A Modal Temporal Dynamic Logic
Doing the Deadline [†]

J. Scheerder [‡] R.J. Wieringa

Faculty of Mathematics and Computer science
Vrije Universiteit, De Boelelaan 1081a
1081 HV Amsterdam
Email: (js|roelw)@cs.vu.nl

August 3, 1997

Abstract

In this paper an investigation into the aspects of formal system specification that manifest themselves when considering both time and system dynamics is carried out. A logic to express temporal dynamic properties is developed.

[†]This work was partly supported by the EU under ESPRIT-IV WG 22704 ASPIRE.

[‡]This research is partly supported by the Netherlands Organization for Scientific Research (NWO), SION project 612-323-419.

Contents

1 Introduction

Complex systems in real life can be submitted to lots of different kinds of constraints. They may have to be proven *safe*, or *correct*; their *liveness* may have to be established, and so on. Many system properties are "timeless" in the sense that timing properties are of no (essential) importance when considering whether requirements are met or not. However, for many systems, timing behaviour *is* crucial – not only must an accurate calculation up to the umpteenth digit be performed, this must also be done within a very real, hard timeframe to be able to make course corrections for a ballistic missile that's underway. For that reason, we need to be able to express properties, behaviour, requirements and constraints in such a way that both the system dynamics – what state induces which other system state by which action – and the system timing can be dealt with. A logic that is rich enough to do so is developed in in this paper.

2 Goals

Unconventionally, we start by summing up some ideas about the kind of things deemed desirable to be able to write down. Later it will be examined how – if at all – these desiderata can be grounded in the usual formal fashion.

Naively, a system exhibits certain aspects that need to be captured in a system specification. A system may have discernable *states*, and these states can be related causally, coincidentally, temporally: and as a whole, the system exhibits interaction with an environment (and has an interface to it) and has a *dynamic* behaviour. If by now the reader is getting anxious because this seems a bit much, and very complex: rest assured, it is. Nevertheless, even if at first glance we may want more than seems feasible, it still indicates the kind of things worth looking into, as well as provide a useful insight into the way we want it done.

So what are typical things we would like to be able to express when describing an observed system, or writing a system specification? The general form seems to be that given a certain state of the system a certain other state can, or must, be reached, possibly within certain time bounds, and possibly with certain actions being undertaken or certain events occurring. A few suggestive examples, of which most will look familiar to the reader (with ϕ, ψ, ... propositions, and α, β, ... actions, and the rest interpreted in the obvious manner), along with some intuitive explanation:

- **next** ϕ – in the next state ϕ will be true;

- $\Diamond\phi$ – in a 'reachable' state ϕ holds;
- $\mathbf{F}\phi$ – at some point in the future ϕ will hold;
- ϕ **until** ψ – ϕ will hold at least until ψ does;
- ϕ **until** α – ϕ will hold at least until α is performed
- $[\alpha;\beta]\phi$ after α and β are done, ϕ is true.

In the following, we will establish the basic notational framework and it will be shown that a few operators suffice to handle quite a lot of the usual modal operators.

3 Propositional Calculus

We first define the language of Propositional Logic (PROP) $\mathcal{L}_{\mathsf{PROP}}$. Let $\mathcal{P}$ be a set of atomic propositional constant symbols containing $\bot$, i.e. $\{\bot, \mathrm{p}_0, \ldots, \mathrm{q}_0, \ldots, \ldots\}$. In the following, P, Q, ... are variables over $\mathcal{P}$, and ϕ, ψ, ... are variables over $\mathcal{L}_{\mathsf{PROP}}$, which is inductively defined as the smallest set such that:

- $\mathcal{P} \subset \mathcal{L}_{\mathsf{PROP}}$
- $\phi, \psi \in \mathcal{L}_{\mathsf{PROP}} \Rightarrow (\phi \rightarrow \psi) \in \mathcal{L}_{\mathsf{PROP}}$

We will omit outer parentheses leisurely, and we will introduce the convenient and familiar abbreviations[1] here: $\top$ ('true'), $\neg$ ('not'), $\vee$ ('or'), $\wedge$ ('and'), and $\leftrightarrow$ ('is equivalent to'):

- $\top \stackrel{\mathsf{df}}{=} \bot \rightarrow \bot$
- $\neg\phi \stackrel{\mathsf{df}}{=} \phi \rightarrow \bot$
- $\phi \vee \psi \stackrel{\mathsf{df}}{=} (\phi \rightarrow \bot) \rightarrow \psi$
- $\phi \wedge \psi \stackrel{\mathsf{df}}{=} (\phi \rightarrow (\psi \rightarrow \bot)) \rightarrow \bot$
- $\phi \leftrightarrow \psi \stackrel{\mathsf{df}}{=} ((\phi \rightarrow \psi) \rightarrow ((\psi \rightarrow \phi) \rightarrow \bot)) \rightarrow \bot$

[1] Cf. Goldblatt (1987).

An *interpretation* $\mathcal{I} \subseteq \mathcal{P}$ establishes truth values for atomic propositions, i.e. $\mathcal{I}$ is the set of true atomic propositions, and cannot contain $\bot$. We write $\mathcal{I} \models_{\mathsf{PROP}} \phi$ to indicate that a propositional formula $\phi \in \mathcal{L}_{\mathsf{PROP}}$ holds under interpretation $\mathcal{I}$. The $\models_{\mathsf{PROP}}$ relation is defined as follows:

- $\mathcal{I} \models_{\mathsf{PROP}} \mathrm{P} \Leftrightarrow \mathrm{P} \in \mathcal{I}$[2]
- $\mathcal{I} \models_{\mathsf{PROP}} \phi \rightarrow \psi \Leftrightarrow (\mathcal{I} \models_{\mathsf{PROP}} \phi \Rightarrow \mathcal{I} \models_{\mathsf{PROP}} \psi)$

The propositional abbreviations defined above have the usual straightforward semantics; we don't bother writing them down here, but will use them at our convenience, considering them to be defined in the usual way. Using this notion of validity under an interpretation, we define general validity: $\models_{\mathsf{PROP}} \phi$ iff $\mathcal{I} \models_{\mathsf{PROP}} \phi$ for all $\mathcal{I} \in \wp(\mathcal{P})$, i.e. there is no interpretation $\mathcal{I}$ under which ϕ does not hold.

Proof systems can be given for classical propositional calculus in many ways – Gentzen-style sequent calculus, Prawitz' Natural Deduction, or Hilbert-style axiomatization (Wansing 1996).

We will write '$\vdash_{\mathsf{PROP}} \phi$' for '$\phi$ is derivable in the proof system of PROP':

- Modens Ponens: to infer ψ from ϕ and $\phi \rightarrow \psi$
- $\vdash_{\mathsf{PROP}} \phi \rightarrow (\psi \rightarrow \phi)$
- $\vdash_{\mathsf{PROP}} (\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi))$
- $\vdash_{\mathsf{PROP}} (\neg\phi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \psi))$

Standard proofs of both *correctness* — $\vdash_{\mathsf{PROP}} \phi \Rightarrow \models_{\mathsf{PROP}} \phi$ — and *completeness* — $\models_{\mathsf{PROP}} \phi \Rightarrow \vdash_{\mathsf{PROP}} \phi$ — abound in the literature (Wansing 1996), and are not dealt with here.

4 Temporal Dynamic Logic

4.1 The Syntax of $\mathcal{L}_{\mathsf{TDL}}$

We first define the language of Temporal Dynamic Logic (TDL) $\mathcal{L}_{\mathsf{TDL}}$, starting from $\mathcal{L}_{\mathsf{PROP}}$. Let $\mathcal{A}$ be a set of atomic action symbols containing ξ, i.e. $\{\xi, \mathrm{a}_0, \ldots, \mathrm{b}_0, \ldots, \ldots\}$. We will use α, β, ... as variables over $\mathcal{A}$, and ϕ, ψ, ... as variables over $\mathcal{L}_{\mathsf{TDL}}$ now. $\mathcal{L}_{\mathsf{TDL}}$ is inductively defined as the smallest set satisfying:

[2] Since by exclusion $\bot \notin \mathcal{I}$, $\mathcal{I} \not\models_{\mathsf{PROP}} \bot$.

- $\mathcal{L}_{\mathsf{PROP}} \subset \mathcal{L}_{\mathsf{TDL}}$
- $\phi, \psi \in \mathcal{L}_{\mathsf{TDL}} \Rightarrow (\phi \mathbf{U}_s^+ \psi), (\phi \mathbf{S}_s^+ \psi), (\phi \mathbf{U}_s^+ \alpha), (\phi \mathbf{S}_s^+ \alpha), (\alpha \mathbf{U}_s^+ \phi), (\alpha \mathbf{S}_s^+ \phi), [\alpha]\phi, <\alpha>\phi \in \mathcal{L}_{\mathsf{TDL}}$

We read $\mathbf{U}_s^+$ and $\mathbf{S}_s^+$ as '(strong irreflexive) until' and '(strong irreflexive) since' respectively, and $[\alpha]$ and $<\alpha>$ as 'after α' and 'α allows'.

4.2 General Considerations

The preceding syntax definition might look surprising. Why not simply defining plain **U** and **S** modal operators instead?

The reason that this level of refinement is desirable, is that there are several reasonable options, that are not to be excluded, when defining a semantics for **U** and **S**. For example, consider the **S**-operator. Clearly, '$\phi \mathbf{S} \psi$' should express something about ϕ holding ever since ψ held. But does it, or doesn't it require that ever such a ψ did hold? When evaluating in a particular state, is the truth value of ϕ in that state to be taken into consideration, or isn't it? There are choices to acknowledge, options to provide. For that reason, a further refinement has to be made when defining the modal operators. We follow Barringer, Fisher, Gabbay, Owens & Reynolds (1996). It should be noted that for reasons of clarity some adaptations are made to notation – the goal here is clarity of understanding rather than strict adherence to notations used in the past. Basically, as introduced above, there are fundamentally two temporal operators: **S** and **U**[3]. However, they can be 'flavoured' differently, in two ways: so there are actually four incarnations of each operator to consider.

The first choice to be made when defining a semantics for the **S** and **U** operator are whether or not it pertains to the present – i.e. does it have to rain, or need it not rain, when you state "it will rain until the dancing ceases"? One option is to include the current moment – it rains now, and will keep doing so. The other option is not to include the current moment – whether it does or does not rain now is immaterial, but from now on it will definitely be raining. The option in which the present is included is *reflexive*, the one in which it isn't is *irreflexive*. The reflexive versions of the operators will be written as $\mathbf{S}^*$ and $\mathbf{U}^*$, their irreflexive counterparts as $\mathbf{S}^+$ and $\mathbf{U}^+$ respectively[4].

[3] Further on, other operators will be introduced on the syntactical level, as abbreviations.

[4] Barringer et al. (1996, pp. 20 vv.) write $\mathbf{S}^*$ and $\mathbf{U}^*$ as **S** and **U**. As a persistent reminder of the subtleties of interpretation, this notation will not be adapted.

The second choice to be made is whether or the **S** and **U** are to be interpreted as *assertive* operators, in the sense that one can wish to include the positive assertion that the dance will eventually surely be over in the proposition "it will rain until the dance is over". The *existential* or *strong* interpretation asserts that there is a reachable point in the future at which the dance will finish, while to the *universal* or *weak* interpretation it is irrelevant whether or it ever ends or not. We write the strong operators as $\mathbf{S}_s$ and $\mathbf{U}_s$, and the weak operators as $\mathbf{S}_w$ and $\mathbf{U}_w$[5].

Onwards in this text, the annotated **S** and **U** will appear, and are to be considered part of $\mathcal{L}_{\mathsf{TDL}}$. It should be noted that considerations of reflexivity play no role when evaluating **U**- and **S**-formulae that have an action component on the right side, since actions are located between, not in, states. Actions are not properties of a particular state.

4.3 Making preparations for the semantics of $\mathcal{L}_{\mathsf{TDL}}$

Having established the language, and taking all musings on required subtlety into consideration, we will now define the semantics for the expressions using $\mathbf{U}_s^+$ and $\mathbf{S}_s^+$. $\mathbf{S}_w^*$, $\mathbf{S}_s^*$, $\mathbf{S}_w^+$, $\mathbf{U}_w^*$, $\mathbf{U}_s^*$, and $\mathbf{U}_w^+$ will be defined as syntactic abbreviations. The intended meaning of $\mathbf{U}_s^+$ is that a certain proposition or action is demanded to hold (or take place) until eventually some other proposition or action holds (or takes place), and that this other proposition or action eventually holds at some point (or takes place). Similarly, $\mathbf{S}_s^+$ means that a certain proposition or action is demanded to have held (or have taken place) ever since some other proposition or action held (or took place), and that this other proposition or action did at some point hold (or take place).

A model-theoretic semantics serves the purpose or interpreting $\mathcal{L}_{\mathsf{TDL}}$-formulae.

Let $\mathcal{S}$ be an arbitary finite fixed set. We view $\mathcal{S}$ as a set of *states* (also commonly referred to as *worlds*). In each state the truth value of all propositional constants is known and given by $\mathcal{I}\colon\ \mathcal{S} \to \wp(\mathcal{P})$, a so-called *interpretation* function. As to be expected, $\bot$ is excluded from $\mathcal{I}(s)$ for all $s \in \mathcal{S}$. Furthermore, let $\mathcal{R} : \mathcal{S} \to \wp(\mathcal{S} \times \mathcal{A})$ be a relation that specifies for each state which states, if any, are (immediately) reachable through which action. For convenience, we will write $s\mathcal{R}_a t$ rather than $\langle s, \langle t, a\rangle\rangle \in \mathcal{R}$ or $\langle t, a\rangle \in \mathcal{R}(s)$, and $s_1\mathcal{R}_\alpha s_2\mathcal{S}_\beta s_3 \ldots s_{n-1}\mathcal{T}_\gamma s_n \ldots$ for $(s_1\mathcal{R}_\alpha s_2 \wedge s_2\mathcal{S}_\beta s_3 \wedge \ldots \wedge s_{n-1}\mathcal{T}_\gamma s_n \wedge \ldots)$.

[5] Note that Barringer et al. (1996, pp. 20 vv.) rename the weak incarnations of **S** and **U** to **Z** and **W**. In order not to suggest a lack of kinship, this notation will not be adapted.

4.4 Reachability and Paths

We define $\mathcal{R}^+$, the transitive closure of $\mathcal{R}$ with respect to $\mathcal{S}$, and $\mathcal{R}^*$, the reflexive-transitive closure, inductively as follows[6]:

- $\forall s,s_1 \in \mathcal{S}(\forall \alpha \in \mathcal{A}(s\mathcal{R}_\alpha s_1 \Rightarrow s\mathcal{R}^+ s_1))$
- $\forall s,s_1,s_2 \in \mathcal{S}((sR^+ s_1 \wedge s_1\mathcal{R}^+ s_2) \Rightarrow s\mathcal{R}^+ s_2)$
- $\forall s \in \mathcal{S}(s\mathcal{R}^* s)$
- $\forall s,s_1 \in \mathcal{S}(sR^+ s_1 \Rightarrow s\mathcal{R}^* s_1)$

In the same vein, we will omit details about the action component of $\mathcal{R}$ at our convenience, whenever the action component is of no relevance. When uninterested in actions, we write '$s\mathcal{R}t$', '$t \in \mathcal{R}(s)$ instead of '$s\mathcal{R}_a t$' and '$\langle t, a\rangle \in \mathcal{R}(s)$'. For practical purposes, we will use the same looseness of notation when examining $\mathcal{R}$. We will, for example, freely write '$\mathcal{R}$ is reflexive', usually meaning something along the lines of '$\forall s \in \mathcal{S}(s\mathcal{R}s)$' instead of providing the definition our framework would require, strictly spoken: '$\forall s \in \mathcal{S}(\exists \alpha \in \mathcal{A}(s\mathcal{R}_\alpha s))$'.

It should be noted that this looseness can *only* occur whenever the action component is of no interest – for example, a different concept 'α-reflexivity' could be defined as '$\mathcal{R}$ is α-reflexive iff $\forall s \in \mathcal{S}(s\mathcal{R}_\alpha s)$', in which case of course no looseness in omitting detail could be permitted.

For convenience, and to enhance the insightfullness of the following definitions, we need yet another definition – of *paths*. A path π between states s, t is a sequence of transitions – i.e. $s\mathcal{R}_{a_0} s_1 \ldots s_n \mathcal{R}_{a_n} t$.

The set of finite[7] paths Π, the **length** ($\overline{\pi}$) of paths, the **concatenation** of paths ($\cdot$), and the **first state** (μ), **last state** (ν), **first action** (f), and **last action** (l)are simultaneously defined:

- For all $s \in \mathcal{S}$, $\pi = \langle s\rangle \in \Pi$, with $\overline{\pi} = 0$, $\mu(\pi) = \nu(\pi) = s$ and $f(\pi) = l(\pi) = \xi$
- For all $s, t \in \mathcal{S}, \alpha \in \mathcal{A}$: if $s\mathcal{R}_\alpha t$, then $\pi = \langle s, \alpha, t\rangle \in \Pi$, with $\overline{\pi} = 1$, $\mu(\pi) = s$, $\nu(\pi) = t$, and $f(\pi) = l(\pi) = \alpha$

[6] Note that $\mathcal{R}^+$ and $\mathcal{R}^*$ discard all action information.

[7] Alternatively, infinite paths could be defined (as De Nicola & Vaandrager (1990) do), using the notion of *fullpath* for the semantics. This approach has not been taken for reasons of simplicity.

- If $\pi_0 = \langle s_1, \ldots, s_k, \alpha_k, t\rangle, \pi_1 = \langle t, \beta_1, u_1, \ldots, u_l\rangle \in \Pi$ then $\pi = \pi_0\pi_1 (= \langle s_1, \ldots, s_k, \alpha_k, t, \beta_1, u_1, \ldots, u_l\rangle) \in \Pi$, with $\overline{\pi} = \overline{\pi_0} + \overline{\pi_1}$, $\mu(\pi) = \mu(\pi_0)$, and $\nu(\pi) = \nu(\pi_0)$. If $f(\pi_0) \neq \xi$, then $f(\pi) = f(\pi_0)$, otherwise $f(\pi) = f(\pi_1)$; if $l(\pi_1) \neq \xi$, then $l(\pi) = l(\pi_1)$, otherwise $l(\pi) = l(\pi_0)$

If $\pi = \langle s_1, \alpha_1, s_2, \ldots, s_k, \alpha_k, s_{k+1}\rangle$, then its **tail** $t(\pi)$ is $\langle s_2, \ldots, s_k, \alpha_k, s_{k+1}\rangle$ and its **rump** $r(\pi)$ is $\langle s_1, \alpha_1, s_2, \ldots, s_k\rangle$. A few important subsets of Π are:

- $\Pi_\mu(s) \stackrel{\text{df}}{=} \{\pi \in \Pi | \overline{\pi} > 0 \wedge s = \mu(\pi)\}$
- $\Pi_\nu(s) \stackrel{\text{df}}{=} \{\pi \in \Pi | \overline{\pi} > 0 \wedge s = \nu(\pi)\}$
- $\Pi_{\mu\nu}(s,t) \stackrel{\text{df}}{=} \Pi_\mu(s) \cap \Pi_\nu(t)$

We furthermore define the following relations on paths: $\doteq_l, \doteq_r, \doteq_{lr}$ (two paths have an initial or terminal end in common, or both, respectively) and $\Subset, \Supset$ (a path is an initial resp. a terminal subpath of another path)[8]:

- $\pi_0 \doteq_l \pi_1 \stackrel{\text{df}}{=} \mu(\pi_0) = \mu(\pi_1)$
- $\pi_0 \doteq_r \pi_1 \stackrel{\text{df}}{=} \nu(\pi_0) = \nu(\pi_1)$
- $\pi_0 \doteq_{lr} \pi_1 \stackrel{\text{df}}{=} \pi_0 \doteq_l \pi_1 \wedge \pi_0 \doteq_r \pi_1$
- $\pi_0 \Subset \pi \stackrel{\text{df}}{=} \exists \pi_1 \in \Pi(\pi_0 \cdot \pi_1 = \pi)$
- $\pi_0 \Supset \pi \stackrel{\text{df}}{=} \exists \pi_1 \in \Pi(\pi_1 \cdot \pi_0 = \pi)$

We say s_i **on** π (s_i occurs on π), and α_i **on** π, iff $\pi = \langle s_0, \alpha_0, s_1, \ldots, s_n, \alpha_n, s_{n+1}\rangle$ and $0 \leq i \leq n$[9].

4.5 Core semantics of $\mathcal{L}_{\mathsf{TDL}}$

Having established the necessary notation, a model-theoretic semantics can be defined. Let *model* $\mathcal{M} = \langle \mathcal{S}, R, I\rangle$. $\mathcal{R}$ gives rise to a set Π of paths, as above; we will use $\pi, \pi_0, \ldots$ to indicate elements of Π. We now define the

[8] Observe that $\doteq_l$, $\doteq_r$, and $\doteq_{lr}$ are equivalence relations; and that $\Subset$ and $\Supset$ are partial orders on Π.

[9] It should be noted that '**on**' ignores the final state of a path. The motivation for this is pragmatic: whenever using the '**on**' in a context concerning states, the final state would have to be explicitly ignored, if it were included.

semantic notion of validity in a state s for all formulae ϕ in $\mathcal{L}_{\mathsf{TDL}}$: $\mathcal{M}, s \models \phi$. We conveniently write 'ϕ **on** π' for '$\exists s{\in}\mathcal{S}(s \,\mathbf{on}\, \pi \wedge \mathcal{M}, s \models \phi)$'.

Propositional bits are evaluated in the usual manner:

Table 1: Propositional Semantics

$\mathcal{M}, s \models P$	$\Leftrightarrow$	$P \in \mathcal{I}(s)$[10]
$\mathcal{M}, s \models \phi \rightarrow \psi$	$\Leftrightarrow$	$\mathcal{M}, s \models \phi \Rightarrow \mathcal{M}, s \models \psi$

However, all modalities express some reachibility property, hence their interpretation essentialy refers to properties of certain paths, not merely of a single state:

Table 2: Modal Semantics

$\mathcal{M}, s \models \alpha \mathbf{U}^+_s \beta$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\mu(s)(\beta \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\mu(s)(\neg(\beta \,\mathbf{on}\, \pi) \Rightarrow (l(\pi) = \alpha))$
$\mathcal{M}, s \models \alpha \mathbf{U}^+_s \phi$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\mu(s)(\phi \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\mu(s)(\neg(\phi \,\mathbf{on}\, \pi) \Rightarrow (l(\pi) = \alpha))$
$\mathcal{M}, s \models \phi \mathbf{U}^+_s \alpha$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\mu(s)(\alpha \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\mu(s)(\neg(\alpha \,\mathbf{on}\, \pi) \Rightarrow \mathcal{M}, \nu(\pi) \models \phi)$
$\mathcal{M}, s \models \phi \mathbf{U}^+_s \psi$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\mu(s)(\psi \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\mu(s)(\neg(\psi \,\mathbf{on}\, \pi) \Rightarrow \mathcal{M}, \nu(\pi) \models \phi)$
$\mathcal{M}, s \models \alpha \mathbf{S}^+_s \beta$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\nu(s)(\beta \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\nu(s)(\neg(\beta \,\mathbf{on}\, \pi) \Rightarrow (f(\pi) = \alpha))$
$\mathcal{M}, s \models \alpha \mathbf{S}^+_s \phi$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\nu(s)(\phi \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\nu(s)(\neg(\phi \,\mathbf{on}\, \pi) \Rightarrow (f(\pi) = \alpha))$
$\mathcal{M}, s \models \phi \mathbf{S}^+_s \alpha$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\nu(s)(\alpha \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\nu(s)(\neg(\alpha \,\mathbf{on}\, \pi) \Rightarrow \mathcal{M}, \mu(\pi) \models \phi)$
$\mathcal{M}, s \models \phi \mathbf{S}^+_s \psi$	$\Leftrightarrow$	$\exists \pi{\in}\Pi_\nu(s)(\psi \,\mathbf{on}\, \pi) \wedge$ $\forall \pi{\in}\Pi_\nu(s)(\neg(\psi \,\mathbf{on}\, \pi) \Rightarrow \mathcal{M}, \mu(\pi) \models \phi)$
$\mathcal{M}, s \models [\alpha]\phi$	$\Leftrightarrow$	$\forall t{\in}\mathcal{S}(s \mathcal{R}_\alpha t \Rightarrow \mathcal{M}, t \models \phi)$
$\mathcal{M}, s \models <\alpha>\phi$	$\Leftrightarrow$	$\exists t{\in}\mathcal{S}(s \mathcal{R}_\alpha t \wedge \mathcal{M}, t \models \phi)$

A little elaboration is — considering the complexity of these definitions — in order. We will consider a few examples. How to understand the definition of $\phi \mathbf{U}^+_s \alpha$? We would like to know if it holds in a certain

[10] As before, since there can be no s with $\bot \in \mathcal{I}(s)$, $\mathcal{M}, s \not\models \bot$ for all M.

state s. The first condition of $\mathcal{M}, s \vDash \phi \mathbf{U}^+ \psi$ requires there to be *a* state, reachable from s, in which ψ holds. This requirement is formulated as 'there must be a path from s to some state t on which, at some point, ψ held'. The second condition requires that on *all* paths π starting from $s\phi$ holds at least up to a state in which ψ is valid. Or, in a different formulation, ϕ will hold finally, if no ψ occurred on π.

Importantly, if ψ does not occur on π, then ψ cannot occur on *subpaths* of π either. Particularly, all *initial* subpaths do not contain ψ, yet they do start with s. Henceforth, they are required to end in a state in which ϕ holds. The important conclusion to be drawn here is that ϕ is required to hold along the entire path, even though it might look as if ϕ is only enthrusted upon its terminal state.

The definitions can also be restated 'negatively' – i.e. instead of stating the criteria for validity, a formulation that prohibits a situation that would invalidate the formula under consideration can be given. For example, de second demand for $\phi \mathbf{U}_s^+ \psi$, can be paraphrased as 'whenever a state is reached in which ϕ does not hold, a previous state must have been encountered on which ψ held'.

Using the notion of semantic validity in a state defined above, we can – as usual – also define the more general notions of *validity in a model* and *validity in a frame*, i.e. validity independent of an interpretation function. Validity in a model is defined as follows: $\mathcal{M} \vDash \phi$ iff $\mathcal{M}, s \vDash \phi$ for all $s \in \mathcal{S}$. A formula is valid in model if – given a particular interpretation function – it holds in all states.

Validity on a frame $\mathcal{F} = \langle \mathcal{S}, \mathsf{R} \rangle$ is consecutively defined in the following way: $\mathcal{F} \vDash \phi$ iff $\mathcal{M} \vDash \phi$ for all $\mathcal{I}$. In other words, a formula is valid on a frame if it holds in all states under all interpretations. We will also refer to frames as *reachability graphs*.

5 Derived Operators

5.1 Definitions

These refinements give – of course – rise to a host of slightly different derived operators; a short overview should prove useful to manage the plethora of operators introduced. First, an informal introduction to a staggering amount of various operators is given; later on, they will be (syntactically) defined in terms of $\mathbf{U}_s^+$ and $\mathbf{S}_s^+$.

Table 3: Overview of Operators

Notat.	Informal meaning	Notat.	Informal meaning
$\mathbf{U}_s^+$	Irreflexive strong **u**ntil	$\mathbf{S}_s^+$	Irreflexive strong **s**ince
$\mathbf{U}_w^+$	Irreflexive strong **u**ntil	$\mathbf{S}_w^+$	Irreflexive strong **s**ince
$\mathbf{U}_s^*$	Reflexive strong **u**ntil	$\mathbf{S}_s^*$	Reflexive strong **s**ince
$\mathbf{U}_w^*$	Reflexive weak **u**ntil	$\mathbf{S}_w^*$	Reflexive weak **s**ince
$\mathbf{F}^+$	Irreflexive sometime in the **f**uture	$\mathbf{P}^+$	Irreflexive sometime in the **p**ast
$\mathbf{F}^*$	Reflexive sometime in the **f**uture	$\mathbf{P}^*$	Reflexive sometime in the **p**ast
$\mathbf{G}^+$	Irreflexive always in the future	$\mathbf{H}^+$	Irreflexive always in the past
$\mathbf{G}^*$	Reflexive always in the future	$\mathbf{H}^*$	Reflexive always in the past
$\mathbf{N}_s$	Strong **n**ext (existential next)	$\mathbf{E}_s$	Strong previous (existential **e**arlier)
$\mathbf{N}_w$	Weak **n**ext (universal next)	$\mathbf{E}_w$	Weak previous (universal **e**arlier)

The operators just introduced can (and must) be defined in terms of the existing $\mathbf{S}_s^+$ and $\mathbf{U}_s^+$. Note that, since $\neg\alpha$ makes no sense, $\mathbf{N}_w\alpha$, $\mathbf{E}_w\alpha$, $\mathbf{G}^+\alpha$, $\mathbf{G}^*\alpha$, $\mathbf{H}^+\alpha$, and $\mathbf{H}^*\alpha$ do not occur; for the same reason, $\alpha\mathbf{U}_w^+\ldots$, $\alpha\mathbf{U}_w^*\ldots$, $\alpha\mathbf{S}_w^+\ldots$, and $\alpha\mathbf{S}_w^*\ldots$ are not defined[11].

Table 4: More Operator Definitions

$\alpha\mathbf{U}_s^*\beta$	$\stackrel{\mathsf{df}}{=} \alpha\mathbf{U}_s^+\beta$	$\alpha\mathbf{S}_s^*\beta$	$\stackrel{\mathsf{df}}{=} \alpha\mathbf{S}_s^+\beta$
$\alpha\mathbf{U}_s^*\phi$	$\stackrel{\mathsf{df}}{=} \phi \vee \alpha\mathbf{U}_s^+\phi$	$\alpha\mathbf{S}_s^*\phi$	$\stackrel{\mathsf{df}}{=} \phi \vee \alpha\mathbf{S}_s^+\phi$
$\phi\mathbf{U}_s^*\alpha$	$\stackrel{\mathsf{df}}{=} \phi \wedge \phi\mathbf{U}_s^+\alpha$	$\phi\mathbf{S}_s^*\alpha$	$\stackrel{\mathsf{df}}{=} \phi \wedge \phi\mathbf{S}_s^+\alpha$
$\phi\mathbf{U}_w^+\alpha$	$\stackrel{\mathsf{df}}{=} \mathbf{G}^+\phi \vee \phi\mathbf{U}_s^+\alpha$	$\phi\mathbf{S}_w^+\alpha$	$\stackrel{\mathsf{df}}{=} \mathbf{H}^+\phi \vee \phi\mathbf{S}_s^+\alpha$

Continued on next page

[11] However, that does not necessarily imply they cannot be given meaning to in some other way, within the very same framework – look further on in this text for a definition that is not stated as an abbreviation using $\mathbf{S}_s^+$ or $\mathbf{U}_s^+$.

(More Operator Definitions – continued)

Continued from previous page

$\phi \mathbf{U}^*_w \alpha \stackrel{df}{=} \phi \wedge \phi \mathbf{U}^+_w \alpha$	$\phi \mathbf{S}^*_w \alpha \stackrel{df}{=} \phi \wedge \phi \mathbf{S}^+_w \alpha$
$\phi \mathbf{U}^*_s \psi \stackrel{df}{=} \psi \vee (\phi \wedge \phi \mathbf{U}^+_s \psi)$	$\phi \mathbf{S}^*_s \psi \stackrel{df}{=} \psi \vee (\phi \wedge \phi \mathbf{S}^+_s \psi)$
$\phi \mathbf{U}^+_w \psi \stackrel{df}{=} \mathbf{G}^+ \phi \vee \phi \mathbf{U}^+_s \psi$	$\phi \mathbf{S}^+_w \psi \stackrel{df}{=} \mathbf{H}^+ \phi \vee \phi \mathbf{S}^+_s \psi$
$\phi \mathbf{U}^*_w \psi \stackrel{df}{=} \psi \vee (\phi \wedge \phi \mathbf{U}^+_w \psi)$	$\phi \mathbf{S}^*_w \psi \stackrel{df}{=} \psi \vee (\phi \wedge \phi \mathbf{S}^+_w \psi)$
$\mathbf{F}^+ \alpha \stackrel{df}{=} \top \mathbf{U}^+_s \alpha$	$\mathbf{P}^+ \alpha \stackrel{df}{=} \top \mathbf{S}^+_s \alpha$
$\mathbf{F}^+ \phi \stackrel{df}{=} \top \mathbf{U}^+_s \phi$	$\mathbf{P}^+ \phi \stackrel{df}{=} \top \mathbf{S}^+_s \phi$
$\mathbf{F}^* \alpha \stackrel{df}{=} \mathbf{F}^+ \alpha$	$\mathbf{P}^* \alpha \stackrel{df}{=} \mathbf{P}^+ \alpha$
$\mathbf{F}^* \phi \stackrel{df}{=} \phi \vee \mathbf{F}^+ \phi$	$\mathbf{P}^* \phi \stackrel{df}{=} \phi \vee \mathbf{P}^+ \phi$
$\mathbf{G}^+ \phi \stackrel{df}{=} \neg \mathbf{F}^+ \neg \phi$	$\mathbf{H}^+ \phi \stackrel{df}{=} \neg \mathbf{P}^+ \neg \phi$
$\mathbf{G}^* \phi \stackrel{df}{=} \phi \wedge \mathbf{G}^+ \phi$	$\mathbf{H}^* \phi \stackrel{df}{=} \phi \wedge \mathbf{H}^+ \phi$
$\mathbf{N}_s \alpha \stackrel{df}{=} \bot \mathbf{U}^+_s \alpha$	$\mathbf{E}_s \alpha \stackrel{df}{=} \bot \mathbf{S}^+_s \alpha$
$\mathbf{N}_s \phi \stackrel{df}{=} \bot \mathbf{U}^+_s \phi$	$\mathbf{E}_s \phi \stackrel{df}{=} \bot \mathbf{S}^+_s \phi$
$\mathbf{N}_w \phi \stackrel{df}{=} \neg \mathbf{N}_s \neg \phi$	$\mathbf{E}_w \phi \stackrel{df}{=} \neg \mathbf{E}_s \neg \phi$

It should be noted that these abbreviations could have been stated in a number of (equivalent) ways: for example, $\mathbf{F}^*\alpha$ could have been defined as $\top \mathbf{U}^*_s \alpha$, $\mathbf{P}^*\alpha$ as $\top \mathbf{S}^*_s \alpha$. The choice for the definition is motivated by the desire to stretch the dualities of the various operators, rather then emphasize on their 'internal encodings'.

5.2 Derived Semantics

Instead of having to take refuge to the semantics of the 'underlying' operator, one would like to have some direct grasp of what the behaviour of a particular abbreviation is. For a few typical abbreviations, a derived semantics is given.

$\phi \mathbf{U}^*_s \alpha$ First, we expand the definition fully, obtaining $\phi \wedge \phi \mathbf{U}^+_s \alpha$. Considering the semantics for $\wedge$ and $\phi \mathbf{U}^+_s \alpha$, and combining them, we can formulate the derived semantics:

> $\mathcal{M}, s \vDash \phi \mathbf{U}^*_s \alpha$ iff $\mathcal{M}, s \vDash \phi$, there is a path $\pi \in \Pi_\mu(s)$ s.t. α **on** π and at the end of all paths $\pi \in \Pi_\mu(s)$ on which α does not occur
> phi holds.

$\mathbf{F}^+\phi$. The derived semantics is the one for $\top\mathbf{U}^+_s\phi$. To reiterate, it is
ested that there both be a reachable path containing ϕ, and that
aths not containing a state where ϕ occurs end in a state in which
olds. Now, since $\top$ holds in all states, the latter condition is
ous – leaving only the following as the derived semantics:

$\mathcal{M}, s \models \mathbf{F}^+\phi$ iff there is a path $\pi \in \Pi_\mu(s)$ s.t. ϕ **on** π.

ing on the derived semantics for $\mathbf{F}^+\phi$, a semantics for $\mathbf{G}^+\phi$ can
ven. $\mathbf{G}^+\phi$ is defined as $\neg\mathbf{F}^+\neg\phi$. We reformulate directly, intro-
g both negations simultaneously:

$\mathcal{M}, s \models \mathbf{G}^+\phi$ iff there is no path $\pi \in \Pi_\mu(s)$ s.t. $\neg\phi$ **on** π.

d positively:

$\mathcal{M}, s \models \mathbf{G}^+\phi$ iff for all paths $\pi \in \Pi_\mu(s)$ we have $\mathcal{M}, \nu(\pi) \models \phi$[12].

nding the definition yields $\mathbf{G}^+\phi \vee \phi\mathbf{U}^+_s\alpha$. Combining the derived
ntics for $\mathbf{G}^+\phi$ and $\phi\mathbf{U}^+_s\alpha$ leads to:

$\mathcal{M}, s \models \phi\mathbf{U}^+_w\alpha$ iff either ϕ holds in all reachable states, or an α-transition can be reached with ϕ holding (at least) in all states up to an α-transition.

g the derived semantics for $\phi\mathbf{U}^+_w\alpha$[13], we give a semantics for $\phi \wedge \alpha$ directly:

$\mathcal{M}, s \models \phi\mathbf{U}^*_w\alpha$ iff firstly $\mathcal{M}, s \models \phi$, and secondly either ϕ holds in all reachable states, or an α-transition can be reached with ϕ holding (at least) in all states up to an α-transition.

Repeating this yields $\mathbf{U}^+_s$, we see this re-
taining α, and that $\perp$ must hold in all
ncountered. However, since $\perp$ cannot
hat every path from s *must* start with
n, can be stated as follows:

[12] Alternatively, this could be stated as "ϕ holds in all reachable states".

[13] Otherwise, a semantics for the horrendous formula $\phi \wedge (\neg(\top\mathbf{U}^+_s\neg\phi) \vee \phi\mathbf{U}^+_s\alpha)$ would have to be found.

$\mathcal{M}, s \vDash \mathbf{N}_s\alpha$ iff there is a path $\pi \in \Pi_\mu(s)$, and for all paths $\pi \in \Pi_\mu(s)$, $f(\pi) = \alpha$.

$\mathbf{N}_w\phi$ The semantics for $\mathbf{N}_w\phi$ are defined strictly analogous to those for $\mathbf{N}_s\alpha$. Adapting them, and introducing both negations, we get:

$\mathcal{M}, s \vDash \mathbf{N}_w\phi$ iff for all paths $\pi \in \Pi_\mu(s)$ $\mathcal{M}, \mu(r(\pi)) \vDash \phi$.

Hopefully, these few examples suffice to provide an idea about the semantics of the other cases[14]. They are derived very similarly (or even analogously); for that reason no exhaustive listing is provided.

5.3 More Semantics

Interestingly, some operators that could not be defined syntactically before due to the absence of a semantics for action negation actually *can* be defined. A few examples (only the cases for $\alpha\mathbf{U}_w^{(+,*)}(\beta, \phi)$, $\mathbf{G}^{(+,*)}\alpha$ and $\mathbf{N}_w\alpha$ are shown[15]) serve to clarify.

$\mathbf{G}^+\alpha$ The semantics for $\mathbf{G}^+\alpha$ should express – analogously to the semantics for $\mathbf{G}^+\phi$ – something like "α is the only reachable transition":

$\mathcal{M}, s \vDash \mathbf{G}^+\alpha$ iff for all paths $\pi \in \Pi_\mu(s)$ we have $l(\pi) = \alpha$[16].

$\mathbf{G}^*\alpha$ Because whether or not the current state is taken into account is of no relevance for actions, since actions are inbetween states, the semantics for $\mathbf{G}^*\alpha$ coincides with the semantics for $\mathbf{G}^+\alpha$.

$\alpha\mathbf{U}_w^+\beta$ Needed to express this one is being able to express $\mathbf{G}^+\alpha$. That one has been defined. Using that, we get:

$\mathcal{M}, s \vDash \alpha\mathbf{U}_w^+\beta$ iff $\mathcal{M}, s \vDash \alpha\mathbf{U}_s^+\beta$ or for all paths $\pi \in \Pi_\mu(s)$ we have $l(\pi) = \alpha$.

$\alpha\mathbf{U}_w^*\beta$ Coincides with $\alpha\mathbf{U}_w^+\beta$.

$\alpha\mathbf{U}_w^+\phi$ Needed to express this one is being able to express $\mathbf{G}^+\alpha$. By now we can, so we get:

[14] Trying out a few might prove insightful for the reader, and will definitely clarify the dualities and dependencies.

[15] The definitions for $\alpha\mathbf{S}_w^{(+,*)}(\beta, \phi)$, $\mathbf{H}^{(+,*)}\alpha$ and $\mathbf{E}_w\alpha$ are, as in the cases already shown, straightforward duals and should not be too hard to imagine.

[16] Once again, the definition ranges implicitly over all subpaths.

$\mathcal{M}, s \models \alpha \mathbf{U}_w^+ \beta$ iff either for all paths $\pi \in \Pi_\mu(s)$ we have $l(\pi) = \alpha$, or $\mathcal{M}, s \models \alpha \mathbf{U}_s^+ \phi$.

$\alpha \mathbf{U}_w^* \phi$ Unsurprisingly, this case coincides with $\alpha \mathbf{U}_w^+ \phi$.

$\mathbf{N}_w \alpha$ Adapting the definition for $\mathbf{N}_w \phi$ to handle the difference between actions (occurring between states) and propositions (evaluated in states):

$\mathcal{M}, s \models \mathbf{N}_w \alpha$ iff for all paths $\pi \in \Pi_\mu(s)$ we have $f(\pi) = \alpha$.

6 Validities

Up to this point, all operators have been defined on a very wide range of models – graphs, without any constraints. They need not be connected; there can be arbitrary circularity, and so on. Yet, when adapting a more realistic viewpoint as to the kind of models constructed, it can be reasonable to choose a more restricted class of models. Naturally, 'weaker' models may behave differently: other formulas may hold. This is – or can be – of interest, because there are restrictions with interesting structures. For example, a subclass of the full graph framework is the class of 'trees', i.e. directed graphs that are not circular, and in which every node has at most one 'incoming' edge.

PROPOSITION 1 (*Strong implies weak)* $(\alpha \mathbf{U}_s^+ \beta \Rightarrow \alpha \mathbf{U}_w^+ \beta)$, $(\alpha \mathbf{U}_s^+ \phi \Rightarrow \alpha \mathbf{U}_w^+ \phi)$, $(\phi \mathbf{U}_s^+ \alpha \Rightarrow \phi \mathbf{U}_w^+ \alpha)$, $(\phi \mathbf{U}_s^+ \psi \Rightarrow \phi \mathbf{U}_w^+ \psi)$, $(\alpha \mathbf{U}_s^* \beta \Rightarrow \alpha \mathbf{U}_w^* \beta)$, $(\alpha \mathbf{U}_s^* \phi \Rightarrow \alpha \mathbf{U}_w^* \phi)$, $(\phi \mathbf{U}_s^* \alpha \Rightarrow \phi \mathbf{U}_w^* \alpha)$, $(\phi \mathbf{U}_s^* \psi \Rightarrow \phi \mathbf{U}_w^* \psi)$, $(\alpha \mathbf{S}_s^+ \beta \Rightarrow \alpha \mathbf{S}_w^+ \beta)$, $(\alpha \mathbf{S}_s^+ \phi \Rightarrow \alpha \mathbf{S}_w^+ \phi)$, $(\phi \mathbf{S}_s^+ \alpha \Rightarrow \phi \mathbf{S}_w^+ \alpha)$, $(\phi \mathbf{S}_s^+ \psi \Rightarrow \phi \mathbf{S}_w^+ \psi)$, $(\alpha \mathbf{S}_s^* \beta \Rightarrow \alpha \mathbf{S}_w^* \beta)$, $(\alpha \mathbf{S}_s^* \phi \Rightarrow \alpha \mathbf{S}_w^* \phi)$, $(\phi \mathbf{S}_s^* \alpha \Rightarrow \phi \mathbf{S}_w^* \alpha)$, $(\phi \mathbf{S}_s^* \psi \Rightarrow \phi \mathbf{S}_w^* \psi)$, $(\mathbf{N}_s \alpha \Rightarrow \mathbf{N}_w \alpha)$, $(\mathbf{N}_s \phi \Rightarrow \mathbf{N}_w \phi)$, $(\mathbf{E}_s \alpha \Rightarrow \mathbf{E}_w \alpha)$. $(\mathbf{E}_s \phi \Rightarrow \mathbf{E}_w \phi)$.
Proof Immediate (propositional, $\vee$-introduction). ■

PROPOSITION 2 $\mathbf{N}_w(\phi \rightarrow \psi) \Rightarrow (\mathbf{N}_w \phi \rightarrow \mathbf{N}_w \psi)$
Proof Take an arbitrary state s. Assume $s \models \mathbf{N}_w(\phi \rightarrow \psi)$. Furthermore, assume $s \models \mathbf{N}_w \phi$. To prove: $s \models \mathbf{N}_w \psi$. If there is no state reachable from s, then trivially $s \models \mathbf{N}_w \psi$. Otherwise, take an arbitrary state t that is directly reachable (i.e. in a single $\mathcal{R}_\alpha$-step). By the semantics of $\mathbf{N}_w$, $t \models (\phi \rightarrow \psi)$, and also $t \models \phi$. Now by the semantics of $\rightarrow$, $t \models \psi$, and since this holds for all states t reachable from s in a single step, $s \models \mathbf{N}_w \psi$. ■

PROPOSITION 3 $[\alpha]\phi \Rightarrow (\mathbf{N}_s \alpha \rightarrow \mathbf{N}_s \phi)$

Proof Take an arbitrary state $\mathfrak{s}$, suppose $\mathfrak{s} \vDash [\alpha]\phi$ and $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\alpha$. We now have to prove that $\mathbf{N}_\mathfrak{s}\phi$. By the semantics of $\mathbf{N}_\mathfrak{s}$ we know there is a state (say, $\mathfrak{t}$) directly reached by an α-step; applying the semantics of $[\alpha]$ we infer that in all states $\mathfrak{u}$ (of which $\mathfrak{t}$ is one) s.t. $\mathfrak{s}\mathcal{R}_\alpha\mathfrak{u}$ we have $\mathfrak{u} \vDash \mathbf{N}_\mathfrak{s}\phi$. Particularly, $\mathfrak{t}$ (sure to exist) must have $\mathfrak{t} \vDash \mathbf{N}_\mathfrak{s}\phi$. Thusly, the conditions for $\mathbf{N}_\mathfrak{s}\phi$ are satisfied. ■

Clairvoyant structures are labeled graphs in which all outgoing edges of each node are labeled identically[17].
Action-recall structures are labeled graphs in which all incoming edges of each node are labeled identically[18].
Eert structures (or *eerts*) are clairvoyant structures in which all nodes have at most one outgoing edge.
Tree structures (or *trees*) are action-recall structures in which all nodes have at most one incoming edge.
Trace structures (or *traces*) have eert as well as tree proporties[19].

PROPOSITION 4 $\mathbf{N}_\mathfrak{s}\mathbf{E}_\mathfrak{s}\alpha \Rightarrow \mathbf{N}_\mathfrak{s}\alpha$.
Proof Suppose in an arbitrary $\mathfrak{s}$ we have $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\mathbf{E}_\mathfrak{s}\alpha$. To prove: $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\alpha$. By the semantics of $\mathbf{N}_\mathfrak{s}$ we know that for all $\mathfrak{t}$ s.t. $\mathfrak{s}\mathcal{R}_\alpha\mathfrak{t}$ we have $\mathfrak{t} \vDash \mathbf{E}_\mathfrak{s}\alpha$, and that there exists such a $\mathfrak{t}$. ■

PROPOSITION 5 (*Actions are intermediary)* On action-recall structures (hence also on trees and traces), we have $\mathbf{N}_\mathfrak{s}\mathbf{E}_\mathfrak{s}\alpha \Leftrightarrow \mathbf{N}_\mathfrak{s}\alpha$[20]
Proof
'$\Rightarrow$': Proposition 4.
'$\Leftarrow$': Suppose in an arbitrary $\mathfrak{s}$ we have $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\alpha$. To prove: $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\mathbf{E}_\mathfrak{s}\alpha$. By the semantics of $\mathbf{N}_\mathfrak{s}$ we know that $\mathsf{f}(\pi) = \alpha$ for all $\pi \in \Pi_\mu(\mathfrak{s})$, and that $\Pi_\mu(\mathfrak{s}) \neq \emptyset$. Therefore, all states $\mathfrak{t}$ immediately reachable from $\mathfrak{s}$ are reached by α. Now by action-recall, α is the *only* action leading to all these $\mathfrak{t}$. We already knew that there exists such a reachable $\mathfrak{t}$; so we can safely conclude that for all these $\mathfrak{t}$ $\mathfrak{t} \vDash \mathbf{E}_\mathfrak{s}\alpha$. But since this held for all immediately (from $\mathfrak{s}$) reachable $\mathfrak{t}$, and such a $\mathfrak{t}$ exists, we can now also conclude that $\mathfrak{s} \vDash \mathbf{N}_\mathfrak{s}\mathbf{E}_\mathfrak{s}\alpha$. ■

[17] Characteristic for clairvoyant structures is that in a state with an outgoing α, we'll always find $\mathbf{N}_\mathfrak{s}\alpha$.
[18] Characteristic for action-recall structures is that in a state with an incoming α, we'll always find $\mathbf{E}_\mathfrak{s}\alpha$.
[19] Hence, traces $\subset$ eerts $\subset$ clairvoyant-structures, and traces $\subset$ trees $\subset$ action-recall-structures.
[20] The reader should note that its obvious similar-looking counterpart – proposition 6 – does not hold on these structures.

PROPOSITION 6 (*Actions shmactions)* $\mathbf{E}_s\alpha \Leftrightarrow \mathbf{E}_s\mathbf{N}_s\alpha$ on clairvoyant structures
Proof
'$\Rightarrow$': Consider an arbitrary s s.t. $s \vDash \mathbf{E}_s\alpha$. To prove: $s \vDash \mathbf{E}_s\mathbf{N}_s\alpha$. By the semantics of $\mathbf{E}_s$, we know that $l(\pi) = \alpha$ for all $\pi \in \Pi_\nu(s)$, and that $\Pi_\nu(s) \neq \emptyset$. Take an arbitrary t that connects to s through α. By clairvoyance, then *all* labels from t must be α-labels: $t \vDash \mathbf{N}_s\alpha$. Therefore $s \vDash \mathbf{E}_s\mathbf{N}_s\alpha$
'$\Leftarrow$': Take an arbitrary s with $s \vDash \mathbf{E}_s\mathbf{N}_s\alpha$. Observe that, as in the converse case, a single state t that connects to s must exist with $t \vDash \mathbf{N}_s\alpha$, hence $s \vDash \mathbf{E}_s\alpha$. ■

PROPOSITION 7 $[\alpha]\phi \Rightarrow \mathbf{N}_w(\mathbf{E}_w\alpha \rightarrow \phi)$.
Proof Suppose $s \vDash [\alpha]\phi$. If there are no transitions from s, trivially $s \vDash \mathbf{N}_w(\mathbf{E}_w\alpha \rightarrow \phi)$; otherwise, there is a t reachable from s either by α, or by some other action. In the former case, $t \vDash \phi$; in the latter case, $t \nvDash \mathbf{E}_w\alpha$, hence in both cases: $t \vDash \mathbf{E}_w\alpha \rightarrow \phi$. Therefore, $s \vDash \mathbf{N}_w(\mathbf{E}_w\alpha \rightarrow \phi)$. ■

PROPOSITION 8 On action-recall structures, $[\alpha]\phi \Leftrightarrow \mathbf{N}_w(\mathbf{E}_w\alpha \rightarrow \phi)$.
Proof
'$\Rightarrow$': See Proposition 7.
'$\Leftarrow$': Suppose $s \vDash \mathbf{N}_w(\mathbf{E}_w\alpha \rightarrow \phi)$. Then for all t reached directly from s, we have $t \vDash \mathbf{E}_w\alpha \rightarrow \phi$. If there is no such t, then $s \vDash [\alpha]\phi$ is immediate. Otherwise, in both cases $t \vDash \mathbf{E}_w\alpha$, (where $t \vDash \phi$) and $t \nvDash \mathbf{E}_w\alpha$ (where no α can occur, by action-recall), we find $s \vDash [\alpha]\phi$. ■

PROPOSITION 9 On action-recall structures, $\phi\mathbf{U}_s^+\alpha \Leftrightarrow \phi\mathbf{U}_s^+\mathbf{E}_s\alpha$
Proof
'$\Rightarrow$': Take an s s.t. $s \vDash \phi\mathbf{U}_s^+\alpha$. There exists a path π from s s.t. $\alpha\,\mathbf{on}\,\pi$. Furthermore, observe that in any state t reached by α, it is reached by α *alone* (action-recall); therefore, in any state reached by α, $\mathbf{E}_s\alpha$ holds. This holds particularly for path π, that is, for the first state reached by an α-step on it: so there exists a state, reachable from s, s.t. $\mathbf{E}_s\alpha\,\mathbf{on}\,\pi$, and on all paths from $s\phi$ holds at least up to a state in which $\mathbf{E}_s\alpha$.
'$\Leftarrow$': Consider an s s.t. $s \vDash \phi\mathbf{U}_s^+\mathbf{E}_s\alpha$. Analogous to the converse case, now observating that whenever $\mathbf{E}_s\alpha$ holds, surely all incoming edges are labeled α suffices to show that $s \vDash \phi\mathbf{U}_s^+\alpha$. ■

PROPOSITION 10 On action-recall structures, $\phi\mathbf{U}_w^+\alpha \Leftrightarrow \phi\mathbf{U}_w^+\mathbf{E}_s\alpha$
Proof Trite, using Propositions 1 and 9; note Propositions 9 and 10 differ only in that for Proposition 9 there is the additional burden of proof regarding the existence of a path on which α (resp. $\mathbf{E}_s\alpha$) occurs. ■

PROPOSITION 11 On action-recall structures, $\phi \mathbf{U}^*_w \alpha \Leftrightarrow \phi \mathbf{U}^*_w \mathbf{E}_s \alpha$, $\phi \mathbf{U}^*_s \alpha \Leftrightarrow \phi \mathbf{U}^*_s \mathbf{E}_s \alpha$
Proof Analogous to Proposition 10; omitted. ■

PROPOSITION 12 On action-recall structures, $\alpha \mathbf{U}^+_s \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_s \phi$, $\alpha \mathbf{U}^+_w \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_w \phi$, $\alpha \mathbf{U}^*_s \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^*_s \phi$, $\alpha \mathbf{U}^*_w \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^*_w \phi$
Proof Similar to Propositions 9, 10, and 11, observing the action-recall property of α and $\mathbf{E}_s \alpha$ coinciding. ■

PROPOSITION 13 On action-recall structures, $\mathbf{F}^+ \alpha \Leftrightarrow \mathbf{F}^+ \mathbf{E}_s \alpha$
Proof Boring[21], hence omitted. ■

PROPOSITION 14 $\mathbf{F}^* \mathbf{N}_s \alpha \Rightarrow \mathbf{F}^+ \alpha$
Proof Suppose $s \vDash \mathbf{F}^* \mathbf{N}_s \alpha$. Then we must either have $s \vDash \mathbf{N}_s \alpha$, or there is some $\pi \in \Pi_\mu(s)$ with $\mathbf{N}_s \alpha \, \mathbf{on} \, \pi$. In the former as well as the latter case, we bumped into a path from s containing an α, therefore $s \vDash \mathbf{F}^+ \alpha$. ■

PROPOSITION 15 On clairvoyant structures, $\mathbf{F}^* \mathbf{N}_s \alpha \Leftrightarrow \mathbf{F}^+ \alpha$
Proof
'$\Rightarrow$': Proposition 14.
'$\Leftarrow$': Suppose $s \vDash \mathbf{F}^+ \alpha$. Then there is a $\pi \in \Pi_\mu(s)$ s.t. $\alpha \, \mathbf{on} \, \pi$. But then (by clairvoyance) either there is a state $\mathbf{t} \, \mathbf{on} \, \pi$ s.t. $\mathbf{t} \vDash \mathbf{N}_s \alpha$, or $\mathbf{t} \vDash \mathbf{N}_s \alpha$ (if $\mathsf{f}(\pi) = \alpha$), hence $s \vDash \mathbf{F}^* \mathbf{N}_s \alpha$. ■

PROPOSITION 16 $\mathbf{G}^* \mathbf{N}_s \alpha \Leftrightarrow \mathbf{G}^* \alpha$
Proof Omitted. ■

Many of the above proofs are in fact variations on the same theme, and depend on a single essential observation concerning the fact that on certain structures action properties and (propositional) expressions about actions coincide. To be more precise: on action-recall structures, $\mathbf{E}_s \alpha$ will hold in *exactly* all states reachable by an α-action. Similarly, in clairvoyant structures being reached through α ensures that in all predecessor states $\mathbf{N}_s \alpha$. Amongst others, propositions 9, 10, 11, and 12 all owe to this property. The following propositions (17 and 18) summarize it.

PROPOSITION 17 On action-recall structures, $\alpha \mathbf{U}^+_s \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_s \phi$, $\alpha \mathbf{U}^+_w \phi \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_w \phi$, ..., $\phi \mathbf{U}^+_s \alpha \Leftrightarrow \phi \mathbf{U}^+_s (\mathbf{E}_s \alpha)$, ..., $\alpha \mathbf{U}^+_s \beta \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_s \beta \Leftrightarrow \alpha \mathbf{U}^+_s (\mathbf{E}_s \beta) \Leftrightarrow (\mathbf{E}_s \alpha)\mathbf{U}^+_s (\mathbf{E}_s \beta)$, ...
Proof Omitted; Proposition 12 reveals the general picture. ■

[21] Well, it *should* be by now.

PROPOSITION 18 On clairvoyant structures, $\alpha \mathbf{S}_s^+ \Leftrightarrow (\mathbf{N}_s\alpha)\mathbf{S}_s^+\phi$, $\alpha \mathbf{S}_w^+\phi \Leftrightarrow (\mathbf{N}_s\alpha)\mathbf{S}_w^+\phi, \ldots, \phi\mathbf{S}_s^+\alpha \Leftrightarrow \phi\mathbf{S}_s^+(\mathbf{N}_s\alpha), \ldots, \alpha\mathbf{S}_s^+\beta \Leftrightarrow (\mathbf{N}_s\alpha)\mathbf{S}_s^+\beta \Leftrightarrow \alpha\mathbf{S}_s^+(\mathbf{N}_s\beta) \Leftrightarrow (\mathbf{N}_s\alpha)\mathbf{S}_s^+(\mathbf{N}_s\beta), \ldots$

Proof Omitted. ■

7 Predicate Logic

Extending PROP, we define the language and semantics for first order predicate logic (PRED). We start by defining the language of predicate logic $\mathcal{L}_{\mathsf{PRED}}$.

Let $\mathcal{P}$[22] be a set of predicate letters, each of wich has some (fixed) arity (written as $\mathrm{Ar}(\mathsf{P})$)[23]. Let $\mathcal{F}$[24] be a set of function symbols (again, each of arity Ar), Let $\mathcal{V}$ be a set of variables[25].

The set of *terms* (TERM) is defined as follows:

- $\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n) \in \mathsf{TERM}$ iff $\mathsf{f} \in \mathcal{F}$, $\mathrm{Ar}(\mathsf{f}) = n$, $\{\mathsf{t}_1, \ldots, \mathsf{t}_n\} \subset \mathsf{TERM}$

The set of *basic predicates* (BASIC) is defined as follows:

- $\mathsf{P}(\mathsf{t}_1, \ldots, \mathsf{t}_n) \in \mathsf{BASIC}$ iff $\mathsf{P} \in \mathcal{P}$, $\mathrm{Ar}(\mathsf{P}) = n$ and $\mathsf{t}_1, \ldots, \mathsf{t}_n \in \mathsf{TERM}$

$\mathcal{L}_{\mathsf{PRED}}$ is now inductively defined (to avoid unnecessary complication, we define only *closed formulae*) as follows (where '$\phi[\mathsf{c}/\mathsf{x}]$' means '$\phi$, with variable x uniformly substituted for some constant c possibly occurring in ϕ'):

- $\mathsf{BASIC} \subset \mathcal{L}_{\mathsf{PRED}}$
- $(\phi \rightarrow \psi) \in \mathcal{L}_{\mathsf{PRED}}$ iff $\phi, \psi \in \mathcal{L}_{\mathsf{PRED}}$
- $\forall \mathsf{x}(\phi[\mathsf{c}/\mathsf{x}]) \in \mathcal{L}_{\mathsf{PRED}}$ iff $\phi \in \mathcal{L}_{\mathsf{PRED}}$ and $\mathsf{x} \in \mathcal{V}$

The 'missing' boolean operators can be added in the usual manner, as can (unsurprisingly) the existential quantor — $\exists \mathsf{x}\phi \stackrel{\mathsf{df}}{=} \neg\forall \mathsf{x}\neg\phi$.

[22] Containing the nullary predicate '$\bot$'.

[23] Propositions are predicates of arity 0.

[24] Constants are functions of arity 0; for convenience, we use Con as shorthand notation for $\{\mathsf{f} \in \mathcal{F} | \mathrm{Ar}(\mathsf{f}) = 0\}$, and Fun for $\{\mathsf{f} \in \mathcal{F} | \mathrm{Ar}(\mathsf{f}) > 0\}$.

[25] For simplicity's sake, the complexity of introducing *sorts*, with typed functions operating on typed arguments, and a distinct set of constants for each sort, is avoided here.

A model-theoretic semantics of $\mathcal{L}_{\mathsf{PRED}}$ is defined as follows. Formulae of $\mathcal{L}_{\mathsf{PRED}}$ are interpreted using a *valuation* function V, given a basic *interpretation* function I for predicates ($\mathrm{I}(\mathsf{P}) : \mathsf{Con}^{\mathrm{Ar}(\mathsf{P})} \to \{0,1\}$; $\mathrm{I}(\bot) = 0$) and functions ($\mathrm{I}(\mathsf{f}) : \mathsf{Con}^{\mathrm{Ar}(\mathsf{f})} \to \mathsf{Con}$)[26].

- if $\mathsf{t} = \mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ then $\mathsf{V}(\mathsf{t}) = \mathrm{I}(\mathsf{f})(\mathsf{V}(\mathsf{t}_1), \ldots, \mathsf{V}(\mathsf{t}_n))$
- $\mathsf{V}(\phi \to \psi) = 0$ iff $\mathsf{V}(\phi) = 1$ and $\mathsf{V}(\psi) = 0$; $\mathsf{V}(\phi \to \psi) = 1$ otherwise.
- if $\phi = \mathsf{P}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$, then $\mathsf{V}(\phi) = \mathrm{I}(\mathsf{P})(\mathsf{V}(\mathsf{t}_1), \ldots, \mathsf{V}(\mathsf{t}_n))$
- if $\phi = \forall \mathsf{x}\phi$, then $\mathsf{V}(\phi) = 1$ iff for all $\mathsf{c} \in \mathsf{Con} : \mathsf{V}(\phi[\mathsf{x}/\mathsf{c}]) = 1$; Otherwise, $\mathsf{V}(\phi) = 0$

The reader should take notice that $\mathcal{L}_{\mathsf{PROP}} \subset \mathcal{L}_{\mathsf{PRED}}$: propositional logic can simply be viewed as a predicate logic in which all predicates are of arity 0. Or, from a different viewpoint, propositional logic is an instance of predicate logic where terms are rendered insignificant – i.e. predicate logic collapses into propositional calculus by adding the axiom '$\forall \mathsf{x} \forall \mathsf{y}(\mathsf{x} = \mathsf{y})$'.

8 Combining Temporal Dynamic Logic and Predicate Logic

TDL and PRED are orthogonal, in the sense that their respective semantics are free from interference. For each state of a TDL-model everything needed to interpret predicates (and terms) can be specified, without causing damage to the temporal dynamic structure by doing so. Consider a single state TDL-model: this is in fact also a PRED-model. This particular predicate logic isn't very interesting – it only has nullary predicates. Now consider a more realistic predicate logic, that has 'real' predicates, terms, ... What does TDL need to deal with it?

The answer is simple: TDL formulae are built from PROP-formulae by adding temporal connectives, with a two-layered semantics: for each state, a full PROP-semantics – i.e. a valuation of all atomic propositions – is given, and the relations between states give meaning to the temporal connectives.

The very same approach can be used, if one wants to construct a temporal dynamic logic TDPL, based on PRED: for each state, provide the semantics for PRED-formulae, and use the state relations to cater for the temporal

[26] Note that an algebra can arbitrarily be given to interpret (define I on) terms in TERM, using Con as the carrier set, Fun as the operator set, by writing down equations as usual; however, this is only one of the options for defining I.

operators, exactly as in TDL. In this way, a temporal dynamic predicate logic TDL can be defined.

More formally: Let the alphabet of predicates and actions be $\mathcal{P}$ and $\mathcal{A}$, respectively, with appropriate arity functions. A TDPL- model is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathsf{R}, \mathrm{I} \rangle$, with S and R as in TDL; the interpretation function I specifies a per-state interpretation of predicates and functions — in state s, $\mathrm{I}(\mathsf{s}, \bot) = \mathsf{0}$, if $\mathsf{P} \in \mathcal{P}$ then $\mathrm{I}(\mathsf{s}, \mathsf{P}) : \mathsf{Con}^{\mathrm{Ar}(\mathsf{P})} \rightarrow \{\mathsf{0}, \mathsf{1}\}$, and if $\mathsf{f} \in \mathcal{F}$ then $\mathrm{I}(\mathsf{s}, \mathsf{f}) : \mathsf{Con}^{\mathrm{Ar}(\mathsf{f})} \rightarrow \mathsf{Con}$.

9 Example

As a first example, we consider the specifications of *features* of a telecommunication system, as described in Middelburg (1994), in which an extension of the logic ACTL is used to formulate desired properties.

First, we consider the so-called *Original Call Screening* (OCS) feature, which informally is the option to reject calls from specific sources automatically. In Middelburg (1994):

1. $\boldsymbol{AG}(\mathrm{A} \neq \mathrm{B} \wedge \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \Rightarrow \neg \mathrm{calling}(\mathrm{A}, \mathrm{B}))$
2. $\boldsymbol{AG}(\mathrm{A} \neq \mathrm{B} \wedge \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \wedge \mathrm{ready}(\mathrm{A}) \wedge \mathrm{idle}(\mathrm{B}) \Rightarrow [\mathrm{dial}(\mathrm{A}, \mathrm{B})]\mathrm{rejecting}(\mathrm{A}))$
3. $\boldsymbol{AG}(\mathrm{A} \neq \mathrm{B} \wedge \neg \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \wedge \mathrm{ready}(\mathrm{A}) \wedge \mathrm{idle}(\mathrm{B}) \Rightarrow [\mathrm{dial}(\mathrm{A}, \mathrm{B})]\mathrm{calling}(\mathrm{A}))$

These properties can easily be expressed in TDL, with predicates added:

1. $\mathbf{F}^*(\mathrm{A} \neq \mathrm{B} \wedge \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \rightarrow \neg \mathrm{calling}(\mathrm{A}, \mathrm{B}))$
2. $\mathbf{F}^*(\mathrm{A} \neq \mathrm{B} \wedge \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \wedge \mathrm{ready}(\mathrm{A}) \wedge \mathrm{idle}(\mathrm{B}) \rightarrow [\mathrm{dial}(\mathrm{A}, \mathrm{B})]\mathrm{rejecting}(\mathrm{A}))$
3. $\mathbf{F}^*(\mathrm{A} \neq \mathrm{B} \wedge \neg \mathrm{OCS}(\mathrm{A}, \mathrm{B}) \wedge \mathrm{ready}(\mathrm{A}) \wedge \mathrm{idle}(\mathrm{B}) \rightarrow [\mathrm{dial}(\mathrm{A}, \mathrm{B})]\mathrm{calling}(\mathrm{A}))$

In this case, only future behaviour is specified. Since in TDL, contrary to ACTL, which models branching time only, with no past-time operators – a duality exists between past and future formulas, the demands can even easily be strengthened to express that the desired property should *always* hold, instead of in all futures only. This can done by trivially replacing $\mathbf{F}^*$ by $\mathbf{A}$, where $\mathbf{A}\phi$ is shorthand for $\mathbf{P}^+\phi \wedge \phi \wedge \mathbf{F}^+\phi$.

The other examples are similar – mapping the ACTL-operator combination $\boldsymbol{AG}$ onto the TDL-operator $\mathbf{F}^*$, after first 'expanding' the $[\mathsf{a}_0, \ldots, \mathsf{a}_n]\phi$ ACTL-shorthand for action sequences to $[\mathsf{a}_0]\phi \wedge \ldots \wedge [\mathsf{a}_n]\phi$.

10 Frame Axioms

A commonly experience problem in writing formal specifications is that they, in some sense, tend to be 'incomplete': usually, a specification consists of 'situations' in which a particular action is enabled (or even: performed), and the results of these particular actions are specified. Suppose some formula of the typical form $\phi \rightarrow [\alpha]\psi$ occurs in a specification. This states that whenever ϕ holds and action α is performed, ψ must hold. How is this incomplete? The answer is: only the effects on the validity of ψ are specified. An implementation can satisfy this constraint – guaranteeing ψ in all possible circumstances in which α took place when ϕ held – while also having 'undesired', that is, unspecified, other effects. Yes, ψ must hold, but the specification did not rule out that α also inadvertently asserts some χ. Surely, after inserting a record into a databases' table the field must be present – but the behaviour in which a new record is inserted by overwriting an existing one will probably not be considered to be correct. This problem is known as the *frame problem*: a specification that describes what's needed, allows (pathological) implementations, that contain undesired, unprescribed, behaviour.

Using TDL-formulae, there is a convenient way of adding conditions to a specifications to prevent at least some of the undesired, yet allowed by the specification, behaviour to occur. This is done by restricting possible change. In the previous example, if α is the *only* action with ψ as a result, it can simply be written down that whenever untrue, ψ should remain so unless possibly after performing α: $\neg\psi \rightarrow (\neg\psi)\mathbf{U}^*_w\alpha$.

A small specification of a so-called *flip-flop*, or *toggle* – your average domestic lighting, for example – could consist of:

$$on \rightarrow [switchoff]off$$

$$off \rightarrow [switchon]on$$

To this specification – which suffers from frame problems – we could now easily add the following constraints.

$$\neg off \rightarrow (\neg off)\mathbf{U}^*_w switchoff$$

$$\neg on \rightarrow (\neg on)\mathbf{U}^*_w switchon$$

Doing so, the frame problem is solved for this particular specification.

In this context, a few things should be noted. First, 'desired behaviour' is a specification choice: in general it is not the case that blindly adding such

formulae to a specification does justice to the intention of the specification. Sometimes it really may not matter what other consequences a particular action has, as long as it satisfies some elementary constraint.

Second, it is not altogether trivial to derive such frame formulae. In particular, whenever two separate actions have a particular result – let's say, for example, that $\top \rightarrow [\mathtt{a}]\mathtt{p}$ and $\top \rightarrow [\mathtt{b}]\mathtt{p}$ are part of a specification. After both $\mathtt{a}$ or $\mathtt{b}$, $\mathtt{p}$ will hold. Now if were to add $\neg\mathtt{p} \rightarrow (\neg\mathtt{p})\mathbf{U}^*_w\mathtt{a}$ and $\neg\mathtt{p} \rightarrow (\neg\mathtt{p})\mathbf{U}^*_w\mathtt{b}$, the specification would rule out $\mathtt{p}$'s ever being valid – clearly this will not be what was intended by the initial specification. For this reason (amongst others), it would be useful to have additional expressive power over actions – i.e. having meaningful notions of action conjunction, disjunction, negation, implication would allow very useful requirements. In the current example, as an adstruction, it might be desired to write down $\neg\mathtt{p} \rightarrow (\neg\mathtt{p})\mathbf{U}^*_w(\mathtt{a} \vee \mathtt{b})$; but while there is a clear, intuitive, meaning to this, there is no place for a semantics for it within the scope of this paper[27].

11 Next of kin

This section contains a terse and informal discussion on some other logics that are designed to deal with aspects of time in formal specification.

11.1 Linear Temporal Logic

Linear Temporal Logic (Barringer et al. 1996) is a modal logic in which modal operators – not unlike ours – express past and future time properties. Classic temporal logic contains unary existential and universal past and/or future time operators, and is interpreted within the framework of Kripke-structures. No action information is present, nor can it be represented ('encoded') in a fully equivalent way. Furthermore, Linear Temporal Logic formulae are defined on *traces* only: more complex time structures do not occur.

There are straightforward translations, however, from LTL-formulae to TDL-formulae, and it is relatively easy to show that when restricting the class of valid TDL-models to the class of traces, the semantics for the translated formulae coincide, i.e. the translation is truth-preserving. Hence, LTL can be viewed as a simplification of TDL.

[27] The topic of 'action structuring' will surely be focused on in future work, however.

11.2 Variations on CTL

In De Nicola & Vaandrager (1990), an action-based logic is proposed, inspired by the logic CTL (Emerson & Halpern 1986). This logic, called ACTL, consists of formulae constructed using actions using boolean-style operators as well as 'path' operators. ACTL-formulae are modeled by Labelled Transition Systems.

In De Nicola & Vaandrager (1990) it is shown that there are a truth-preserving mappings between CTL and ACTL; using that fact, in De Nicola, Fantechi, Gnesi & Ristori (1993) it is shown how (existing) tools can be used to deal with ACTL-specifications.

Compared to our framework, it is noteworthy that ACTL does not contain propositions, while CTL does not contain any actions. Furthermore, the intended models for ACTL are *trees* rather than arbitrary graphs. Therefore, it is untrivial to draw a comparison between TDL and this family of logics, and such a comparison would only be meaningful for 'enriched' logics. For now, we suffice to say that TDL allows direct expression of both action (dynamic) and logic (static) information simultaneously, in a manner that is by intention intuitive. By 'intuitive' we mean here that properties can be expressed directly rather than via some encoding which by a technical equivalence translates to the property that needed expressing, thus facilitating a conceptual grasp of the system to be specified at the level of the specification language. Without going into details here, it should also be remarked that an enhanced version of ACTL (let's just call it ACTL, for simplicity's sake) can be conceptualized that contains full propositional logic. Defining a mapping $\mathsf{f_f}$ from $\mathcal{L}_{\mathsf{TDL}}$ to (enhanced) $\mathcal{L}_{\mathsf{ACTL}}$, along with a mapping $\mathsf{f_m}$ from TDL-models to ACTL-models, in such a way that $\mathcal{M} \models_{\mathsf{TDL}} \phi \Rightarrow \mathsf{f_m}(\mathcal{M}) \models_{\mathsf{TDL}} \mathsf{f_f}(\phi)$ looks feasible[28] . If this is indeed the case, then (in a sense) TDL $\subseteq$ ACTL, and, given that f be not too complex, this fact can be used as a basis for TDL-model checking and such, applying the (available) ACTL-tools to 'translated' TDL-formulae. Whether or not f^{-1} can also be defined (giving rise to a full morphism) remains to be seen as of yet.

In future work, extensions of logics in the ACTL-family will be examined that are comparable with TDL regarding suitedness for practical (specification) purposes as well as expressiveness.

[28] Broersen (1997) defines the typical TDL operators in other logics.

11.3 Process Logics

In yet another framework, Harel & Singerman (1997) propose a logic called *Process Logic* in which graphs serve to provide a semantics for a language in which actions and propositions are dealt with in a uniform manner. This logic, however, is fundamentally different from usual logics, like TDL. For example, its operators do not constitute a boolean algebra. This makes the PL-family of logics (PL and a few predecessors) rather exotic, and hard to compare. For theoretical purposes, however, it is of interest to consider it with concepts like expressibility and decidability in mind: these issues too will receive more attention in future work.

12 Conclusion and Future Work

A logic called TDL is proposed, in which the usual logical formulae occur as well as dynamic (action) information. Although propositional, it is relatively straightforward to make it into a first-order (predicate) logic.

Things that can be seen as shortcomings of TDL as is, are the fact that at present only *atomic* actions have received a semantics: since, in the practice of writing specifications, it is desirable to express such things as "I'll keep driving until I get tired, or lose interest, or run out of gas": while in some sense these things can be expressed indirectly (i.e. by using the proposition 'I've run out of gas' instead of the action 'to run out of gas') in this case, this is still not satisfactory.

Additionally, we focused exclusively on semantic issues, neglecting questions of axiomatization, and the issues an axiomatization would give rise to: soundness, correctness. It must also not be left unmentioned here that in some context decidability – and, by extension, complexity – issues can be very relevant, or even essential. We will pay attention to these matters on a by-need basis in future work: when they need to be dealt with these matters will be addressed.

An interesting issue to be investigated into is how the proposed logic compares in detail to other logics, akin in spirit. This question needs to be expressed on multiple levels: it is of interest how they *technically* relate. Expressiveness is a key issue here, but decidability and (again) complexity play an important role too.

However, for practical purposes a few other points of comparison are relevant. However technically adequate, if a logic has formulae that are cryptic, hard to understand and even worse to come up with, it is of little

use in the practice of writing specifications.

Taking this point even further, a logic that is used in the process of writing system specification does not function in isolation: it must *fit*, or connect to, its surroundings. In a framework of design only a specification formalism receives its meaning and usefulness. However (technically) elegant a logic may be, it will be of little use unless it integrates well in the general process of systems development. Those developing the (formal part of) a system specification need to be able to use it, and use it as a part of the entire process of design, as seamless as possible. For any specification logic, its usability in practical contexts is therefore a relevant issue.

Future work will address these matters: we will examine if, and how, compound actions can be integrated, and detailed comparisons will be made with a number of existing logics, particularly the ones already mentioned briefly.

Also, a few trial specifications will be undertaken using TDL – or the logic it evolves into as experience with it matures. Having done so, then, the logic itself will once again be placed under scrutiny, and hopefully the iteration of this process will give rise to a logic that is as intuitive, expressive, direct as possible, that fits well in the practice of writing formal specifications in the context of designing complex systems.

References

Barringer, H., Fisher, M., Gabbay, D., Owens, R. & Reynolds, M. (1996), *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd.

Broersen, J. (1997), 'Expressing temporal (and deontic?) properties in a logic that generalizes PDL'. Under preparation, preliminary title.

De Nicola, R. & Vaandrager, F. (1990), Action versus state based logics for transition systems, *in* I. Guessarian, ed., 'Semantics of Systems of Concurrent Processes', Springer, pp. 407–419. Lecture Notes in Computer Science 469.

De Nicola, R., Fantechi, A., Gnesi, S. & Ristori, G. (1993), An action-based framework for verifying logical and behavioural properties of concurrent systems, *in* 'Computer Networks and ISDN Systems', Elsevier Science Publishers, pp. 761–778.

Emerson, E. & Halpern, J. (1986), '"Sometimes" and "not never" revisited: on branching time versus linear time temporal logic', *J. ACM* **1**(33), 151–178.

Goldblatt, R. (1987), *Logics of Time and Computation*, number 7 *in* 'CSLI Lecture Notes', CSLI, Stanford, Ca.

Harel, D. & Singerman, E. (1997), 'Computation paths logic: An expressive, yet elementary, process logic'.

Middelburg, C. (1994), 'A simple language for expressing properties of telecommunication services and features'.

Wansing, H., ed. (1996), *Proof Theory of Modal Logic*, Institute of Logic and Philosophy of Science.