $\mathbf{T}/\mathcal{K}$;
- For every formula α in $\mathcal{L}_{\mathbf{L}}$, if $\vdash_{\mathbf{L}} \alpha$ then $\vdash_{\mathbf{T(L)}} \alpha$.

□

The third item above constitutes a new inference rule needed to preserve the theoremhood of formulae of the logic system $\mathbf{L}$. Therefore we call it **Preserve**. The only inference rules we are considering in this paper are **Subst**, **MP** and **TG**, but other rules such as the irreflexivity rule **IRR**, (Gabbay and Hodkinson 1990), can also be added.

The first concern about the axiomatisation is its soundness, i.e. if whenever $\vdash_{\mathbf{T(L)}} \alpha$ we have $\mathbf{T(L)} \models \alpha$.

THEOREM 2.2. **(Soundness of T(L))** *If the logic system* **L** *is sound and* **S,U**$/\mathcal{K}$ *is sound over the class of flows of time* $\mathcal{K}$*, then so is the logic system* **T(L)**$/\mathcal{K}$*.*

PROOF. Soundness of **S,U**$/\mathcal{K}$ gives us the validity of the axioms over $\mathcal{K}$. As for the inference rules, soundness of **L** guarantees that all formulae generated by **Preserve** are valid; soundness of **S,U**$/\mathcal{K}$ guarantees that the other inference rules, when applied to valid formulae, always generate valid formulae. □

Completeness is discussed later in 2.4. Let us first present a few examples of the temporalisation of an existing logic system.

EXAMPLE 2.1. *Temporalising modal logic of belief*

Suppose we have a propositional modal logic of belief $\mathbf{B} = \langle \mathcal{L}_\mathbf{B}, \vdash_\mathbf{B} \rangle$ with the modal operator B, in which Bp is intended to mean that p is a proposition that is believed by an agent. The axiomatisation, $\vdash_\mathbf{B}$, is given by the basic modal logic system **K** plus the transitivity axiom **4** as one of the introspective properties of belief systems in (Hintikka 1962):

$$K \begin{cases} \text{All propositional tautologies} \\ B(p \to q) \to (Bp \to Bq) \\ \text{Rules: } \textbf{Subst}, \textbf{MP}, \textbf{Generalisation} \end{cases} + \; Bp \to BBp$$

The transitivity axiom means that, if some fact is believed, it is believed to be believed, which represents a positive introspection of the believing agent; for a discussion on modal logics of belief, see (Halpern and Moses 1985). This system is provided with a standard Kripke semantics for modal logics (Hughes and Cresswell 1968), with a set of possible worlds W, an accessibility relation R and a valuation function V, so that $\mathcal{M}_\mathbf{B} = (W, R, V)$ is a model structure in which the accessibility relation R is transitive. Actually, we are considering $\mathcal{M}_\mathbf{B} = (W, R, V, o)$, where o is a “current world” from which the observations are made, so that we may have both validity and satisfiability in the model theory of **B**.

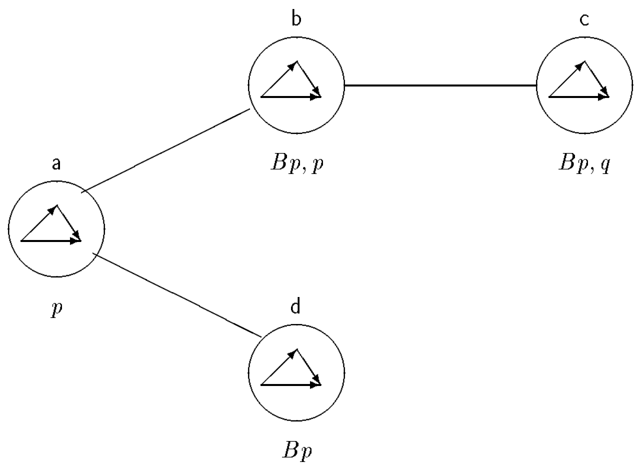
Consider the temporalised logic system $\mathbf{T}(\mathbf{B})$ over the class $\mathcal{K}_0$ of all flows of time. Its inference system $\vdash_{\mathbf{T}(\mathbf{B})}$, for example, gives us as theorems

$$B(p \to q) \to (Bp \to Bq)$$
$$\neg(Bp \wedge \neg Bp)$$
$$GB\neg(Bp \wedge \neg Bp)$$
$$G(Bp \to q) \to (U(Bp, Bq) \to U(q, Bq)).$$

If we have a theory $\Gamma = \{GBp, Bp \to Fp, U(q, Bp)\}$. We construct one possible model $\mathcal{M}_{\mathbf{T}(\mathbf{B})}$ by choosing a flow of time with $T = \{a, b, c, d\}$ and the partial order $< \, = \{(a, b), (b, c), (a, c), (a, d)\}$. We construct the assignment g such that:

$g(b) = \mathcal{M}^{a}_{\mathbf{B}} \models p$
$g(b) = \mathcal{M}^{b}_{\mathbf{B}} \models Bp \wedge p$,
$g(c) = \mathcal{M}^{c}_{\mathbf{B}} \models Bp \wedge q$ and
$g(d) = \mathcal{M}^{d}_{\mathbf{B}} \models Bp$

In the resulting model $\mathcal{M}_{\mathbf{T(B)}}$ we have $\mathcal{M}_{\mathbf{T(B)}}, a \models \Gamma$ as illustrated below.



EXAMPLE 2.2. *Temporalising propositional logic*

Consider classical propositional logic $\mathbf{PL} = \langle \mathcal{L}_{\mathbf{PL}}, \vdash_{\mathbf{PL}} \rangle$. Its temporalisation generates the logic system $\mathbf{T}(\mathbf{PL}) = \langle \mathcal{L}_{\mathbf{T(PL)}}, \vdash_{\mathbf{T(PL)}} \rangle$.

It is not difficult to see that $\mathcal{L}_{\mathbf{T(PL)}} = \mathcal{L}_{\mathbf{S,U}}$ and $\vdash_{\mathbf{T(PL)}} = \vdash_{\mathbf{S,U}}$, i.e. the temporalised version of **PL** over any $\mathcal{K}$ is actually the temporal logic **T** = **S,U**/$\mathcal{K}$. With respect to $\mathcal{M}_{\mathbf{T(L)}}$, the function h actually assigns, for every time point, a **PL** model.

EXAMPLE 2.3. *Temporalising S, U-temporal logic*

If we temporalise over $\mathcal{K}$ the one-dimensional logic system **S,U**/$\mathcal{K}$ we get the logic system $\mathbf{T}(\mathbf{S},\mathbf{U}) = \langle \mathcal{L}_{\mathbf{T(S,U)}}, \vdash_{\mathbf{T(S,U)}} \rangle = \mathbf{T}^2(\mathbf{PL})/\mathcal{K}$. In this case we have to rename the two-place operators S and U of the temporalised alphabet to, say, S_2 and U_2.

In order to obtain a model for **T(S,U)**, we must fix a "current time", o, in $\mathcal{M}_{\mathbf{S,U}} = (T_1, <_1, g_1)$, so that we can construct the model $\mathcal{M}_{\mathbf{T(S,U)}} = (T_2, <_2, g_2)$ as previously described. Note that, in this case, the flows of time $(T_1, <_1)$ and $(T_2, <_2)$ need not to be the same. $(T_2, <_2)$ is the flow of time of the upper-level temporal system whereas $(T_1, <_1)$ is the flow of time of the underlying logic which, in this case, happens to be a temporal logic.

The logic system we obtain by temporalising S, U-temporal logic is the two-dimensional temporal logic described in (Finger 1992).

EXAMPLE 2.4. *N-dimensional temporal logic*

If we repeat the process started in the last two examples, we can construct an n-dimensional temporal logic $\mathbf{T^n(PL)/K}$ (its alphabet including S_n and U_n) by temporalising a $(n-1)$-dimensional temporal logic.

Every time we add a temporal dimension, we are able to describe changes in the underlying system. Temporalising the system $\mathbf{L}$ once, we are creating a way of describing the history of $\mathbf{L}$; temporalising for the second time, we are describing how the history of $\mathbf{L}$ is viewed in different moments of time. We can go on indefinitely, although it is not clear what is the purpose of doing so.

The assumption that the underlying logic system $\mathbf{L}$ is an extension of classical logic allows us to make a clear distinction between boolean and monolithic formulae, avoiding double parsing and reconstructing the boolean formulae and its semantics in the temporalised system $\mathbf{T(L)}$. If we were to temporalise a logic system that is not an extension of classical logic, or any system in which we do not have the notion of satisfiability, only validity, we could consider all its formulae as being monolithic. The problem would then be the different semantics of the boolean connectives in the underlying system and in the upper-level (classical) temporal system, if those symbols are identical in both systems. The solution would be renaming the boolean connectives, say, in the underlying system. The applications of such a hybrid logic system are not clear so, to avoid extra difficulties in the results we are going to prove, we will stick to the constraint on $\mathbf{L}$ being an extension of classical logic.

2.3. The correspondence mapping

We are now going to relate the temporalised logic system $\mathbf{T(L)}$ with the original S, U-temporal logic used as a base for the temporalisation process. Consider $\mathcal{P}$, a denumerably infinite set of propositional letters, and let $\mathbf{S,U}$ be the propositional temporal logic system induced by $\mathcal{P}$. The following defines a relationship between a temporalised language $\mathcal{L}_{\mathbf{T(L)}}$ and a propositional temporal language $\mathcal{L}_{\mathbf{S,U}}$.

DEFINITION 2.7. **The correspondence mapping**

Consider an enumeration $p_1, p_2, \ldots$, of elements of $\mathcal{P}$ and consider an enumeration $\alpha_1, \alpha_2, \ldots$, of formulae in $ML_{\mathbf{L}}$. The *correspondence mapping*

$\sigma : \mathcal{L}_{\mathbf{T(L)}} \to \mathcal{L}_{\mathbf{S,U}}$ is given by:

$$
\begin{aligned}
\sigma(\alpha_i) &= p_i \text{ for every } \alpha_i \in ML_{\mathbf{L}}, i = 1, 2 \dots \\
\sigma(\neg\alpha) &= \neg\sigma(\alpha) \\
\sigma(\alpha \wedge \beta) &= \sigma(\alpha) \wedge \sigma(\beta) \\
\sigma(S(\alpha, \beta)) &= S(\sigma(\alpha), \sigma(\beta)) \\
\sigma(U(\alpha, \beta)) &= U(\sigma(\alpha), \sigma(\beta))
\end{aligned}
$$

□

The following is the *correspondence lemma*.

LEMMA 2.1. *The correspondence mapping is a bijection*

PROOF. By two straighforward structural inductions we can prove that σ is both injective and surjective. Details are omitted. □

As a consequence, we can always refer to an element Q of $\mathcal{L}_{\mathbf{S,U}}$ as $\sigma(\alpha)$, because there is guaranteed to be a unique $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$ such that α is mapped into Q by σ. We can then establish a connection between consistent formulae in $\mathbf{T(L)}/\mathcal{K}$ and in $\mathbf{S,U}/\mathcal{K}$.

LEMMA 2.2. *If α is $\mathbf{T(L)}$-consistent then $\sigma(\alpha)$ is $\mathbf{S,U}$-consistent.*

PROOF. Suppose $\sigma(\alpha)$ is inconsistent. Since all axioms and inference rules in $\mathbf{S,U}/\mathcal{K}$ are also in $\mathbf{T(L)}/\mathcal{K}$, the derivation of $\vdash_{\mathbf{S,U}} \sigma(\alpha) \to \bot$ can be imitated to derive $\vdash_{\mathbf{T(L)}} \alpha \to \bot$, which contradicts α being $\mathbf{T(L)}$-consistent. □

The results above are very useful for the proof of completeness and decidability of $\mathbf{T(L)}$.

2.4. Completeness of $\mathbf{T(L)}$

We are going to show here that whenever there exists complete axiomatisation for $\mathbf{S,U}/\mathcal{K}$ and for $\mathbf{L}$, where $\mathcal{K} \subseteq \mathcal{K}_{lin}$ is any linear class of flows of time, then the temporalised logic system $\mathbf{T(L)}/\mathcal{K}$ is also complete.

The strategy of the completeness proof is illustrated in Figure 3. We prove the completeness of $\mathbf{T(L)}/\mathcal{K}$ indirectly by transforming a consistent formula of $\mathbf{T(L)}$ and then mapping it into a consistent formula of $\mathbf{S,U}$. Completeness of $\mathbf{S,U}/\mathcal{K}$ is used to find a model for the mapped formula that is used to construct a model for the original $\mathbf{T(L)}$ formula.

The transformation function ε is introduced to deal with the differences between deductions in $\mathbf{S,U}$ and $\mathbf{T(L)}$ due to the presence of the inference rule **Preserve** in $\mathbf{T(L)}$. This inference rule states that theorems in $\mathbf{L}$ are also theorems in $\mathbf{T(L)}$. The model theoretic counterpart of this property that valid formulae in $\mathbf{L}$ are also valid in $\mathbf{T(L)}$. The idea behind the transformation ε is to extract "valid and contradictory content" that formulae

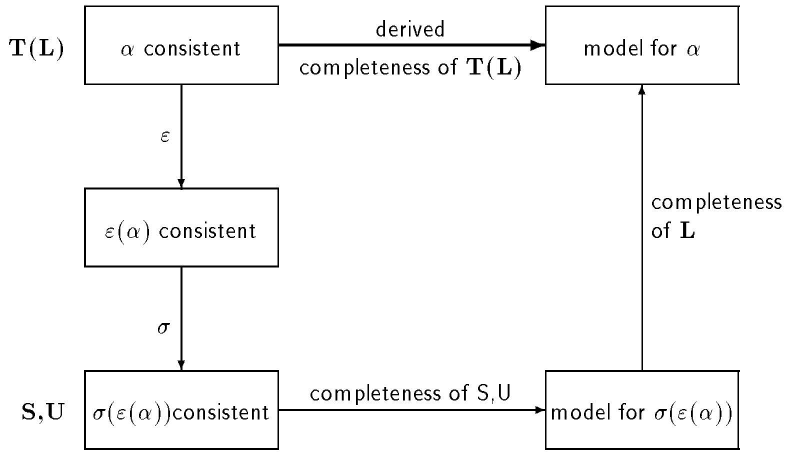


Fig. 3. Strategy for the proof of completeness

of **T(L)** may have due to the validity or unsatisfiability of some set of its subformulae in **L**.

DEFINITION 2.8. **The transformations η and ε**

Given a formula $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$, consider the following sets:

$$Lit(\alpha) = Mon(\alpha) \cup \{\neg\beta \mid \beta \in Mon(\alpha)\}$$
$$Inc(\alpha) = \{\bigwedge \Gamma \mid \Gamma \subseteq Lit(\alpha) \text{ and } \Gamma \vdash_{\mathbf{L}} \bot\}$$

where $Mon(\alpha)$ is the set of maximal monolithic subformulae of α. We define then the operator $\Box$ (always) and the formulae $\eta(\alpha)$ and $\varepsilon(\alpha)$:

$$\Box\beta = \beta \wedge G\beta \wedge H\beta$$
$$\eta(\alpha) = \bigwedge_{\beta \in Inc(\alpha)} \Box\neg\beta$$
$$\varepsilon(\alpha) = \alpha \wedge \eta(\alpha)$$

$\Box$

Since $\eta(\alpha)$ is a theorem of **T(L)**, we have the following lemma.

LEMMA 2.3. $\vdash_{\mathbf{T(L)}} \varepsilon(\alpha) \leftrightarrow \alpha$

If $\mathcal{K}$ is a subclass of linear flows of time, we also have the following property:

LEMMA 2.4. *Let $\mathcal{M}_{\mathbf{S,U}}$ be a temporal model over $\mathcal{K} \subseteq \mathcal{K}_{lin}$ such that for some $o \in T$, $\mathcal{M}_{\mathbf{S,U}}, o \models \sigma(\Box\alpha)$. Then, for every $t \in T$, $\mathcal{M}_{\mathbf{S,U}}, t \models \sigma(\Box\alpha)$.*

Therefore, if some subset of $Lit(\alpha)$ is inconsistent, the transformed formula $\varepsilon(\alpha)$ puts that fact in evidence so that, when σ maps it into $\mathbf{S,U}$, inconsistent subformulae will be mapped into falsity.

To prove the completeness of $\mathbf{T(L)}/\mathcal{K}$ given the completeness of $\mathbf{S,U}/\mathcal{K}$, we fix an α and assume it is a $\mathbf{T(L)}$-consistent formula. We have then to construct a model for α over $\mathcal{K}$.

By lemma 2.3, $\varepsilon(\alpha)$ is $\mathbf{T(L)}$-consistent and, by Lemma 2.2, $\sigma(\varepsilon(\alpha))$ is $\mathbf{S,U}$-consistent. Then, by the completeness of $\mathbf{S,U}/\mathcal{K}$, there exists a model $\mathcal{M}_{\mathbf{S,U}} = (T, <, h)$ with $(T, <) \in \mathcal{K}$ such that for some $o \in T$, $\mathcal{M}_{\mathbf{S,U}}, o \models \sigma(\varepsilon(\alpha))$.

For every $t \in T$, define $G_\alpha(t)$:

$$G_\alpha(t) = \{\beta \in Lit(\alpha) \mid \mathcal{M}_{\mathbf{S,U}}, t \models \sigma(\beta)\}$$

LEMMA 2.5. *If α is $\mathbf{T(L)}$-consistent, then for every $t \in T$, $G_\alpha(t)$ is finite and $\mathbf{L}$-consistent.*

PROOF. Since $Lit(\alpha)$ is finite, $G_\alpha(t)$ is finite for every t. Suppose $G_\alpha(t)$ is inconsistent for some t, then there exist $\{\beta_1, \ldots, \beta_n\} \subseteq G_\alpha(t)$ such that $\vdash_{\mathbf{L}} \bigwedge \beta_i \rightarrow \bot$. So $\bigwedge \beta_i \in Inc(\alpha)$ and $\Box\neg(\bigwedge \beta_i)$ is one of the conjuncts of $\varepsilon(\alpha)$. Applying Lemma 2.4 to $\mathcal{M}_{\mathbf{S,U}}, o \models \sigma(\varepsilon(\alpha))$ we get that for every $t \in T$, $\mathcal{M}_{\mathbf{S,U}}, t \models \neg(\bigwedge \sigma(\beta_i))$ but by, the definition of G_α, $\mathcal{M}_{\mathbf{S,U}}, t \models \bigwedge \sigma(\beta_i)$, which is a contradiction. $\Box$

We are finally ready to prove the completeness of $\mathbf{T(L)}/\mathcal{K}$.

THEOREM 2.3. **(Completeness for $\mathbf{T(L)}$)** *If the logical system $\mathbf{L}$ is complete and $\mathbf{S,U}/\mathcal{K}$ is complete over a subclass of linear flows of time $\mathcal{K} \subseteq \mathcal{K}_{lin}$, then the logical system $\mathbf{T(L)}/\mathcal{K}$ is complete over $\mathcal{K}$.*

PROOF. Assume that α is $\mathbf{T(L)}$-consistent. By Lemma 2.5, we have $(T, <) \in \mathcal{K}$ and associated to every time point in T we have a finite and $\mathbf{L}$-consistent set $G_\alpha(t)$. By (weak) completeness of $\mathbf{L}$, every $G_\alpha(t)$ has a model, so we define the temporalised valuation function g:

$$g(t) = \{\mathcal{M}^t_{\mathbf{L}} \mid \mathcal{M}^t_{\mathbf{L}} \text{ is a model of } G_\alpha(t)\}$$

Consider the model $\mathcal{M}_{\mathbf{T(L)}} = (T, <, g)$ over $\mathcal{K}$. By structural induction over β, we show that for every β that is a subformula of α and for every time point t,

$$\mathcal{M}_{\mathbf{S,U}}, t \models \sigma(\beta) \text{ iff } \mathcal{M}_{\mathbf{T(L)}}, t \models \beta$$

We show only the basic case, $\beta \in Mon(\alpha)$. Suppose $\mathcal{M}_{\mathbf{S,U}}, t \models \sigma(\beta)$; then $\beta \in G_\alpha(t)$ and $\mathcal{M}^t_{\mathbf{L}} \models \beta$, and hence $\mathcal{M}_{\mathbf{T(L)}}, t \models \beta$. Suppose $\mathcal{M}_{\mathbf{T(L)}}, t \models \beta$ and assume $\mathcal{M}_{\mathbf{S,U}}, t \models \neg\sigma(\beta)$; then $\neg\beta \in G_\alpha(t)$ and $\mathcal{M}^t_{\mathbf{L}} \models \neg\beta$, which contradicts $\mathcal{M}_{\mathbf{T(L)}}, t \models \beta$; hence $\mathcal{M}_{\mathbf{S,U}}, t \models \sigma(\beta)$. The inductive cases are straightforward and details are omitted.

So, $\mathcal{M}_{\mathbf{T(L)}}$ is a model for α over $\mathcal{K}$ and the proof is finished. □

Theorem 2.3 gives us sound and complete axiomatisations for **T(L)** over many interesting classes of flows of time, such as the class of all linear flows of time, $\mathcal{K}_{lin}$, the integers, $\mathbb{Z}$, and the reals, $\mathbb{R}$. These classes are, in their **S,U** versions, decidable and the corresponding decidability of **T(L)** is dealt in Section 3. Integer and real flows of time also have the separation property, which is discussed in Section 5.

3. The Decidability of T(L) and its Complexity

The main goal of this section is to show that, if the logic system **L** is decidable and the logic system **S,U** is decidable over $\mathcal{K} \subseteq \mathcal{K}_{lin}$, then the logic system **T(L)** is also decidable over $\mathcal{K}$. We assume throughout this section that **S,U**$/\mathcal{K}$ is complete.

DEFINITION 3.1. **Decidability of a Logic System**

A logic system **L** is said to be *decidable* if there exists an algorithm (a decision procedure) that, for every formula $\alpha \in \mathcal{L}_{\mathbf{L}}$, outputs "yes" if α is a theorem in the logic system **L** and "no" otherwise. □

There are several results of decidability of **S,U** over several linear classes of flows of time, among which the class $\mathcal{K}_{lin}$ of all linear flows of time (Burgess 1984), the integer and the real flows of time, (Burgess and Gurevich 1985).

As in the proof of completeness, we are going to prove the decidability result using the correspondence mapping σ and the transformation η. Recall Definition 2.8, in which the sets $Mon(\alpha)$, $Lit(\alpha)$ and $Inc(\alpha)$ were all finite, so that we have the following result about $\eta(\alpha)$.

LEMMA 3.1. *For any $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$, if the logic system* **L** *is decidable then there exists an algorithm for constructing $\eta(\alpha)$.*

The relationship between **T(L)** and **S,U** that we need to prove the decidability of **T(L)** is the following:

LEMMA 3.2. *Over a linear flow of time, for every $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$,*

$\vdash_{\mathbf{T(L)}} \alpha$ *iff* $\vdash_{\mathbf{S,U}} \sigma(\eta(\alpha) \to \alpha)$.

PROOF. The *if* case comes trivially from the definition of $\vdash_{\mathbf{T(L)}}$. For the *only if* part, suppose $\vdash_{\mathbf{T(L)}} \alpha$. We prove by induction on the deduction of α that $\vdash_{\mathbf{S,U}} \sigma(\eta(\alpha) \to \alpha)$.

Basic cases:

1. α is obtained using the inference rule **Preserve**. Then $\eta(\alpha) = \neg\neg\alpha$ and $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\neg\neg\alpha \to \alpha)$.
2. α is obtained using the inference rule **Subst**. Suppose α was obtained by substituting p_i by β_i in some axiom **A**. Then $\vdash_{\mathbf{S},\mathbf{U}} \alpha$ can be obtained by substituting $\sigma(p_i)$ by $\sigma(\beta_i)$ in axiom **A**.

Inductive cases:

1. $\alpha = G\beta$ is obtained using the inference rule **TG**. Note that $\eta(\alpha) = \eta(\beta)$. Then
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\alpha)) \to \sigma(\beta)$ by induction hypothesis
   $\vdash_{\mathbf{S},\mathbf{U}} G(\sigma(\eta(\alpha)) \to \sigma(\beta))$ by **T(G)**
   $\vdash_{\mathbf{S},\mathbf{U}} G(\sigma(\eta(\alpha))) \to \sigma(\alpha)$ by temporal logic and $\alpha = G\beta$
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\alpha)) \to G(\sigma(\eta(\alpha)))$by the definition of η and $\mathcal{K}$ linear
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\alpha) \to \alpha)$ from the two previous lines
   Similarly for $\alpha = H\beta$.
2. α is obtained from β and $\beta \to \alpha$ by **MP**. Then
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\beta)) \to \sigma(\beta)$ by induction hypothesis
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\beta \to \alpha)) \to \sigma(\beta \to \alpha)$by induction hypothesis
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\beta \to \alpha)) \to \sigma(\eta(\beta))$ by the definition of η
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\beta \to \alpha)) \to \sigma(\beta)$ from the 3^{rd} and 1^{st} lines
   $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\beta \to \alpha)) \to \sigma(\alpha)$ from the 4^{th} and 2^{nd} lines
   Let p be a proposition that occurs in $\sigma(\beta)$ but not in $\sigma(\alpha)$. If we eliminate from $\sigma(\eta(\alpha \to \beta))$ all the conjuncts in which p occurs, obtaining $\sigma(\gamma)$, using the completeness of **S,U**/$\mathcal{K}$ we can get $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\gamma) \to \sigma(\alpha)$. If we do that for all such propositions, we end up with $\vdash_{\mathbf{S},\mathbf{U}} \sigma(\eta(\alpha) \to \alpha)$.

□

THEOREM 3.1. **(Decidability of T(L))** *If* **L** *is a decidable logic system, and* **S,U** *is decidable over $\mathcal{K} \subseteq \mathcal{K}_{lin}$, then the logic system* **T(L)** *is also decidable over $\mathcal{K}$.*

PROOF. Consider $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$. Since **L** is decidable, by Lemma 3.1 there is an algorithmic procedure to build $\eta(\alpha)$. Since σ is a recursive function, we have an algorithm to construct $\sigma(\eta(\alpha) \to \alpha)$, and due to the decidability of **S,U** over $\mathcal{K}$, we have an effective procedure to decide if it is a theorem or not. Since $\mathcal{K}$ is linear, by Lemma 3.2 this is also a procedure for deciding whether α is a theorem or not. □

Once we have a decidability result, the next natural question is about the complexity of the decision procedure. We briefly discuss here an upper bound for the complexity analysis. Let N be the number of (boolean and modal) connectives in a formula, let the complexity of the decision procedure in **L** be $\mathcal{O}(f_{\mathbf{L}}(N))$ and in **S,U** be $\mathcal{O}(f_{\mathbf{S},\mathbf{U}}(N))$. The decision procedure for **T(L)** as given by the proof above consists of basically two steps:

1. constructing $\eta(\alpha)$;

2. deciding whether $\sigma(\eta(\alpha) \to \alpha)$ is a theorem or not;

The construction of $\eta(\alpha)$ involves generating all subsets of $Lit(\alpha)$ and applying the decision procedure for each subset, therefore its complexity is $\mathcal{O}(2^N \times f_{\mathbf{L}}(N))$. The second step is dominated by the decision procedure of **S,U** since the application of σ can be done in polynomial time; in the worst case, when all tests in **L** succeed, the size of $\eta(\alpha)$ is $\mathcal{O}(2^N)$ and therefore the decision is $\mathcal{O}(f_{\mathbf{S,U}}(2^N))$. So an upper bound for the decision procedure for $\mathbf{T(L)}$ is given by the dominating term of $\mathcal{O}(2^N \times f_{\mathbf{L}}(N))$ and $\mathcal{O}(f_{\mathbf{S,U}}(2^N))$. As for a lower bound for the decision procedure of $\mathbf{T(L)}$, it cannot be any lower than the highest of the lower bounds for **S,U** and **L**.

4. Conservativeness of T(L)

Conservativeness can be easily derived from a the soundness of **S,U** and the completeness of **L**, without any assumptions on the flow of time.

DEFINITION 4.1. **Conservative extension**

A logic system $\mathbf{L_1}$ is an *extension* of a logic system $\mathbf{L_2}$ if $\mathcal{L}_{\mathbf{L_2}} \subseteq \mathcal{L}_{\mathbf{L_1}}$ and if $\vdash_{\mathbf{L_2}} \alpha$ then $\vdash_{\mathbf{L_1}} \alpha$. A logic $\mathbf{L_1}$ is a *conservative extension* of $\mathbf{L_2}$ if it is an extension of $\mathbf{L_2}$ such that if $\alpha \in \mathcal{L}_{\mathbf{L_2}}$, then $\vdash_{\mathbf{L_1}} \alpha$ only if $\vdash_{\mathbf{L_2}} \alpha$. □

We know that all complete **S,U** are conservative extensions of predicate logic **PL**. Clearly, $\mathbf{T(L)}$ is an extension of **L**. We prove that it is also conservative.

THEOREM 4.1. **(Conservativeness of T(L))** *Let* **L** *be a complete logic system and* **S,U** *be sound over* $\mathcal{K}$. *The logic system* $\mathbf{T(L)}$ *is a conservative extension of* **L**.

PROOF. Let $\alpha \in \mathcal{L}_{\mathbf{L}}$ such that $\vdash_{\mathbf{T(L)}} \alpha$. Suppose by contradiction that $\not\vdash_{\mathbf{L}} \alpha$, so by completeness of **L**, there exists a model $\mathcal{M}_{\mathbf{L}}$ such that $\mathcal{M}_{\mathbf{L}} \models \neg\alpha$. We construct a temporalised model $\mathcal{M}_{\mathbf{T(L)}} = (T, <, g)$ by making $g(t) = \mathcal{M}_{\mathbf{L}}$ for all $t \in T$. $\mathcal{M}_{\mathbf{T(L)}}$ clearly contradicts the soundness of $\mathbf{T(L)}$ and therefore that of **S,U**, so $\vdash_{\mathbf{L}} \alpha$. □

5. Separation over the Added Dimension

The separation property of the S, U-temporal logic allows us to rewrite any temporal formula into a conjunction of formulae of the form

past formula and present formula $\to$ future formula.

Once a formula is in the format above, it can be imperatively interpreted against a partial temporal model according to (Gabbay 1987), so that if the antecedent holds in the past and present in the model, then we must execute the consequent in the future so as to make the formula true in the model. The imperative interpretation of a formula (also called the execution of a temporal specification) is based on an asymmetric view of the flow of time; in a symmetric view of time, whenever the antecedent is true in the past and present, we could either make the consequent true in the future or we could try to falsify the antecedent itself, in both cases maintaining the validity of the temporal specification. In this asymmetric view of time, we discard the latter possibility and remain with the former as the only possibility for the execution of a temporal specification.

In this section we want to extend this imperative interpretation of a temporal formula over a logic system **L** so that, after temporalising **L** over a flow of time that is like the integers or reals, we can execute temporal specifications in **T(L)**. The concept of a separated formula is based on the notion of a pure formula, so we present the definitions of pure formula and separated formula for the **S,U** logic.

DEFINITION 5.1. **Pure formulae in S,U**

1. A *pure present formula* is a boolean combination of propositional letters.
2. A *pure past formula* is a boolean combination of formulae of the form $S(\alpha, \beta)$ where α and β are either pure present or pure past formulae.
3. A *pure future formula* is a boolean combination of formulae of the form $U(\alpha, \beta)$ where α and β are either pure present or pure future formulae.

A *separated formula* is a formula that is a boolean combination of pure formulae only. □

Once we have a separated formula, it can be brought to a conjunctive normal form, i.e. a conjunction of disjuncts, so that each conjunct can be finally brought to the form:

pure-present and pure-past $\rightarrow$ pure-future.

The following is the basic result about separation over the integers.

THEOREM 5.1. **(Separation Theorem)** *For any formula* $A \in \mathcal{L}_{\mathbf{S},\mathbf{U}}$ *there exists a separated formula* $B \in \mathcal{L}_{\mathbf{S},\mathbf{U}}$ *such that* A *is equivalent to* B *over an integer-like flow of time.*

A proof of the separation theorem can be found in (Gabbay 1987; Gabbay 1991c). It also holds for the reals.

The generalisation of pure formula for a temporalised logic system **T(L)** is given below.

DEFINITION 5.2. **Pure temporalised formulae**

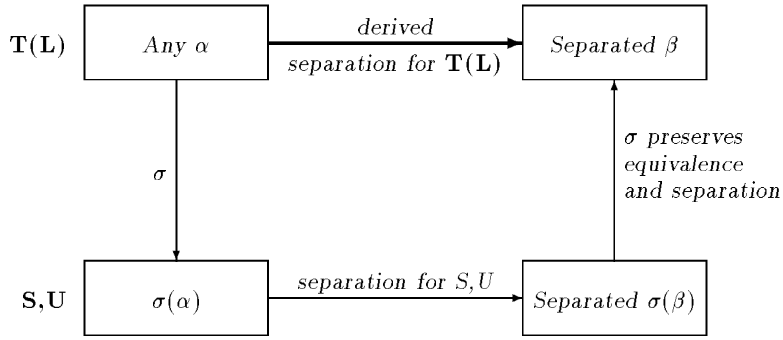


Fig. 4. Separation of $\mathbf{T(L)}$-formulae via separation of $\mathbf{S,U}$-formulae

1. every formula $\alpha \in \mathcal{L}_{\mathbf{L}}$ is a *pure present temporalised formula.*
2. A *pure past temporalised formula* is a boolean combination of formulae of the form $S(\alpha, \beta)$ where α and β are either pure present or pure past temporalised formulae.
3. A *pure future temporalised formula* is a boolean combination of formulae of the form $U(\alpha, \beta)$ where α and β are either pure present or pure future temporalised formulae.

A *separated temporalised formula* is a boolean combination of pure formulae of $\mathbf{T(L)}$. □

EXAMPLE 5.1. *Temporalising a modal logic of belief*

Suppose $\mathbf{L}$ is the modal logic system of belief, with the modal operator B. Here are some examples of pure temporalised formulae in $\mathbf{T(L)}$:

1. Pure present: $Bp \rightarrow p$, $\neg(p \wedge \neg p)$, and any other formula of the logic $\mathbf{L}$.
2. Pure past: $P(Bp) \rightarrow S(Bp, \neg p)$.
3. Pure future: $F(Bp) \rightarrow \neg Fp \vee G(Bp \rightarrow \neg p)$.

In order to prove the separation theorem for the temporalised logic $\mathbf{T(L)}$ we will use the correspondence mapping. The basic strategy of the proof is illustrated in figure 4.

The following is a helpful result that will lead us to the proof of separation for the temporalised logic $\mathbf{T(L)}$.

LEMMA 5.1. *Let σ be a correspondence mapping between $\mathcal{L}_{\mathbf{T(L)}}$ and $\mathcal{L}_{\mathbf{S,U}}$. $\sigma(\alpha)$ is a separated formula in the logic* $\mathbf{S,U}$ *iff α is a separated formula in* $\mathbf{T(L)}$.

PROOF. From the definition of the correspondence mapping it follows that if α is a boolean combination of $\alpha_1, \ldots \alpha_n \in \mathcal{L}_{\mathbf{T(L)}}$ then $\sigma(\alpha)$ is a boolean combination of $\sigma(\alpha_1), \ldots, \sigma(\alpha) \in \mathcal{L}_{\mathbf{S,U}}$. The converse is also true since σ is a bijection.

Therefore, fo show that α is separated in $\mathbf{T(L)}$ iff $\sigma(\alpha)$ is separated in $\mathbf{S,U}$, all we have to do is to prove that $\sigma(\alpha)$ is a pure formula iff α is a pure formula. We show the proof for the *only if* case; the *if* part is completely analogous.

Suppose $\sigma(\alpha)$ is a pure present, then it is a boolean combination of propositional letters. Therefore α is a boolean combination of monolithic formulae of $\mathbf{L}$, therefore α is a formula of $\mathbf{L}$, and pure present in $\mathbf{T(L)}$.

Suppose $\sigma(\alpha)$ is pure past, then it is a boolean combination of formulae in $\mathcal{L}_{\mathbf{S,U}}$ of the form $S(\sigma(\beta), \sigma(\gamma))$ where $\sigma(\beta)$ and $\sigma(\gamma)$ are pure present or pure past. Therefore α must be a boolean combination of formulae in $\mathcal{L}_{\mathbf{T(L)}}$ of the form $S(\gamma, \delta)$, where γ and δ are, by induction hypothesis, either pure present and pure past. Therefore α is a pure past formula in $\mathcal{L}_{\mathbf{T(L)}}$.

Suppose $\sigma(\alpha)$ is pure future, then by an argument analogous to the previous case, α is a pure future formula. Therefore we have proved that if $\sigma(\alpha)$ is a pure formula in $\mathcal{L}_{\mathbf{S,U}}$, α is a pure formula in $\mathcal{L}_{\mathbf{T(L)}}$. $\square$

THEOREM 5.2. **(Separation Theorem for $\mathbf{T(L)}$)** *If α is any formula in $\mathcal{L}_{\mathbf{T(L)}}$, then there exists a separated formula $\beta \in \mathcal{L}_{\mathbf{T(L)}}$ such that β is equivalent to α over an integer-like flow of time.*

PROOF. All we have to do is to prove that if α and β are formulae of $\mathbf{T(L)}$ and $\vdash_{\mathbf{S,U}} \sigma(\alpha) \leftrightarrow \sigma(\beta)$ then $\vdash_{\mathbf{T(L)}} \alpha \leftrightarrow \beta$. In fact, since all axioms and inference rules of $\mathbf{S,U}$ also belong to $\mathbf{T(L)}$, the deduction of $\vdash_{\mathbf{S,U}} \sigma(\alpha) \leftrightarrow \sigma(\beta)$ also leads to $\vdash_{\mathbf{T(L)}} \alpha \leftrightarrow \beta$.

Let then α be any formula of $\mathbf{T(L)}$. From the separation theorem of $\mathbf{S,U}$, we know that there exist a separated β, such that $\vdash_{\mathbf{S,U}} \sigma(\alpha) \leftrightarrow \sigma(\beta)$ and $\sigma(\beta)$ is separated. So by Lemma 5.1, β is also a separated formula equivalent to α. $\square$

Once we have the separation property for the temporalised system $\mathbf{T(L)}$, we can rewrite any temporalised formula into a separated equivalent one of the form

$$\text{pure temporalised past and present} \ \rightarrow \ \text{pure temporalised future.}$$

The imperative interpretation of such a formula is the following. If the antecedent holds in past and present models of the logic system $\mathbf{L}$, then we execute the temporalised formula by constructing a future model (or a series or future models) of $\mathbf{L}$ so as to make the consequent true.

Since the separation property also holds for a real flow of time, the proof above can be trivially adapted to a real flow of time. Note that the separation property for the temporalised system was obtained without any assumptions on the underlying logic system **L**, as opposed to the results of soundness, completeness and decidability, all of which depend on whether the property holds for the underlying logic system **L**.

6. Temporalising First-Order Logic

In this section we examine in more detail the addition of a temporal dimension to a first-order language. We will consider a first-order language with the quantifier $\forall$, an equality symbol $=$, a countable set of variables $X = \{x_1, x_2, \ldots\}$, a countable set of predicate symbols $P = \{p_1, p_2, \ldots\}$ such that every predicate symbol has an associated natural number $n > 0$, called its arity, a set C of constant symbols and a set F of functional symbols; C and F are possibly empty. The quantifier $\exists$ can be defined in the normal way as $\exists = \neg\forall\neg$. A term is either a variable, a constant symbol or an n-ary function symbol applied to n terms. The notion of the set of free variables of a formula is the usual one. A sentence is a formula with no free variables.

A first-order domain $\mathcal{D}$ is a non-empty set. An interpretation $\mathcal{I}$ is a mapping that associates, for every constant in the language an element in the domain, and for every n-place predicate symbol an n-ary relation over $\mathcal{D}^n$. An assignment function $\mathcal{A}$ is a mapping that associates every variable with an element of the domain. A first-order model is a pair $\mathcal{M} = (\mathcal{D}, \mathcal{I})$. If t is a term, $[\![t]\!]^{\mathcal{I},\mathcal{A}} \in \mathcal{D}$ represents its extension over the domain $\mathcal{D}$ under interpretation $\mathcal{I}$ and assignment $\mathcal{A}$. The semantics of a first-order language is then defined in the usual way, where $\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \alpha$ reads "$\mathcal{M}$ is a model of the formula α under assignment $\mathcal{A}$":

$\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} p_i(t_1, \ldots, t_n)$ iff $\langle [\![t_1]\!]^{\mathcal{I},\mathcal{A}}, \ldots, [\![t_n]\!]^{\mathcal{I},\mathcal{A}} \rangle \in \mathcal{I}(p_i)$, for all n-ary predicate symbols $p_i \in P$.
$\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \neg\alpha$ iff $\mathcal{M}, \mathcal{A} \not\models_{\mathbf{FOL}} \alpha$.
$\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \alpha \wedge \beta$ iff $\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \alpha$ and $\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \beta$.
$\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} t_1 = t_2$ iff $[\![t_1]\!]^{\mathcal{I},\mathcal{A}} = [\![t_2]\!]^{\mathcal{I},\mathcal{A}}$.
$\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \forall x \alpha$ iff for any assignment $\mathcal{A}'$ which agrees with $\mathcal{A}$, except possibly on variable x, $\mathcal{M}, \mathcal{A}' \models_{\mathbf{FOL}} \alpha$.

We say that α has a model $\mathcal{M}$, and write $\mathcal{M} \models_{\mathbf{FOL}} \alpha$, if $\mathcal{M}, \mathcal{A} \models_{\mathbf{FOL}} \alpha$ for all assignments $\mathcal{A}$ (this is always the case when α is a sentence).

The derivability relation, $\vdash_{\mathbf{FOL}}$, can be any of the existent ones for first-order logic. It can be an axiomatic system, but it need not.

Since in first-order logic we have a basic distinction between sentences and ordinary formulae, we have to consider both cases of adding a temporal dimension to monolithic sentences and to monolithic formulae in general.

Element	*Fixed*	*Variable*
Domain	constant domains	variable domains
Constant and Functional Symbols	rigid	non-rigid or flexible
Predicate Symbols	rigid	non-rigid or flexible
Assignment	global	local

TABLE I
Degrees of freedom in temporalising first-order models.

6.1. Temporalising First-Order Sentences

If we temporalise first-order sentences, we have no problems in following the methodology we have developed so far. We first identify the monolithic sentences as those that are not in the format $\alpha \wedge \beta$ or $\neg\alpha$. For instance, $\forall x p(x)$ and $\forall x \neg(q(x) \wedge \neg q(x))$ are monolithic sentences, whereas $\exists x p(x)$ (implicit negation) and $\forall x p(x) \wedge \forall y \neg q(y)$ are boolean combinations. We then follow the procedure described in Section 2, obtaining the logic system **T(FOs)**. Note that in **T(FOs)** a temporal operator never occurs inside the scope of a quantifier.

The structure of the first-order models that compose the temporalised model deserves some special attention, since one model may differ form another in several different ways, as if we had various "degrees of freedom" in generating a temporalised version of first-order models. Those degrees of freedom are illustrated in Table I.

If all first-order models that compose a temporalised model $\mathcal{M}_{\mathbf{T(FOs)}}$ refer to the same domain, a constant domain assumption is satisfied; otherwise, we have varying domains. We may have rigid constant and rigid functional symbols, i.e. they have the same interpretation in every model of the temporalised structure; they are called non-rigid or flexible otherwise. A rigid predicate symbol has the same interpretation at all time; otherwise it is a flexible predicate symbol. And finally, the assignment function may be global, i.e. all variables are assigned the same domain element in all models of the temporalised structure (global assignments make sense only under a constant domain assumption); otherwise, it is a local assignment.

In fact, constant domains or rigid terms or predicates are not a consequence of the temporalisation; they are, actually, further assumptions on the temporalised first-order model made so as to impose some external intended meaning of adding a temporal dimension to a logic system. All the previously established results of soundness, completeness and separation are valid for unconstrained $\mathbf{T(FOs)}$; decidability is obviously not applicable.

Nevertheless, there is no quantification over the temporal operators in $\mathbf{T(FOs)}$, which means that the expressivity of this logic is clearly limited. In the following, we examine one step further in increasing this expressivity, while still keeping the original idea of adding a temporal dimension to a logic system.

6.2. Temporalising First-Order Formulae

We take now general monolithic first-order formulae as a basis for the addition of a temporal dimension, i.e. all first-order formulae that are not of the form $\neg\alpha$ or $\alpha \wedge \beta$. We generate thus the logic system $\mathbf{T(FOf)}$. Note that the language of $\mathbf{T(FOs)}$ is contained in the language of $\mathbf{T(FOf)}$.

The particular feature that distinguishes this system from all the previously considered systems is that, since we are considering first-order formulae that may contain free variables, monolithic formulae with free variables only have a defined semantics over a first-order model $\mathcal{M}_{\mathbf{FOL}}$ if a variable assignment function is provided, and the free variables of a first-order formula used to build a temporalised formula α remain free in α.

Therefore, while constructing a model for the system $\mathbf{T(FOf)}$, we must consider the existence of a *global assignment function*, $\mathcal{A}_g$, to cope with the free variables. A global assignment function makes sense only in a constant domain context, so we must have this assumption as well; we further assume that all terms are rigid. The effect of the global assignment $\mathcal{A}_g$ is to ground all the free variables of a temporalised formula α. Only the interpretation of predicate symbols changes among the models of $\mathbf{L}$ in the temporalised model structure. We write

$$\mathcal{M}_{\mathbf{T(FOf)}} \models \alpha \ \text{ iff } \ \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}_g \models \alpha \text{ for any } \mathcal{A}_g.$$

Since the construction of its temporalised model and inference system does not follow exactly the way other temporal systems were constructed, the results previously established of soundness, completeness and separation cannot be applied directly.

We know that the more expressive full first-order temporal logic has no possible finite axiomatisation over several useful classes of linear flows of time like $\{\mathbb{R}\}$, $\{\mathbb{Z}\}$ and $\{\mathbb{N}\}$, e.g. see (Garson 1984), but we do have a finite axiomatisation for $\mathbf{T(FOs)}$. The logic system $\mathbf{T(FOf)}$ has an intermediary expressive power and it can be shown that $\mathbf{T(FOf)}$ cannot be finitely axiomatised over linear flows of time that contain the natural numbers, although we will not do it here. Perhaps more interesting is that separation can be achieved for this logic through model theory.

Since the concept of separated formula is purely syntactic and does not depend on the model or the inference system, the definition of a separated temporalised formulae given by Definition 5.2 is also valid for $\mathbf{T(FOf)}$. For the same reasons, the definition of a correspondence mapping σ and the correspondence lemma 2.1 stating that σ is a bijection are also valid in $\mathbf{T(FOf)}$.

DEFINITION 6.1. **Corresponding Model**

Let $\mathcal{M}_{\mathbf{T(FOf)}} = (T, <, g)$ be a model of $\mathbf{T(FOf)}$, and let $\mathcal{A}$ be a global assignment. We construct the valuation function g_σ such that, for every time point $t \in T$ and for every propositional letter $p = \sigma(\alpha) \in \mathcal{P}$ we have

$$\sigma(\alpha) \in g_\sigma(t) \text{ iff } \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \alpha.$$

A model of the temporal logic system $\mathbf{S,U}$, $\mathcal{M}^{\sigma}_{\mathbf{S,U}} = (T, <, g_\sigma)$, is then called the *corresponding model* of $\mathcal{M}_{\mathbf{T(L)}}$ under the corresponding mapping σ and assignment $\mathcal{A}$. □

LEMMA 6.1. *If $\mathcal{M}^{\sigma}_{\mathbf{S,U}}$ is the corresponding model of $\mathcal{M}_{\mathbf{T(FOf)}}$ under σ and $\mathcal{A}$ then*

$$\mathcal{M}^{\sigma}_{\mathbf{S,U}}, t \models \sigma(\alpha) \textit{ iff } \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \alpha$$

for every $\alpha \in \mathcal{L}_{\mathbf{T(L)}}$ and for every $t \in T$.

PROOF. Straightforward by structural induction on α. □

THEOREM 6.1. **(Separation for T(FOf))** *For every $\alpha \in \mathcal{L}_{\mathbf{T(FOf)}}$ there exists a separated formula $\beta \in \mathcal{L}_{\mathbf{T(FOf)}}$ such that β is equivalent to α over an integer-like flow of time.*

PROOF. Let σ be a correspondence mapping and $\mathcal{A}$ an arbitrary global assignment . Consider a temporalised model $\mathcal{M}_{\mathbf{T(FOf)}} = (T, <, h)$, $(T, <) \in \mathbb{Z}$, and let $\mathcal{M}^{\sigma}_{\mathbf{S,U}} = (T, <, g_\sigma)$ be its correspondent model under σ and $\mathcal{A}$. By Lemma 6.1, we have

$$\mathcal{M}^{\sigma}_{\mathbf{S,U}}, t \models \sigma(\alpha) \text{ iff } \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \alpha \tag{1}$$

for every $\alpha \in \mathcal{L}_{\mathbf{T(FOf)}}$ and for every $t \in T$.

By the separation theorem for $\mathbf{S,U}$ we get that, for every formula $\sigma(\alpha) \in \mathcal{L}_{\mathbf{S,U}}$ there exists a separated formula $\sigma(\beta) \in \mathcal{L}_{\mathbf{S,U}}$ such that

$$\mathcal{M}^{\sigma}_{\mathbf{S,U}}, t \models \sigma(\alpha) \text{ iff } \mathcal{M}^{\sigma}_{\mathbf{S,U}}, t \models \sigma(\beta) \tag{2}$$

for all time points $t \in T$.

By Lemma 5.1, we have that the corresponding mapping preserves separation, i.e. β is a separated formula iff $\sigma(\beta)$ is a separated formula and, by application of (1)

$$\mathcal{M}^{\sigma}_{\mathbf{S},\mathbf{U}}, t \models \sigma(\beta) \text{ iff } \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \beta \tag{3}$$

for all time points $t \in T$.

Combining (1), (2) and (3) we get that, for every $\alpha \in \mathcal{L}_{\mathbf{T(FOf)}}$ there exists a separated $\beta \in \mathcal{L}_{\mathbf{T(FOf)}}$ such that, for all $t \in T$

$$\mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \alpha \text{ iff } \mathcal{M}_{\mathbf{T(FOf)}}, \mathcal{A}, t \models \beta \tag{4}$$

Since the assignment $\mathcal{A}$ was arbitrarily chosen and the separated β does not depend on the particular choice of $\mathcal{A}$, expression (4) holds for any global assignment $\mathcal{A}$, and separation for **T(FOf)** remains proved. □

We note that if we fix a current time, o, and a global assignment $\mathcal{A}_g$, we can apply the temporalisation process to the logic system **T(FOf)**, obtaining a two-dimensional temporal predicate system, $\mathbf{T}^2\mathbf{(FOf)}$, as a predicate version of the two-dimensional propositional system described in example 2.3.

7. Internalising the Temporal Dimension

There are three basic approaches to adding a temporal dimension to a logic system, namely:

1. The temporal operators approach.
2. The first-order internalisation of the temporal dimension.
3. A mixed approach combining the two approaches above.

Those three different approaches are discussed in detail in (Gabbay 1990) in the context of propositional temporal logic. The first approach is the one we have been following so far. Here we briefly present the other ones in the context of temporalised formulae.

Consider the temporalised first-order formula in **T(FOf)**

$$believed(x) \to F\ happens(x)$$

expressing that whatever is believed now will become true in the future. This statement could actually be completely coded in the original first-order language by adding a temporal argument to the predicates *believed* and *happens*. The resulting formulation would be

$$believed^*(t, x) \to \exists s(t < s \wedge happen^*(s, x)).$$

This process of getting rid of the temporal operators by adding a new temporal argument to the predicates plus some extra conditions on those arguments can be done systematically by an *internalisation function* $*$ defined inductively over the structure of a formula of $\mathbf{T(FOf)}$ and also taking as argument a reference time point, generating a two-sorted predicate formula, one sort over time and the other sort over domain elements. We call this process the *internalisation* of the temporal dimension. The internalisation of the temporal dimension is basically obtained by the standard translation of temporal logic into predicate logic, e.g. (Benthem 1983), with an extra argument to incorporate the temporal reference; this extra argument can be interpreted as the result of Quine's "eternalisation" of first-order sentences (Quine 1960).

In the internalised version it is necessary to incorporate a theory expressing the properties of the flow of time $\mathcal{K} = (T, <)$ to restore the deductive capability of temporal formulae. However, there are several flows of time over which there are complete temporal axiomatisations that are not definable in first-order logic, e.g. the integers and the reals.

Another way of getting to a first-order predicate logic approach to temporal logic, as proposed by Gabbay (1991b), is by mixing two predicate logic languages in the following way. Let $\mathbf{G}$ (for global) and $\mathbf{L}$ (for local) be two first-order languages. The two-sorted predicate language $\mathbf{L}^*_k(\mathbf{G})$ is the result of mixing the $\mathbf{G}$ and $\mathbf{L}$ (in our present notation it would be $\mathbf{G}(\mathbf{L}^*_k)$). If we consider the language $\mathbf{L}^*_1(\mathbf{G})$, then a formula of the form $P^*(t, x_1, \ldots, x_n)$ means that $P(x_1, \ldots, x_n)$ holds at time t. This language is the same language of the internalised temporal dimension system. But this approach gives us a way of creating an internalised logic system in a very similar way to that in which a temporalised system was created, i.e. as a result of putting two languages together. In fact, the original languages $\mathbf{G}$ and $\mathbf{L}$ can be seen as two linked languages "sharing variables" in the language $\mathbf{L}^*_1(\mathbf{G})$. One of the original languages, $\mathbf{G}$, has the exclusivity of dealing with temporal facts, as the upper-level S, U-temporal logic system, whereas the language $\mathbf{L}$ is responsible only for the local behaviour at each point in the flow of time.

The temporal operators approach to a temporalised formula can be seen as treating time points implicitly, always referring to a current time. The first-order internalisation refers explicitly to the points in the flow of time. A hybrid form of internalisation of the temporal dimension can be obtained by combining temporal operators with first-order internalised formulae, mixing the explicit reference with the implicit reference of time.

In the combined approach, every temporalised formula α is associated with a first-order atomic formula $holds(t, \alpha)$, where α is now treated as a first-order term, and the free variables of α are considered free in $holds(t, \alpha)$. A set of axioms is added to combine the $holds(t, \alpha)$ formulae with the first-order internalised formulae, for example:

$holds(t, \alpha) \leftrightarrow (\alpha)^*[t]$, for all monolithic $\alpha \in \mathcal{L}_{\mathbf{L}}$
$holds(t, \alpha \wedge \beta) \leftrightarrow holds(t, \alpha) \wedge holds(t, \beta)$
$holds(t, \alpha \wedge \beta) \leftrightarrow \exists s[s < t \wedge holds(s, \alpha) \wedge \forall u(s < u < t \rightarrow holds(u, \beta))]$
etc.

As in the internalised approach, in the combined approach we still have to provide axioms for the flow of time.

Conclusion

We have shown in this paper a way of composing an upper-level temporal logic system with a generic underlying logic system **L** and the resulting logic system **T(L)** was called the temporalisation of system **L**. We used the correspondence mapping method to prove soundness, completeness, decidability, conservativeness and separation for the temporalised logic system over linear flows of time. All those properties were initially properties of the temporal logic system. Many other properties remain to be analised, such as compactness, finite model property and interpolation among others; the properties discussed here over classes of linear flows of time remain to be expanded for all classes of flows of time.

We need by no means restrict the upper-level logic system to temporal logic. In fact, the temporalisation presented in this paper can be generalised to any propositional modal logic system **M** in the role of the upper-level logic system, so as to create a modalised logic system **M(L)**. Its language and inference system can be obtained following the method we used to derive the those of **T(L)**, based on the monolithic formulae of **L**. If the logic **L** has a possible world semantics, each possible world may be substituted by a model of **L**, so as to construct a model for the system **M(L)** in the same way a model was built for **T(L)**. The correspondence mapping method may then be used to study how the properties of the modal logic system **M** are preserved in the modalised logic system **M(L)**.

Acknowledgements

This work benefited from the comments of Jose Fiadero, Ian Hodkinson, Tony Hunter, Leonardo Lazarte, Mark Reynolds, Ben Strulo and the referee. The work was supported by the ESPRIT project under Basic Research Action 3096 (SPEC). Marcelo Finger was supported by the CAPES-Brazil, grant 1481/89.

References

Johan van Benthem. *The Logic of Time.* Reidel, Dordrecht, 1983.

J. P. Burgess and Yuri Gurevich. The Decision Problem for Linear Logic. *Notre Dame Journal of Formal Logic*, 26(2):566–582, April 1985.

John P. Burgess. Axioms for Tense Logic I: "Since" and "Until". *Notre Dame Journal of Formal Logic*, 23(4):367–374, October 1982.

John P. Burgess. Basic Tense Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 89–133. D. Reidel Publishing Company, 1984.

M. A. Casanova and A. L. Furtado. A Family of Temporal Languages for the Description of Transition Constraints. In *3rd Worksohp on Logical Bases for Databases*, Toulouse, France, December 1982.

Kit Fine and Gerhard Schurz. Transfer theorems for stratified multimodal logics. To appear, 1992.

Marcelo Finger. Handling Database Updates in Two-dimensional Temporal Logic. *J. of Applied Non-Classical Logic*, 2(2):201–224, 1992.

Dov M. Gabbay and Ian M. Hodkinson. An axiomatization of the temporal logic with Until and Since over the real numbers. *Journal of Logic and Computation*, 1(2):229–259, 1990.

Dov M. Gabbay. The Declarative Past and the Imperative Future. In B. Banieqbal et al., editors, *Coloquium on Temporal Logic and Specifications — Lecture Notes in Computer Science 389*, Manchester, April 1987. Springer-Verlag.

Dov M. Gabbay. Temporal Logic, Tense or Non-tense — inaugural lecture at Imperial College, 17 May 1988. In R. Spencer-Smith and S. Torrance, editors, *Machinations — Computational Studies of Logic, Language and Cognition*. Ablex Publishing Co., 1990.

Dov M. Gabbay. Labelled Deductive Systems – Part I. Technical Report CIS–Bericht–90–22, Universität München, Centrum für Informations – und Sprachverarbeitung, February 1991.

Dov M. Gabbay. Modal and Temporal Logic Programming III — metalevel features in the object level. In L. F. del Cerro and M. Penttonen, editors, *Non-Classical Logic Programming*. Oxford University Press, 1991.

Dov M. Gabbay. *Temporal Logic — Mathematical Foundations and Computational Aspects*. To appear, 1991.

Dov M. Gabbay. Theoretical Foundations for Non-monotonic Reasoning Part 2: Structured Non-monotonic Theories. In *SCAI 91 – Third Scandinavian Conference on AI*, pages 19–40, May 1991.

Dov M. Gabbay. Fibred semantics and combinations of logics. Manuscript, Imperial College, 1992.

James W. Garson. Quantification in Modal Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 249–307. D. Reidel Publishing Company, 1984.

Joseph Y. Halpern and Yoram Moses. A Guide to the Modal Logics of Knowledge and Belief. In *Proceedings of the 9th International Joint Con-*

ference on Artificial Intelligence (IJCAI-85), pages 480–490, 1985.

Jaakko Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.

G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic.* Methuen, 1968.

K. Konolige. *A Deductive Model of Belief.* Research notes in Artificial Intelligence. Morgan Kaufmann, 1986.

Marcus Kracht and Frank Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

Willard Van Orman Quine. *Word and Object.* MIT Press, 1960.

Mark A. Reynolds. An Axiomatisation for Until and Since over the reals without the IRR Rule. *Studia Logica*, 51(2):165–194, 1992.

Richmond H. Thomason. Combinations of Tense and Modality. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 135–165. D. Reidel Publishing Company, 1984.

Ming Xu. On some U, S-Tense Logics. *Journal of Philosophical Logic*, 17:181–202, 1988.