A Complete Axiomatisation for Timed Automata

Huimin Lin
Laboratory for Computer Science
Inst. of Software, Chinese Academy of Sciences
Email: lhm@ios.ac.cn

Wang Yi
Dept. of Computer Systems
Uppsala University
Email: yi@csd.uu.se

Abstract

A proof system of timed bisimulation equivalence for timed automata is presented, based on a CCS-style regular language for describing timed automata. It consists of the standard monoid laws for bisimulation and a set of inference rules. The judgments of the proof system are *conditional equations* of the form $\phi \rhd t = u$ where ϕ is a clock constraint and t, u are terms denoting timed automata. The proof of the completeness result relies on the notion of *symbolic timed bisimulation*, adapted from the work on value–passing processes.

1 Introduction

The last decade has seen a growing interest in extending various concurrency theories with timing constructs so that real-time aspects of concurrent systems can be modeled. Among them timed automata [AD94] has stood out as a fundamental model for real-timed systems.

A timed automaton is a finite automaton extended with a finite set of real-valued *clock variables*. A node of a timed automata is associated with an *invariant* constraint on the clock variables, while an edge is decorated with a clock constraint, an action label, and a subset of clocks to be reset after the transition. At each node a timed automaton may perform two kinds of transitions: it may let time pass for any amount (a delay transition), as long as the invariant is satisfied, or choose an edge whose constraint is met, make the move, reset the relevant clocks to zero, and arrive at the target node (an action transition). Although a timed automaton has only finite number of nodes, due to (real-valued) delay transitions it typically exhibits infinite-state behaviour. Two timed automata are *timed bisimilar* if they can match each other's action transitions as well as delay transitions, and their residuals remain timed bisimilar. The explicit presence of clock variables and resetting, features that mainly associated with the so-called "imperative languages", distinguishes timed automata from process calculi such as CCS, CSP and their timed extensions which are "applicative" in nature and therefore more amenable to axiomatisation. By now

most theoretical aspects of timed automata have been well studied, but they still lack a satisfactory algebraic theory.

In this paper we shall develop a complete axiomatisation for timed automata, in the form of an inference system, in which the equalities between pairs of timed automata that are timed bisimilar can be derived. To this end we first propose a language, in CCS style, equipping it with a symbolic transitional semantics in such a way that each term in the language denotes a timed automaton. The language has a conditional construct $\phi{\rightarrow}t$, read "if ϕ then t", an action prefixing $a(\mathbf{x}).t$, meaning "perform the action a, reset the clocks in $\mathbf{x}$ to zero, then behave like t", and a recursion $\mathbf{fix}Xt$ which allows infinite behaviours to be described. The proof system consists of a set of inference rules and the standard monoid laws for bisimulation. Roughly speaking the monoid laws characterize bisimulation, while the inference rules deal with specific constructs in the language. The judgments of the inference system are of the form

$$\phi \rhd t = u$$

where ϕ is a time constraint and t, u are terms. Intuitively it means: t and u are timed bisimilar over clock evaluations satisfying ϕ. A typical inference rule takes the form:

$$\text{GUARD} \quad \frac{\phi \wedge \psi \rhd t = u \quad \phi \wedge \neg\psi \rhd \mathbf{0} = u}{\phi \rhd (\psi{\rightarrow}t) = u}$$

It performs a case analysis on the constraint ψ: $\psi{\rightarrow}t$ behaves like t when ψ is true, and like the inactive process $\mathbf{0}$ otherwise. Note that the guarding constraint ψ of $\psi{\rightarrow}t$ in the conclusion is *part of the object language* describing timed automata, while in the premise it is shifted to the condition part of the judgment in our *meta language* for reasoning about timed automata.

A crucial rule, as might be expected, is the one for action prefixing:

$$\text{ACTION} \quad \frac{\phi{\downarrow_{\mathbf{x}}}{\Uparrow} \rhd t = u}{\phi \rhd a(\mathbf{x}).t = a(\mathbf{x}).u}$$

Here $\downarrow_{\mathbf{x}}$ and $\Uparrow$ are postfixing operations on clock constraints. $\phi{\downarrow_{\mathbf{x}}}{\Uparrow}$ is a clock constraint obtained from ϕ by first setting the clocks in $\mathbf{x}$ to zero (operator $\downarrow_{\mathbf{x}}$), then removing up-bounds on all clocks of ϕ (operator $\Uparrow$). Readers familiar with Hoare Logic may notice some similarity between this rule and the rule dealing with assignment there:

$$\{P[e/x]\}\ x := e\ \{P\}$$

But here the operator $\downarrow_{\mathbf{x}}$ is slightly more complicated than substitution with zero, because clocks are required to increase uniformly. Also we need $\Uparrow$ to allow time to pass indefinitely.

A standard way to reasoning with recursion is to use, apart from the usual rule for folding/unfolding recursions, the following *unique fixpoint induction*:

$$\text{UFI} \quad \frac{t = u[t/X]}{t = \mathbf{fix}Xu} \quad X \text{ guarded in } u$$

This rule was adopted in [Mil84] for a complete axiomatisation of bisimulation equivalence for regular pure-CCS. Here we use it in a quite different context: terms in our setting normally contain clock variables, namely they are *open terms*. In spite of this, it turns out that this rule is still sound and sufficient for a complete axiomatisation of regular behaviour, though the proof is slightly more complicated than in the pure calculi.

The completeness proof relies on the introduction of the notion of *symbolic timed bisimulation*, $t \sim^{\phi} u$, which captures timed bisimulation in the following sense: $t \sim^{\phi} u$ if and only if $t\rho$ and $u\rho$ are timed bisimilar for any clock evaluation ρ satisfying ϕ. Following [Mil84], to show that the inference system is complete, that is $t \sim^{\phi} u$ implies $\vdash \phi \rhd t = u$, we first transform t and u into *standard equation sets* which are the syntactical representations of timed automata. We then construct a product equation set out of the two and prove that t and u both satisfy this new equation set, by exploiting the assumption that t and u are symbolically timed bisimilar. Due to the presence of

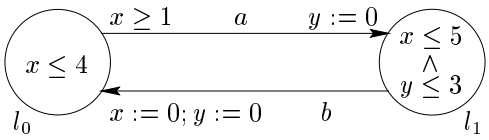


Figure 1: A Timed Automaton.

Consider the timed automaton of Figure 1. It has two control nodes l_0 and l_1 and two clocks x and y. A *tate* of the automaton is of the form $(l, < s, t >)$, where l is a control node and s and t are non–negative reals giving the values of x and y. Assuming that the automaton starts to operate in the state $(l_0, < 0, 0 >)$, it may stay in node l_0 for any amount of time, as long as the invariant $x \leq 4$ of l_0 is satisfied. During this time the values of x and y increase uniformly, at the same rate. Thus from the initial state, all states of the form $(l_0, < t, t >)$ with $0 \leq t \leq 4$ are reachable, but only at the states $(l_0, < t, t >)$, where $t \geq 1$, the edge from l_0 to l_1 is enabled. When following the edge from l_0 to l_1 the action a is performed to synchronize with the environment and the clock y is reset to 0 leading to states of the form $(l_1, < t, 0 >)$ where $t \geq 1$.

For the formal definition, we assume a finite set $\mathcal{A}$ for synchronization actions and a finite set $\mathcal{C}$ for real-valued clock variables. We use a, b etc. to range over $\mathcal{A}$ and x, y etc. to range over $\mathcal{C}$. We use $\mathcal{B}(C)$, ranged over by ϕ, ψ etc., to denote the set of conjunctive formulas of atomic constraints in the form: $x_i \bowtie m$ or $x_i - x_j \bowtie n$, where $x_i, x_j \in \mathcal{C}$, $\bowtie \in \{\leq, <, \geq, >\}$ and m, n are natural numbers. The elements of $\mathcal{B}(C)$ are called *clock con traint* .

Definition 2.1 *A timed automaton over action* $\mathcal{A}$ *and clock* $\mathcal{C}$ *i a*

$$\textsc{Action}\quad \frac{}{a(\mathbf{x}).t \xrightarrow{\mathsf{tt},a,\mathbf{x}} t} \qquad \textsc{Choice}\quad \frac{t \xrightarrow{b,a,\mathbf{x}} t'}{t + u \xrightarrow{b,a,\mathbf{x}} t'}$$

$$\textsc{Guard}\quad \frac{t \xrightarrow{,a,\mathbf{x}} t'}{\phi{\rightarrow}t \xrightarrow{\phi\wedge\ ,a,\mathbf{x}} t'} \qquad \textsc{Rec}\quad \frac{t[\mathbf{fix}Xt/X] \xrightarrow{b,a,\mathbf{x}} t'}{\mathbf{fix}Xt \xrightarrow{b,a,\mathbf{x}} t'}$$

$\phi \equiv \mathtt{tt}$ and $t \equiv a(\mathbf{x}).t' \xrightarrow{\mathtt{tt},a,\mathbf{x}} t'$. Then $(a(\mathbf{x}).t')\rho \xrightarrow{a} t'\rho\{\mathbf{x} := 0\}$ by ACTION and $\rho \models \phi$.

$\phi \equiv \phi' \wedge \psi$ and t

In the following, "atomic constraint" always means "atomic constraint over C with ceiling N". Note that given two timed automata there are only finite number of such atomic constraints. We shall use c to range over atomic constraints.

A constraint, or *zone*, is a boolean combination of atomic constraints. A constraint ϕ is consistent if there is some ρ such that $\rho \models \phi$. Let ϕ and ψ be two constraints. We write $\phi \models \psi$ to mean $\rho \models \phi$ implies $\rho \models \psi$ for any ρ. Note that the relation $\models$ is decidable.

A *region constraint*, or *region* for short, over n clock variables $x_1, \ldots, x_n$ is a consistent constraint containing the following atomic conjuncts:

> For each $i \in \{1, \ldots, n\}$ either $x_i = m_i$ or $m_i < x_i < m_i + 1$ or $x_i > N$;
>
> For each pair of $i, j \in \{1, \ldots, n\}$, $i \neq j$, such that both x_i and x_j are not greater than N, either $x_i - m_i = x_j - m_j$ or $x_i - m_i < x_j - m_j$ or $x_j - m_j < x_i - m_i$.

where the m_i in $x_i - m_i$ of the second clause refers to the m_i related to x_i in the first clause. In words, m_i is the integral part of x_i and $x_i - m_i$ its fractional part.

Given a finite set of clock variables C and a ceiling N, the set of region constraints over C is finite and is denoted $\mathcal{RC}_N^C$. In the sequel, we will omit the sub- and superscripts when they can be supplied by the context.

Fact 1 *Let ϕ be a region constraint. If $\rho \models \phi$ and $\rho' \models \phi$ then*

> *For all $i \in \{1, \ldots, n\}$, if $\rho(x_i) \leq N$ then $\lfloor \rho(x_i) \rfloor = \lfloor \rho'(x_i) \rfloor$.*
>
> *For any $i, j \in \{1, \ldots, n\}$, $i \neq j$,*
>
> - *$\{\rho(x_i)\} = \{\rho(x_j)\}$ iff $\{\rho'(x_i)\} = \{\rho'(x_j)\}$ and*
> - *$\{\rho(x_i)\} < \{\rho(x_j)\}$ iff $\{\rho'(x_i)\} < \{\rho'(x_j)\}$.*

where $\lfloor x \rfloor$ and $\{x\}$ are the integral and fractional parts of x, respectively.

That is, two valuations satisfying the same region constraint must agree on their integral parts as well as on the ordering of their fractional parts.

Lemma 3.1 *Suppose that ϕ is a region constraint and ψ a zone. Then either $\phi \Rightarrow \psi$ or $\phi \Rightarrow \neg\psi$.*

Proof: We first transform ψ into disjunctive normal form: $\psi = \bigvee_i \bigwedge_j e_{ij}$ where each e_{ij} is an atomic constraint. Now $\psi \wedge \phi = \bigvee_i \bigwedge_j (e_{ij} \wedge \phi)$. It is easy to see, by examining the possible forms of e_{ij}, that each $e_{ij} \wedge \phi$ is either equal to ϕ or false. Hence $\psi \wedge \phi$ is either equal to ϕ or false. In the former case we have $\phi \Rightarrow \psi$, and in the later case we get $\phi \Rightarrow \neg\psi$. □

According to this lemma, a region is either entirely contained in a zone, or is completely outside a zone. In other words, regions are the finest polyhedra that can be described by our constraint language.

Fact 2 *Let t, u be two terms with disjoint sets of clock variables and ϕ a region constraint over the union of the two clock sets. Suppose that both ρ and ρ' satisfy ϕ. Then $t\rho \sim u\rho$ iff $t\rho' \sim u\rho'$.*

A *canonical* constraint is a disjunction of regions. Given a

where $\Uparrow'$ is defined

equalities between clock variables (the e_{ij} component in the above definition), which guarantees the "same rate" requirement when such constraints are over the union of the two clock sets.

Given a constraint ϕ, a finite set of constraints Φ is called a *ϕ-partition* if $\bigvee \Phi = \phi$. A ϕ-partition Φ is called *finer* than another such partition Ψ if Φ can be obtained from Ψ by decomposing some of its elements. By the corollary to Lemma 3.1, $\mathcal{RC}(\phi)$ is a ϕ-partition, and is the finest such partition. In particular, if ϕ is a region constraint then $\{\phi\}$ is the only partition of ϕ.

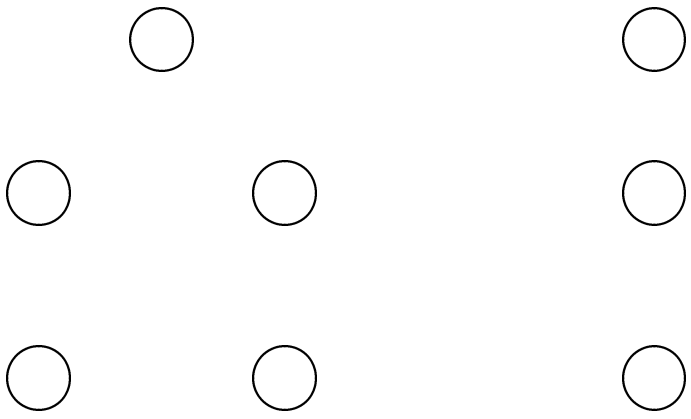
Definition 3.4 A constraint indexed family of symmetric relations over terms $\mathbf{S} = \{ S^\phi \mid \phi \text{ is } \Uparrow\text{-closed} \}$ is a symbolic timed bisimulation if $(t,u) \in S^\phi$ implies

1. $\phi \models Inv(t) \Leftrightarrow Inv(u)$ and
2. whenever $t \xrightarrow{\psi,a,\mathbf{x}} t'$ then there is a $Inv(t) \wedge \phi \wedge \psi$-partition Φ such that for each $\phi' \in \Phi$ there is $u \xrightarrow{\psi',a,\mathbf{y}} u'$ for some ψ', $\mathbf{y}$ and u' such that $\phi' \Rightarrow \psi'$ and $(t',u') \in S^{\phi'\downarrow_{\mathbf{xy}}\Uparrow}$.

We write $t \sim^\phi u$ if $(t,u) \in S^\phi \in \mathbf{S}$ for some symbolic bisimulation $\mathbf{S}$. □

Note that there is no clause for delay transitions in the definition, because delays are encoded in the $\Uparrow$-closeness property of the indexing constraints.

The use of a partition when matching a symbolic transition is essential. Without it we will not be able to characterise timed bisimulation using symbolic transitions. For example, consider the two timed automata t_1 and t_2 below (we have omitted the empty resets). They are apparently timed bisimilar. But the symbolic transition $t_2 \xrightarrow{\mathtt{tt},a,\{\}}$ can not be entirely matched by either of the two symbolic transitions from t_1. We must use a partition, say $\{x \leq 1,\ x > 1\}$: t_1 can match the symbolic transition from t_2 using its left branch over the constraint $x \leq 1$, and the right branch over $x > 1$.



Proof: ($\Longrightarrow$) Assume $(t,u) \in S^\phi \in \mathbf{S}$ for some symbolic bisimulation $\mathbf{S}$. Define

$$R = \{\, (t\rho, u\rho) \mid \text{there exists some } \phi \text{ such that } \rho \models \phi \text{ and } (t,u) \in S^\phi \in \mathbf{S} \,\}$$

We show R is a timed bisimulation. Suppose $(t\rho, u\rho) \in R$, i.e. there is some ϕ such that $\rho \models \phi$ and $(t,u) \in S^\phi$. By the first claus in Definition 3.4, we have $\rho \models Inv(t)$ if and only if $\rho \models Inv(u)$.

> $t\rho \xrightarrow{a} t'\rho'$. By Lemma 2.3 there are ψ, $\mathbf{x}$ such that $\rho \models \psi \wedge Inv(t)$, $\rho' = \rho\{\mathbf{x} := 0\}$ and $t \xrightarrow{,a,\mathbf{x}} t'$. So there is a $\phi \wedge \psi$-partition Φ with the properties specified in Definition 3.4. Since $\rho \models \phi \wedge \psi$, $\rho \models \phi'$ for some $\phi' \in \Phi$. Let $u \xrightarrow{',a,\mathbf{y}} u'$ be the symbolic transition associated with this ϕ', as guaranteed by Definition 3.4. Then $\phi' \Rightarrow \psi'$ and $(t',u') \in S^{\phi'\downarrow_{\mathbf{xy}}\Uparrow}$. Since $\rho \models \psi' \wedge Inv(u)$, $u\rho \xrightarrow{a} u'\rho\{\mathbf{y} := 0\}$. By Lemma 3.2, $\rho\{\mathbf{xy} := 0\} \models \phi'\downarrow_{\mathbf{xy}}$. By Lemma 3.3, $\rho\{\mathbf{xy} := 0\} \models \phi'\downarrow_{\mathbf{xy}}\Uparrow$. Therefore $(t'\rho\{\mathbf{xy} := 0\}, u'\rho\{\mathbf{xy} := 0\}) \in R$. Since $t'\rho\{\mathbf{xy} := 0\} \equiv t'\rho\{\mathbf{x} := 0\}$ and $u'\rho\{\mathbf{xy} := 0\} \equiv u'\rho\{\mathbf{y} := 0\}$, this is the same as $(t'\rho\{\mathbf{x} := 0\}, u'\rho\{\mathbf{y} := 0\}) \in R$.

> $t\rho \xrightarrow{d} t(\rho + d)$. Since ϕ is $\Uparrow$-closed, $\rho + d \models \phi$. Then $\rho + d \models Inv(u)$ and hence $u\rho \xrightarrow{d} u(\rho + d)$. Therefore $(t(\rho + d), u(\rho + d)) \in R$.

($\Longleftarrow$) Assume $t\rho \sim u\rho$ for any $\rho \models \phi_0 \wedge Inv(t) \wedge Inv(u)$, *i.e.* $(t\rho, u\rho) \in R$ for some timed bisimulation R, we show $t \sim^{\phi_0} u$ as follows. For each $\Uparrow-$closed ϕ, define

$$S^\phi = \{\, (t,u) \mid \forall \phi' \in \mathcal{RC}(\phi),\ (t\rho, u\rho) \in R \text{ for any } \rho \models \phi' \wedge Inv(t) \wedge Inv(u) \,\}$$

and let $\mathbf{S} = \{\, S^\phi \mid \phi \text{ is } \Uparrow - \text{closed} \,\}$. Then $(t,u) \in S^{\phi_0}$. $\mathbf{S}$ is well-defined because of Fact 2. We show $\mathbf{S}$ is a symbolic bisimulation. Suppose $(t,u) \in S^\phi$. Consider any $\phi' \in \mathcal{RC}(\phi)$. There exists $\rho \models \phi' \wedge Inv(t) \wedge Inv(u)$ such that $(t\rho, u\rho) \in R$. Since ϕ' is a region it must be entirely contained in $Inv(t) \wedge Inv(u)$, i.e.

EQUIV $\dfrac{}{t = t}$ $\quad \dfrac{\phi \rhd t = u}{\phi \rhd u = t}$ $\quad \dfrac{\phi \rhd t = u \quad \phi \rhd u = v}{\phi \rhd t = v}$

AXIOM $\dfrac{}{t = u}$ $\quad t = u$ an axiom instance

They are so-called "structural rules" used to "glue" pieces of derivation together.

Taking $\phi_1 = \phi_2$ PARTITION specialises to a useful rule

$$\text{CONSEQUENCE} \quad \frac{\phi_1 \rhd t = u}{\phi \rhd t = u} \quad \phi \models \phi_1$$

Let us write $\vdash \phi \rhd t = u$ to mean $\phi \rhd t = u$ can be derived from this proof system.

Some useful properties of the proof system are summarised in the following proposition:

Proposition 4.1 *1.* $\vdash \phi{\rightarrow}(\psi{\rightarrow}t) = \phi \wedge \psi{\rightarrow}t$

2. $\vdash t = t + \phi{\rightarrow}t$

3. If $\phi \models \psi$ *then* $\vdash \phi \rhd t = \psi{\rightarrow}t$

4. $\vdash \phi \wedge \psi \rhd t = u$ *implies* $\vdash \phi \rhd \psi{\rightarrow}t = \psi{\rightarrow}u$

5. $\vdash \phi{\rightarrow}(t + u) = \phi{\rightarrow}t + \phi{\rightarrow}u$

6. $\vdash \phi{\rightarrow}t + \psi{\rightarrow}t = \phi \vee \psi{\rightarrow}t$

7. For any t *and* u, $\vdash \{\mathsf{ff}\}t = \{\mathsf{ff}\}u$

Proof: We only give proofs for *1*, *4* and *7*, leaving the others to the readers.

We first prove a le/R12308a:

$$\begin{array}{llll} \text{S1} & X + \mathbf{0} = X & \text{S2} & X + X = X \\ \text{S3} & X + Y = Y + X & \text{S4} & (X + Y) + Z = X + (Y + Z) \end{array}$$

Figure 5: The Equational Axioms

which can be settled by EQUIV (plus CONSEQUENCE) and ABSURD, respectively.
4. By GUARD, $\vdash \phi \rhd \psi{\rightarrow}t = \psi{\rightarrow}u$ can be reduced to

$$\phi \wedge \psi \rhd t = \phi{\rightarrow}u \quad \text{and} \quad \phi \wedge \neg\psi \rhd \mathbf{0} = \phi{\rightarrow}u$$

The second subgoal is an instance of (1). For the first one we apply GUARD again obtaining

$$(\phi \wedge \psi) \wedge \psi \rhd t = u \quad \text{and} \quad (\phi \wedge \psi) \wedge \neg\psi \rhd t = \mathbf{0}$$

Now the first subgoal follows from the hypothesis and the second from ABSURD.
7. It is sufficient to prove $\vdash \{\mathsf{ff}\}t = \{\mathsf{ff}\}\mathbf{0}$ for any t. By INV this can be reduced to $\vdash \mathsf{ff} \rhd t = \{\mathsf{ff}\}\mathbf{0}$ and $\vdash \neg\mathsf{ff} \rhd \{\mathsf{ff}\}\mathbf{0} = \{\mathsf{ff}\}\mathbf{0}$. The first subgoal is settled by ABSURD while the second by EQUIV. □

The following lemma shows how to "push" a condition through an action prefix:

Lemma 4.2 $\vdash \phi \rhd a(\mathbf{x}).\{\psi\}t = a(\mathbf{x}).\{\psi\}\phi{\downarrow_{\mathbf{x}}}{\Uparrow}{\rightarrow}t$.

Proof: By ACTION this can be reduced to

$$\phi{\downarrow_{\mathbf{x}}}{\Uparrow} \rhd \{\psi\}t = \{\psi\}\phi{\downarrow_{\mathbf{x}}}{\Uparrow}{\rightarrow}t$$

An applications of INV gives two subgoals:

$$\phi{\downarrow_{\mathbf{x}}}{\Uparrow} \wedge \psi \quad \rhd \quad t = \{\psi\}\phi{\downarrow_{\mathbf{x}}}{\Uparrow}{\rightarrow}t \tag{4}$$
$$\phi{\downarrow_{\mathbf{x}}}{\Uparrow} \wedge \neg\psi \quad \rhd \quad \{\mathsf{ff}\}\mathbf{0} = \{\psi\}\phi{\downarrow_{\mathbf{x}}}{\Uparrow}{\rightarrow}t \tag{5}$$

Apply INV again to (4) we get

$$\phi{\downarrow_{\mathbf{x}}}{\Uparrow} \wedge \psi \wedge \psi \rhd t = \phi{\downarrow_{\mathbf{x}}}{\Uparrow}{\rightarrow}t \quad \text{and} \quad \phi{\downarrow_{\mathbf{x}}}{\Uparrow} \wedge \psi \wedge \neg\psi \rhd t = \{\mathsf{ff}\}\mathbf{0}$$

the first follows from Proposition 4.1.*3*, while the second from ABSURD.

(5) can be settled similarly by an application of INV followed by EQUIV and ABSURD. □

The UFI rule, as presented in Figure 4, is unconditional. However, a conditional version can be derived:

Proposition 4.3 *Suppose X is guarded in u. Then from $\vdash \phi \rhd t = u[\phi{\rightarrow}t/X]$ infer $\vdash \phi \rhd t = \mathbf{fix}X\phi{\rightarrow}u$.*

Proof: Assume $\vdash \phi \rhd t = u[\phi{\rightarrow}t/X]$. By Proposition 4.1.4 we have $\vdash \phi{\rightarrow}t = \phi{\rightarrow}u[\phi{\rightarrow}t/X]$, i.e.

$$\vdash \phi{\rightarrow}t = (\phi{\rightarrow}u)[\phi{\rightarrow}t/X]$$

Since X is guarded in u, it is also guarded in $\phi{\rightarrow}u$. By UFI, $\vdash \phi{\rightarrow}t = \mathbf{fix}X\phi{\rightarrow}u$. Hence

$$\begin{array}{rcl} \vdash \phi{\rightarrow}t & \stackrel{\text{REC}}{=} & (\phi{\rightarrow}u)[\mathbf{fix}X\phi{\rightarrow}u/X] \\ & = & \phi{\rightarrow}u[\mathbf{fix}X\phi{\rightarrow}u/X] \\ & = & \phi{\rightarrow}(\phi{\rightarrow}u)[\mathbf{fix}X\phi{\rightarrow}u/X] \\ & \stackrel{\text{REC}}{=} & \phi{\rightarrow}\mathbf{fix}X\phi{\rightarrow}u \end{array}$$

Therefore, by Proposition 4.1.4 again, $\vdash \phi \rhd t = \mathbf{fix}X\phi{\rightarrow}u$. □

The rule PARTITION has a more general form:

Proposition 4.4 *Suppose Ψ is a ϕ-partition and $\vdash \psi \rhd t = u$ for each $\psi \in \Psi$, then $\vdash \phi \rhd t = u$.*

Proof: By induction on the size of Ψ. The base case when Ψ contains only one element is trivial. For the induction step, assume the statement of the proposition holds for ϕ-partitions of size k and let $\Psi = \{\, \psi_i \mid 1 \leq i \leq k+1 \,\}$. Set $\Psi' = \{\, \neg\psi_{k+1} \wedge \psi_i \mid 1 \leq i \leq k \,\}$. Since $\vdash \psi_i \rhd t = u$, by CONSEQUENCE $\vdash \neg\psi_{k+1} \wedge \psi_i \rhd t = u$. Therefore by the induction hypothesis,

$$\vdash \bigvee \Psi' \rhd t = u$$

From this and the assumption $\vdash \psi_{k+1} \rhd t = u$, by PARTITION we obtain

$$\vdash \psi_{k+1} \vee \bigvee \Psi' \rhd t = u$$

Since $\psi_{k+1} \vee \bigvee \Psi' = \psi_{k+1} \vee (\neg\psi_{k+1} \wedge \bigvee_{1 \leq i \leq k} \psi_i) = \bigvee_{1 \leq i \leq k+1} \psi_i = \bigvee \Psi = \phi$, this completes the induction. □

In the rest of this section we discuss the soundness of the proof system. First we show that the rule UFI is sound with respect to $\sim$. Following [Mil89] we use the technique of *bisimulation up to*.

Definition 4.5 A symmetric relation R is a *timed bisimulation up to* $\sim$ if $(p,\ \ ) \in R$ implies

whenever $p \xrightarrow{d} p'$ then $\xrightarrow{d}\ \ '$ for some $\ \ '$ and $(p',\ \ ') \in R$.

whenever $p \xrightarrow{a} p'$ then $\xrightarrow{a}\ \ '$ for some $\ \ '$ and $(p',\ \ ') \in \sim R \sim$.

□

Note that the derivatives of delay transitions are required to be in the same relation, while those of action transitions are allowed to be related modular $\sim$.

Lemma 4.6 *If R is a timed bisimulation up to $\sim$ then $R \subseteq \sim$.*

Proof: Let $(p,\ ) \in R$ and $p \xrightarrow{\mu} p'$. We need to show that there is some $'$ such that $\xrightarrow{\mu}\ '$ and $(p',\ ') \in R$. The case when μ is an action is settled in the same way as in the proof of Proposition 6, Section 4.3, [Mil89]. The case when μ is a delay follows directly from Definition 4.5. □

Lemma 4.7 *If X is guarded in v and $v[t/X] \xrightarrow{a} t'$, then t' has the form $v'[t/X]$, and moreover, for any u, $v[u/X] \xrightarrow{a} v'[u/X]$.*

This lemma concerns only action transitions and its proof is the same as that of Lemma 13, Section 4.5, [Mil89].

Proposition 4.8 *Suppose $fv(v) \subseteq \{X\}$ and X is guarded in v. If $t\rho \sim v[t/X]\rho$ and $u\rho \sim v[u/X]\rho$ then $t\rho \sim u\rho$.*

Proof: We show the relation

$$R = \{\ (v[t/X]\rho, v[u/X]\rho) \mid fv(v)$$

Definition 4.10 Two processes p and $\quad$ are timed bisimular up to $d_0 \in \mathbf{R}^{\geq 0}$, written $p \sim^{d_0} \quad$, if for any d such that $0 \leq d \leq d_0$

whenever $p \xrightarrow{d} p'$ then $\quad \xrightarrow{d} \quad '$ for some $\quad '$ and $p' \dot{\sim} \quad '$,

whenever $\quad \xrightarrow{d} \quad '$ then $p \xrightarrow{d} p'$ for some p' and $p' \dot{\sim} \quad '$.

where $p \dot{\sim} \quad$ is defined thus

whenever $p \xrightarrow{a} p'$ then $\quad \xrightarrow{a} \quad '$ for some $\quad '$ and $p' \sim \quad '$,

whenever $\quad \xrightarrow{a} \quad '$ then $p \xrightarrow{a} p'$ for some p' and $p' \sim \quad '$.

□

The difference between timed bisimulation up to d and the standard notion of timed bisimulation only concerns initial delay transitions: in timed bisimulation up to d two processes are required to match only those initial delay transitions $intimedbisimupatind$

A term t provably ϕ-satisfies an equation set E if there exist a vector of terms $\{ t_i \mid i \in I \}$, each t_i being of the form $\{\psi_i'\}t_i'$, and a vector of conditions $\{ \phi_i \mid i \in I \}$ such that $\phi_1 = \phi$, $\vdash \phi \rhd t_1 = t$, $\phi_i \models Inv(u_i) \Leftrightarrow \psi_i'$, and

$$\vdash \phi_i \rhd t_i = u_i[\{\psi_i'\}(\phi_i \to t_i')/X_i | i \in I]$$

for each $i \in I$. We will simply say "t provably satisfies E" when $\phi_i = \mathtt{tt}$ for all $i \in I$.

Proposition 5.1 *For any guarded term t with free process variables $\mathbf{W}$ there exists a standard equation set E, with free process variables in $\mathbf{W}$, which is provably satisfied by t. In particular, if t is closed then E is also closed.*

Proof: By induction on the structure of t. The only non-trivial case is recursion when $t \equiv \mathbf{fix}Xt'$ with X guarded in t'. By induction there is a standard equation set $E' : \{ X_i = u_i \mid i \in I \}$ with free process variables in $FV(t) \cup \{X\}$ and $t_i' : s$ such that $\vdash t' = t_1'$ and

$$\vdash t_i' = u_i[t_i'/X_i | i \in I]$$

We may assume that X is different from any X_i. Let $v_i = u_i[u_1/X]$ for each i. Note that since X is under an action prefixing in t', it does not occur free in u_1. Hence $v_1 = u_1$. Consider the equation set

$$E : \quad \{ X_i = v_i \mid i \in I \}$$

To show t satisfies E, set $t_i = t_i'[t/X]$. Then

$$\begin{aligned} \vdash t &= \mathbf{fix}Xt' \\ &= \mathbf{fix}Xt_1' \\ &\overset{REC}{=} t_1'[\mathbf{fix}Xt_1'/X] \\ &= t_1'[t/X] \\ &= t_1 \end{aligned}$$

Now

$$\begin{aligned} \vdash t &= t_1'[t/X] \\ &= u_1[t_i'/X_i | i \in I][t/X] \\ &= u_1[t_i'[t/X]/X_i | i \in I] \\ &= u_1[t_i/X_i | i \in I] \end{aligned}$$

and

$$\begin{aligned} \vdash t_i &= t_i'[t/X] \\ &= u_i[t_i'/X_i | i \in I][t/X] \\ &= u_i[t, t_i'[t/X]/X, X_i | i \in I] \\ &= u_i[t, t_i/X, X_i | i \in I] \\ &= u_i[u_1[t_i/X_i | i \in I], t_i/X, X_i | i \in I] \\ &= u_i[u_1/X][t_i/X_i | i \in I] \\ &= v_i[t_i/X_i | i \in I] \end{aligned}$$

□

Proposition 5.2 *For closed terms t and u, if $t \sim^{\phi} u$ then there exist a ϕ' such that $\phi \Rightarrow \phi'$ and a standard, closed equation set E which is provably ϕ'-satisfied by both t and u.*

Proof: It easy to see that, using rule UNG, any unguarded term can be transformed into a guarded one, so we may assume both t and u are guarded.

Let the sets of clock variables of $t,\ u$ be $\mathbf{x},\ \mathbf{y}$, respectively, with $\mathbf{x} \cap \mathbf{y} = \emptyset$. Let also E_1 and E_2 be the standard equation sets for t and u, respectively:

$$E_1 : \quad \{\, X_i = \{\phi_i\} \sum_{k \in K_i} \phi_{ik} \rightarrow a_{ik}(\mathbf{x}_{ik}).X_{f(i,k)} \mid i \in I \,\}$$

$$E_2 : \quad \{\, Y_j = \{\psi_j\} \sum_{l \in L_j} \psi_{jl} \rightarrow b_{jl}(\mathbf{y}_{jl}).Y_{g(j,l)} \mid j \in J \,\}$$

So there are $t_i \equiv \{\phi_i'\}t_i',\ u_j \equiv \{\psi_j'\}u_j'$ with $\vdash t_1 = t, \vdash u_1 = u$ such that $\models \phi_i \Leftrightarrow \phi_i'$, $\models \psi_i \Leftrightarrow \psi_i'$, and

$$\vdash t_i = \{\phi_i\} \sum_{k \in K_i} \phi_{ik} \rightarrow a_{ik}(\mathbf{x}_{ik}).t_{f(i,k)} \qquad \vdash u_j = \{\psi_j\} \sum_{l \in L_j} \psi_{jl} \rightarrow b_{jl}(\mathbf{y}_{jl}).u_{g(j,l)}$$

Without loss of generality, we may assume $a_{ik} = b_{jl} = a$ for all i, k, j, l.

For each pair of $i,\ j$, let

$$\Phi_{ij} = \{\, \Delta \in \mathcal{RC}(\mathbf{xy}) \mid t_i \sim^{\Delta\Uparrow} u_j \,\}$$

Set $\phi_{ij} = \bigvee \Phi_{ij}$. By the definition of Φ_{ij}, ϕ_{ij} is the weakest condition over which t_i and u_j are symbolically bisimilar, that is, $\psi \Rightarrow \phi_{ij}$ for any ψ such that $t_i \sim\ u_j$. In particular, $\phi \Rightarrow \phi_{11}$. Also for each $\Delta \in \Phi_{ij}$, $\Delta \models Inv(t_i) \Leftrightarrow Inv(u_j)$, *i.e.*, $\Delta \models \phi_i' \Leftrightarrow \psi_j'$, hence(a)Tj/R2570.12Tf6.12031Td(j)Tj/32031Td()Tj/R1910.1910.12Tf12.48010Td(u)Tj/R1

Φβ allRCXf

By

Let w_{m+1} be $\mathbf{fix}X_{m+1}\phi_{m+1}{\to}v_{m+1}$. We have

$$\vdash \phi_{m+1} \rhd t_{m+1} = w_{m+1}[\phi_i{\to}t_i/X_i|1 \leq i \leq m]$$

By Proposition 4.1,

$$\vdash \phi_{m+1}{\to}t_{m+1} = \phi_{m+1}{\to}w_{m+1}[\phi_i{\to}t_i/X_i|1 \leq i \leq m]$$

Now, writing w_i for $v_i[\phi_{m+1}{\to}w_{m+1}/X_{m+1}]$, we have

$$\begin{aligned}
\vdash \phi_i \rhd t_i &= v_i[\phi_i{\to}t_i/X_i|1 \leq i \leq m+1] \\
&= v_i[\phi_i{\to}t_i/X_i|1 \leq i \leq m][\phi_{m+1}{\to}t_{m+1}/X_{m+1}] \\
&= v_i[\phi_i{\to}t_i/X_i|1 \leq i \leq m][\phi_{m+1}{\to}w_{m+1}[\phi_i{\to}t_i/X_i|1 \leq i \leq m]/X_{m+1}] \\
&= v_i[\phi_{m+1}{\to}w_{m+1}/X_{m+1}][\phi_i{\to}t_i/X_i|1 \leq i \leq m] \\
&= w_i[\phi_i{\to}t_i/X_i|1 \leq i \leq m]
\end{aligned}$$

This shows t provably ϕ-satisfies the equation set

$$E' :$$

in the timed world. This result agrees with the previous works on proof systems for value-passing processes [HL96] and for -calculus [Lin94], providing a further evidence that the four monoid laws capture the essence of bisimulation.

The

[DAB96] P.R. D'Argenio and Ed Brinksma. A Calculus for Timed Automata (Extended Abstract). In *FTRTFTS'96*, LNCS 1135, pp.110-129. Springer–Verlag. 1996.

[HL95] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.

[HL96] M. Hennessy and H. Lin. Proof systems for message-passing process algebras. *Formal Aspects of Computing*, 8:408–427, 1996.

[Lin94] H. Lin. Symbolic bisimulations and proof systems for the -calculus. Report 7/94, Computer Science, University of Sussex, 1994.

[LW00] H. Lin and Y. Wang. A proof system for timed automata. Fossacs'2000, LNCS 1784. March 2000.

[Mil84] R. Milner. A complete inference system for a class of regular behaviours. *J. Computer and System Science*, 28:439–466, 1984.

[Mil89] R. Milner. *Communication and ationIs199T212j54.Td,Tj7.xaall4.4Td7niv5619.dp-24*