Completeness of an Action Logic for Timed Transition Systems

Fernando Náufel do Amaral, Edward Hermann Haeusler

Abstract

This paper defines an action logic featuring an operator that denotes necessary conditions, postconditions, and time bounds of actions in a timed computational transition system. Weak completeness of an axiomatization for the logic is proved.

1 Introduction

In [8], Segerberg introduced the notion of a "bringing-it-about" δ operator, which, when applied to a given proposition q, denotes the set of actions that bring about the truth of q. In this paper, we discuss the use of a similar operator in a logical framework having computational transition systems as its semantics.

Segerberg's δ operator has led to an extension of dynamic logic containing elements from logics of action, where δq denotes actions that lead to states where q holds. In [3, 4], the operator was modified in two different ways: the first extension was introduced to reason about preconditions as well as postconditions, with the dyadic form $p\delta q$. The second extension, concerned with real-time applications, added a minimal and a maximal delay for the occurrence of each action: here, the action term $p_l\delta^u q$ denotes the actions that achieve q in less than u units of time, provided they were enabled in states satisfying p for at least l units of time.

In [3, 4], this last extension was dubbed RETOOL (Real-Time Object-Oriented Logic, in reference to the ultimate goal of achieving a formalization of real-time object-oriented software). The exact denotation of an action term $p\delta q$, however, was not agreed upon. Two

different semantics have been proposed for RETOOL, and this paper presents yet a third one: in [3], the term $p\delta q$ denotes all actions that have p as their *enabling condition* (i.e., actions that are enabled if and only if p holds); in [4], $p\delta q$ was defined as denoting all actions a such that, if a starts from a state where p holds, then it achieves q; in this paper, we present a definition of $p\delta q$ where this term denotes all actions having p as a *necessary condition*.

2 Defining RETOOL

2.1 The Language

We assume given a set A of attribute symbols, a set Γ of action symbols, an infinite totally ordered set (TIME, $\leq$), with minimum 0, a constant ∞ such that $\infty \notin$ TIME and $t \leq \infty, \forall t \in$ TIME. Also available are countably many constants, one for each member of TIME. An adequate theory of (TIME $\cup\{\infty\}$, $\leq$) is assumed to be contained in the logic (but this theory is not made explicit here). The syntax of RETOOL is:

State propositions (SP):	$p ::= a \mid \neg p \mid p \rightarrow p'$;
Action terms (AT):	$t ::= g \mid p_l \delta^u q$;
Formulae:	$\phi ::= a \mid t_1 \supset t_2 \mid \neg\phi \mid \phi \rightarrow \phi' \mid [t]\phi \mid [\,]p.$

where $a \in A$, and $g \in \Gamma$, and $p, q \in SP$, and $l \in$ TIME, and $u \in$ TIME $\cup$ { ∞ }, and $l \leq u$, and $t, t_1, t_2 \in AT$.

There are also two unary function symbols, l and u, which can be applied to action terms to yield their time bounds (i.e., elements of TIME $\cup\{\infty\}$ – see below). The time bounds $l(g)$ and $u(g)$ of a primitive action symbol g are of an extra-logical nature; the time bounds $l(p_x\delta^y q)$ and $u(p_x\delta^y q)$ of an action term built with the δ operator are, respectively, the constants x and y, and can be considered abbreviations thereof.

2.2 Semantics

The semantics of RETOOL is defined over structures that are based on the notion of *timed transition systems* [6]: given a set A of attribute symbols and a set Γ of action symbols, a *timed frame* $\mathcal{F}$ for A and Γ is

a sextuple $(W, \rightarrow, l, u, I, w_0)$, where W is a set of states; for each $g \in \Gamma$, $\stackrel{g}{\rightarrow} \subseteq W \times W$ is the transition relation for action g; l maps each $g \in \Gamma$ to an element $l(g) \in$ TIME; u maps each $g \in \Gamma$ to an element $u(g) \in$ TIME $\cup\{\infty\}$ such that $u(g) \geq l(g)$; $I : A \rightarrow 2^W$ is an interpretation of the attributes, where each $a \in A$ is assigned the set of worlds where a is true; and w_0 is the initial state.

Every action $g \in \Gamma$ has a lower bound $l(g)$ and an upper bound $u(g)$. Formally, lower and upper bounds are defined through the use of the notion of *computation*:

A *timed state sequence* [6] for a timed frame is a pair $\rho =< \sigma, \mathrm{T} >$, where σ is an infinite sequence of states ($\sigma_i \in W$) and T is an infinite sequence of corresponding times ($\mathrm{T}_i \in$ TIME), satisfying:

1. for all $i \geq 0$, either $\mathrm{T}_{i+1} = \mathrm{T}_i$, or ($\mathrm{T}_{i+1} > \mathrm{T}_i$ and $\sigma_{i+1} = \sigma_i$).
2. for every $t \in$ TIME, there is $i \geq 0$ such that $\mathrm{T}_i \geq t$.

A *computation* [6] is a timed state sequence $< \sigma, \mathrm{T} >$ such that

1. σ is a computation of the underlying transition system, i.e., for every $i \geq 0$, there is a transition $\sigma(i)$ such that $\sigma_i \stackrel{\sigma(i)}{\rightarrow} \sigma_{i+1}$;
2. (lower bound): for every $i \geq 0$ in the domain of σ, there is a $j \leq i$ such that $\mathrm{T}_i - \mathrm{T}_j > l(\sigma(i))$ and $\sigma(i)$ is enabled in every state σ_k for $j \leq k \leq i$.
3. (upper bound): for every $g \in \Gamma$ and $i \geq 0$, there is $j \geq i$ with $\mathrm{T}_j - \mathrm{T}_i \leq u(g)$ such that either g is not enabled at σ_j or $g = \sigma(j)$.

The denotation of a state proposition p in a timed frame $\mathcal{F}$ is the set of states defined as follows:

$$\begin{array}{ll} [\![a]\!] & = I(a); \\ [\![\neg p]\!]^{\mathcal{F}} & = W \setminus [\![p]\!]^{\mathcal{F}}; \\ [\![p \rightarrow p']\!]^{\mathcal{F}} & = (W \setminus [\![p]\!]^{\mathcal{F}}) \cup [\![p']\!]^{\mathcal{F}}. \end{array}$$

The denotation of an action term t in a timed frame $\mathcal{F}$ is the set of transitions defined as follows (where $en(g)$ is the set of states where g is enabled):

$$
\begin{array}{lcl}
[\![g]\!]^{\mathcal{F}} & = & \{(w,w') \mid w \xrightarrow{g} w'\}; \\
[\![p_l\delta^u q]\!]^{\mathcal{F}} & = & \{(w,w') \mid \exists g \in \Gamma \ [(w \xrightarrow{g} w') \ \wedge \ en(g) \subseteq [\![p]\!]^{\mathcal{F}} \ \wedge \\
 & & \forall v, v'((v \xrightarrow{g} v') \Rightarrow (w' \in [\![q]\!]^{\mathcal{F}})) \ \wedge \\
 & & (l \leq l(g) \leq u(g) \leq u)]\}.
\end{array}
$$

Finally, the satisfaction of a formula by a timed frame $\mathcal{F}$ at a state w is defined by:

$$
\begin{array}{ll}
\mathcal{F}, w \models p & \text{iff } w \in [\![p]\!]^{\mathcal{F}}; \\
\mathcal{F}, w \models (t_1 \supset t_2) & \text{iff } [\![t_1]\!]^{\mathcal{F}} \subseteq [\![t_2]\!]^{\mathcal{F}}; \\
\mathcal{F}, w \models \neg\phi & \text{iff not } \mathcal{F}, w \models \phi; \\
\mathcal{F}, w \models \phi \rightarrow \phi' & \text{iff } \mathcal{F}, w \models \phi \text{ implies } \mathcal{F}, w \models \phi'; \\
\mathcal{F}, w \models [t]\phi & \text{iff } \mathcal{F}, w' \models \phi \text{ for every } w' \text{ such that } (w,w') \in [\![t]\!]^{\mathcal{F}}; \\
\mathcal{F}, w \models [\ ]p & \text{iff } \mathcal{F}, w_0 \models p.
\end{array}
$$

2.3 Axiomatization

Figure 1 shows an adequate axiomatization of RETOOL. Λ represents a set of RETOOL formulae, the derivability relation $\vdash$ is defined in the usual manner, and $enabled(t)$ is an abbreviation for the formula $\neg[t]-$.

2.4 Comments

The present work departs from the semantics of the δ operator given in [4]. There, $[\![p_l\delta^u q]\!]$ is defined as the set of all transitions (w,w') such that $w \in [\![p]\!]$ implies $w' \in [\![q]\!]$. This semantics mirrors the meaning of the Hoare triples $\{p\}g\{q\}$ (see [7]), namely that all terminating runs of program g starting in states satisfying p will end in states satisfying q (and there is no guarantee about runs that start in states not satisfying p). According to this view, p could be considered a precondition of program g, with q a relative postcondition.

Now, the semantics presented here requires that for (w,w') to be denoted by $p_l\delta^u q$, it must be the case that $w \xrightarrow{g} w'$ for some g such that $en(g) \subseteq [\![p]\!]$. The requirement that p must be true in all worlds where g is enabled forces us to see p as a necessary condition of the program represented by g. It is simply impossible to have runs of g starting in states not satisfying p.

(PC)	All axioms and rules of propositional calculus
(K)	$[t](\phi \rightarrow \psi) \rightarrow ([t]\phi \rightarrow [t]\psi)$ $[\,](p \rightarrow q) \rightarrow ([\,]p \rightarrow [\,]q)$
(I)	$[\,]p \rightarrow [t][\,]p$ $[t][\,]p \rightarrow (enabled(t) \rightarrow [\,]p)$ $[\,]\neg p \leftrightarrow \neg[\,]p$
(N)	$\dfrac{\Lambda \vdash \phi}{\Lambda \vdash [t]\phi} \quad \dfrac{\Lambda \vdash p}{\Lambda \vdash [\,]p}$
(δ)	$\dfrac{\Lambda \vdash enabled(t) \rightarrow p \quad \Lambda \vdash [t]q \quad \Lambda \vdash l \leq l(t) \leq u(t) \leq u}{\Lambda \vdash t \supset p_l \delta^u q}$
(S1)	$t \supset t$
(S2)	$(t_1 \supset t_2) \rightarrow ((t_2 \supset t_3) \rightarrow (t_1 \supset t_3))$
(S3)	$(t_1 \supset t_2) \rightarrow ([t_2]\phi \rightarrow [t_1]\phi)$
(NC)	$(t \supset p_l \delta^u q) \rightarrow (enabled(t) \rightarrow p)$
(Post)	$(t \supset p_l \delta^u q) \rightarrow ([t]q)$
(Bounds)	$(t \supset p_l \delta^u q) \rightarrow (enabled(t) \rightarrow l \leq l(t) \leq u(t) \leq u)$

Figure 1: An Axiomatization of RETOOL

This "necessary-condition" semantics of the δ operator seems to us more appropriate to reason about timed transition systems as abstract models of complex real-time reactive systems. The "functionality" of an action, in the sense captured by Hoare triples, can still be expressed using action modalities. I.e., the Hoare triple $\{p\}g\{q\}$ is closely related to the RETOOL formula $p \rightarrow [g]q$.

3 Weak Completeness

We now construct a canonic model for RETOOL and proceed to a completeness proof for the axiomatization given above.

The set A^c of attribute symbols of the canonic model is arbitrary: $A^c = \{a_1, a_2, ...\}$. For the construction of the set Γ^c of action symbols, we introduce the following definitions:

Definition 3.1 *For each* $p \in SP$, $\tilde{p} = \{r \mid \vdash p \leftrightarrow r\}$.

Definition 3.2 $\Gamma^c = \{\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \mid p, q \in SP \setminus \tilde{-}, l \in TIME, u \in TIME \cup \{\infty\}\}$.

We define here an action symbol for each quadruple (p, q, l, u) composed of satisfiable state propositions p and q (up to tautological equivalence) and time bounds l and u. In order to ensure that these action symbols have the desired meaning, we define the following set Σ of RETOOL formulae:

Definition 3.3 $\Sigma = \{(\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \supset p_l\delta^u q) \wedge (p_l\delta^u q \supset \text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}) \mid p, q \in SP \setminus \tilde{-},\ l \in TIME\ ,\ u \in TIME \cup \{\infty\}\}$.

Note that in every model of Σ, each action symbol “$\tilde{p}_l\delta^u\tilde{q}$” will function as a "witness" to action term $p_l\delta^u q$, the denotation of the symbol corresponding exactly to the denotation of the term.

More extra-logical information is necessary to build the canonic model: we must specify the initial state through a set Θ of formulae. If we want state propositions $p_1, p_2, ...$ to be true at the initial state w_0, we define $\Theta = \{[\ ]p_1, [\ ]p_2, ...\}$.[1] Obviously, the set $\{p_1, p_2, ...\}$ must be consistent.

We also introduce some special notation to refer to "demodalized" formulae, where w is a set of formulae and t is any action term:

Definition 3.4 $w \setminus [t] = \{\phi \mid [t]\phi \in w\}$.

[1] Actually, this leads to a *class* of canonic models, one for each choice of Θ.

Definition 3.5 (The canonic model) *Given the sets A^c, Σ, Θ and Γ^c, we define the canonic model as $M^c = (W^c, \{\stackrel{g}{\rightarrow} | \ g \in \Gamma^c\}, l^c, u^c, I^c, w_0^c)$, where:*

W^c	$= \{\Phi \mid \Phi$ is a maximally consistent set containing $\Sigma \cup \Theta\}$;
$\stackrel{\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}}{\rightarrow}$	$= \{(w, w') \in W^c \times W^c \mid w \setminus [\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}] \subseteq w'\}$;
l^c	is such that $l^c(\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}) = l$ for every $\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \in \Gamma^c$;
u^c	is such that $u^c(\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}) = u$ for every $\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \in \Gamma^c$;
I^c	is such that $I^c(p) = \{w \in W^c \mid p \in w\}$ for every $p \in SP$;
w_0^c	is some world in W^c satisfying $\Theta \setminus [\,]$.

We now want to prove weak completeness:

$$\forall \phi : \quad \models \phi \quad \Rightarrow \quad \vdash \phi$$

This is equivalent to showing that every consistent RETOOL formula is satisfiable. Now, every consistent formula is a member of some maximally consistent set of formulae (see [5], for instance, for definitions and results). We have built the canonic model so that its set of states includes all the maximally consistent sets we are interested in. We now show that the canonic model indeed satisfies all the formulae contained in its states. This is done via the Coincidence Lemma:

$$\forall w \in W^c, \ \ \forall \phi : \quad \phi \in w \quad \Leftrightarrow \quad M^c, w \models \phi$$

This is proved by induction over the formation of ϕ. The following lemmas are used in the proof:[2]

Lemma 3.1 *For all $w \in W^c$, for all $p, q \in SP \setminus \tilde{-}$, for all $l \in TIME$, $u \in TIME \cup \{\infty\}$, for all $t \in AT$, and for all formulae ϕ:*

$$\begin{array}{lcl} w \vdash t \supset p_l\delta^u q & \Leftrightarrow & w \vdash t \supset \text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \\ w \vdash p_l\delta^u q \supset t & \Leftrightarrow & w \vdash \text{“}\tilde{p}_l\delta^u\tilde{q}\text{”} \supset t \\ w \vdash [p_l\delta^u q]\phi & \Leftrightarrow & w \vdash [\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}]\phi \end{array}$$

Lemma 3.2 *For all $w \in W^c$, for all $p, q \in SP \setminus \tilde{-}$, for all $l \in TIME, u \in TIME \cup \{\infty\}$:* $\quad w \in en(p_l\delta^u q) \quad \Leftrightarrow \quad w \vdash enabled(p_l\delta^u q)$.

Lemma 3.3 *In the canonic model, $[\![\text{“}\tilde{p}_l\delta^u\tilde{q}\text{”}]\!] = [\![p_l\delta^u q]\!]$.*

[2]Detailed proofs of these and all other relevant lemmas can be found in [1].

4 Concluding Remarks

Work in progress includes the development of an automated theorem-proving strategy for RETOOL and the use of a combination of RETOOL with a temporal logic as a specification and verification tool for real-time reactive systems (see [2]).

References

[1] Amaral, F.N., **RETOOL: Uma Lógica de Ações para Sistemas de Transição Temporizados**, M.Sc. Dissertation, Dept. of Informatics, PUC-RJ, Brazil, 2000.

[2] Amaral, F.N. and Haeusler, E.H. *A Logic-based Approach for Real-Time Object-oriented Software Development*, in **Revista de Informática Teórica e Aplicada** VII(1), UFRGS, Brazil, 2000.

[3] Carvalho, S., Fiadeiro, J. and Haeusler, E.H., *A Formal Approach to Real-Time Object-Oriented Software*, in **Proc. 22nd IFAC/IFIP Workshop on Real-Time Programming WRTP'97**, Elsevier 1997.

[4] Fiadeiro, J. and Haeusler, E.H., *Bringing It About On Time (Extended Abstract)*, in **Proc. I IMLLAI, Fortaleza, CE, Brazil**, 1998.

[5] Goldblatt, R., **Logics of Time and Computation**, CSLI Lecture Notes 7, CSLI, 1992.

[6] Henzinger, T., Manna, Z. and Pnuelli, A., *Timed Transition Systems*, in **Real Time: Theory in Practice**, LNCS 600, Springer-Verlag, 1992.

[7] Hoare, C.A.R., *An Axiomatic Basis for Computer Programming*, in **Comm. ACM** 12, 1967.

[8] Segerberg, K., *Bringing It About*, in **Journal of Philosophical Logic** 18, 1989.

PUC-RJ (Catholic University of Rio de Janeiro) – Brazil
{fnaufel, hermann}@inf.puc-rio.br