Axiomatizing Timed Automata with Deadlines*

Pedro R. D'Argenio[1]** and Biniam Gebremichael[2]

[1] CONICET – FaMAF, Universidad Nacional de Córdoba.
Ciudad Universitaria, 5000 Córdoba, Argentina.
dargenio AT famaf.unc.edu.ar
[2] Institute for Computing and Information Sciences. Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
B.Gebremichael AT cs.ru.nl

Abstract. It is known that the usual timed bisimulation fails to be a congruence for timed automata with deadlines – a variant of timed automata where component synchronization is delayable, and time progress is controlled by deadlines on transitions instead of invariants on locations. Recently, we found the coarsest congruence relation that is included in timed bisimulation for timed automata with deadlines. In the present paper we provide an algebraic proof system for direct derivation of such relation by syntactic manipulation. We establish that the proof system is sound and complete.

1 Introduction

Due to increasingly growing involvement of computer systems (chips) in decision making and manipulation of real-time machinery, there has been a parallel growing interest to provide a mathematical foundation for the design and analysis of real-time systems. For example automata based [2, 3, 15, 21, etc.], and process algebra based [11, 9, 17, 4, etc.] are few of such theories. Notably, the theory of timed automata by Alur and Dill [3] has gained a remarkable popularity, and it has been applied as a fundamental modeling scheme for real-time analysis tools such as UPPAAL [5] and KRONOS [8].

The ability to build and analyze a system from its components [16, 14] is a key criterion for successful application of the method to large systems. Unfortunately this is quite a challenging problem and it has not been incorporated enough with the above theories. For example in timed automata, even though systems can be modeled as a network of timed automata, this is limited to modeling purpose only. There is no means to infer properties of the system from its components.

A variant of timed automata called *Timed Automata with Deadlines* (TADs for short) is proposed by Bornot and Sifakis [6, 13] to address some of the issues of compositionality in timed automata at analysis level. In particular, it addresses the problem of *timelock* (a state from which there is no path to a time passing transition). Timelock [7] in timed automata is a serious issue because: (1) it is a generic problem, that

* Supported by the EC Project IST-2001-35304 AMETIST, http://ametist.cs.utwente.nl.

** Part time researcher at Formal Methods and Tools Group, Dep. of Comp. Sci. University of Twente, NL. Supported by the NWO Vernieuwingsimpuls project "Verification of performance and dependability" and the ANPCyT project PICT 11-11738.

is, if an independent component is composed with a component that is timelocked, then the composed system inherit the timelock; and (2) the verification of many properties explicitly depend on the absence of timelock. The work in [6, 22, 13] shows how timelock-freedom of a system can be inferred from its components, in such a way that, if all components satisfy some *timelock-freedom condition* then it is guaranteed that the system is timelock-free by construction.

Recently [10] provided additional results to the theory of timed automata with deadlines. It presented the notion of congruence between components. That is, if two component are equivalent then the parent system remains the same no matter which of the two components is coupled with the system. In particular the timelock-freedom and equivalence of components are preserved under composition. The paper also proves that this relation, called ∇-bisimulation, is the coarsest (timed) bisimulation for TADs which is also a congruence.

In the present paper we further develop the results obtained in [10] to an algebraic theory that allows direct derivation of semantic equivalence of automata by purely syntactic manipulation. We present an axiomatization of such proof system for timed automata with deadlines, and we show that the proof system is sound and complete.

Related Work: Axiomatizations of timed automata have already appeared in [9] and [17]. The former one presents a sound axiomatization for safe timed automata [15]. The latter one presents a sound and complete proof system for bisimulation in the same class of automata. Our work is closely related to this one, but focused on a different model and a different type of bisimulation. Apart from the different setting, the following new results (w.r.t. [17]) are given: (1) our algebra has only one sort (in [17], the algebra contains two sorts —one with invariants and the other without— which simplifies the proof system); (2) we present a completeness result also for unguarded recursion; and (3) additionally, we correct a small technical mistake in the proof of Proposition 5.2 in [17] (see Lemma 9).

Organization of the paper: Section 2 defines the language to describe TADs. Its semantics and bisimulation relation is defined at the end of this section. Section 3 and 4 contains the axioms, inference rules and interesting properties of the language. Section 5 proves soundness of the proof system and Sections 6 and 6.3 prove completeness for guarded and all terms, respectively.

2 Algebra for Timed Automata with Deadlines

Clocks and Clock Constraints. A *clock* is a non-negative real-valued variable, which can be reset to zero at the occurrence of an event, and between two resets, its derivative with respect to time is equal to 1. Let $C = \{x_1, \ldots, x_N\}$ be a finite set of clocks. Let $\mathcal{F}(C)$ be the set consisting of propositional formulas containing atomic constraints in the form of $x_i \bowtie n$ or $x_i - x_j \bowtie m$, where x_i and x_j are clocks in C, $\bowtie \in \{<, >, \leq, \geq, =\}$ and n, m are natural numbers. Let **tt** and **ff** denote, respectively, the atomic constraints which are constantly true and false. Furthermore we let $\mathbf{x}, \mathbf{y} \subseteq C$ to denote sets of clocks.

A *clock valuation* is a function $\rho : C \to \mathbb{R}_{\geq 0}$ mapping each clock to the time elapsed since the last time it was reset to 0. Given a clock valuation ρ and $d \in \mathbb{R}_{\geq 0}$ the function

$\rho{+}d$ denotes the valuation such that for each clock $x \in C$, $(\rho{+}d)(x) = \rho(x) + d$. The function $\rho\{\mathbf{x}{:=}0\}$ denotes the valuation such that for each clock $x \in \mathbf{x}$, $\rho\{\mathbf{x}{:=}0\}(x) = 0$, otherwise $\rho\{\mathbf{x}{:=}0\}(x) = \rho(x)$. A constraint ϕ is called *left closed* iff for all valuations ρ, $\rho \models \neg\phi \;\Rightarrow\; \exists\varepsilon > 0 : \forall\varepsilon' \leq \varepsilon : \rho + \varepsilon' \models \neg\phi$. For a constraint ϕ and a set of clocks $\mathbf{x}$, the *reset* $\phi{\downarrow_{\mathbf{x}}}$ is a constraint such that for all ρ, $\rho \models \phi{\downarrow_{\mathbf{x}}}$ iff $\rho = \rho'\{\mathbf{x} := 0\}$ and $\rho' \models \phi$ for some ρ'. For a constraint ϕ, the *time successor* $\phi \Uparrow$ is a constraint such that $\forall\rho$, $\rho \models \phi \Uparrow$ iff $\rho = \rho' + d$ and $\rho' \models \phi$ for some ρ' and $d \geq 0$. See [**?**,17] for syntactical definition of reset and time successor. A clock constraint ϕ is $\Uparrow$-*closed* iff $\phi = \phi \Uparrow$. Given a constraint ϕ, a ϕ-*partition* [17] is a finite set of constraints Φ such that $\bigvee\Phi \Leftrightarrow \phi$ and for any two distinct $\psi, \psi' \in \Phi$, ψ and ψ' are disjoint (i.e. $\psi \wedge \psi'$ is a contradiction). The set $\mathcal{RC}(\phi)$ denotes the set of all regions [3] that constitute ϕ.

2.1 The Language

Let $\mathcal{A}$ be a finite set of actions, ranged over by a, b. Let $\mathbf{X}$ be a set of process variables ranged over by X, Y, and let $\gamma, \delta \in \mathcal{F}(C)$ be clock constraints. The *Algebra for Timed Automata with Deadlines* $\mathbb{A}$ over $\mathcal{A}, C$ and $\mathbf{X}$ is given by the following BNF grammar:

$$t ::= \mathbf{0} \mid \gamma \rightarrow t \mid \delta : t \mid t + t \mid \mathbf{fix}X\ t \mid a(\mathbf{x}).t \mid X \tag{1}$$

The expression $a(\mathbf{x}).t$ with $a \in \mathcal{A}$ is the action prefixing operator with clock resetting. The clock constraints γ and δ are called *guard* and *deadline* constraints, respectively. The term $\gamma \rightarrow t$ represents a conditional construction such that when the guard γ is true, it *may* perform any action t is able to perform. The term $\delta : t$ represents a deadline construction such that when the deadline δ is true, the process *must* perform some action that t can perform. We assume δ is left closed. The term $\mathbf{0}$ denotes an inactive process which can do nothing except allowing time to pass. The process $\mathbf{tt} : t$ behaves the same as t except it forces the execution of any enabled action before letting time pass. We call $\mathbf{tt} : t$ an *urgent* process. As usual, $t_1{+}t_2$ and $\mathbf{fix}Xt$ are, respectively, the non-deterministic choice and the recursion operation.

To reduce the number of parenthesis we, adopt the following binding power in decreasing order on the operators: action prefix, $\mathbf{fix}X$, deadline, guard and summation.

Example 1. Consider the following simple `ssh` server login procedure. Initially the server is idle, until a client program requests a connection via action a. The server accepts the request and it waits for 2 minutes for the user to enter his/her user name and password. If this is achieved the server passes control to a login verifier via action b. If the user name and password matches (action e) the user enters the system. Otherwise, the server loops back (action d) and asks the user to enter his/her user name and password again. After waiting for 2 minutes if no user name and password is entered, the connection is broken (action c) and the server is back to its idle state. This can be modeled in $\mathbb{A}$, using one clock variable x, by the process `ssh` below

$$\begin{aligned}
&\texttt{ssh} \equiv s_0\\
&\quad s_0 \equiv \mathbf{fix}X_0(a(\{x\}).s_1)\\
&\quad s_1 \equiv \mathbf{fix}X_1((x{=}2) : (x{\leq}2 \rightarrow b(\varnothing).s_2 + x{\geq}2 \rightarrow c(\varnothing).X_0))\\
&\quad s_2 \equiv d(\{x\}).X_1 + e(\varnothing).\mathbf{0}
\end{aligned}$$

$$dl(t_0 + t_1, A) = dl(t_0, A) \vee dl(t_1, A) \qquad I(t_0 + t_1) = I(t_0) \cup I(t_1)$$
$$dl(\mathbf{fix}\, \mathbf{x} X t, A) = dl(t[\mathbf{fix}\, \mathbf{x} X t/X], A) \qquad I(\mathbf{fix}\, \mathbf{x} X t) = I(t[\mathbf{fix}\, \mathbf{x} X t/X])$$
$$dl(\delta : t, A) = \begin{cases} (\delta \wedge gd(t, A)) \vee dl(t, A) & \text{if } A \cap I(t) \neq \varnothing \\ \mathbf{ff} & \text{otherwise} \end{cases} \qquad I(\delta : t) = I(t)$$
$$dl(\gamma \to t, A) = \gamma \wedge dl(t, A) \qquad I(\gamma \to t) = I(t)$$
$$dl(a(\mathbf{x}).t, A) = dl(\mathbf{0}, A) = dl(X) = \mathbf{ff} \qquad I(a(\mathbf{x}).t) = \{a\}$$
$$I(\mathbf{0}) = I(X) = \varnothing$$

$$gd(\mathbf{0}, A) = \mathbf{ff} \quad gd(a(\mathbf{x}).t, A) = gd(X, A) = \mathbf{tt} \qquad gd(t + u, A) = gd(t, A) \vee gd(u, A)$$
$$gd(\gamma \to t, A) = \begin{cases} \gamma \wedge gd(t, A) & \text{if } A \cap I(t) \neq \varnothing \\ \mathbf{ff} & \text{otherwise} \end{cases} \qquad gd(\mathbf{fix}\, \mathbf{x} X t, A) = gd(t[\mathbf{fix}\, \mathbf{x} X t/X], A)$$
$$gd(\delta : t, A) = gd(t, A)$$

Fig. 1. Definitions of deadline (dl) and set of initial actions (I)

In the following, we say that a variable X occurs *unguarded* in a term t if such an occurrence is not within the scope of an action prefix. If X does not occur unguarded in t we say that X is *guarded* in t. Hence, X occurs unguarded in $(x \geq 5) : ((x \geq 2) \to X + a(\{y\}).Y)$, but Y is guarded. (Note that that the concept of guarded variable is *not* related the guard operation.) We say that a term t is *guarded* if all of its subterms of the form $\mathbf{fix} X u$ and X is guarded in u.

2.2 Transitional Semantics

The semantics of $\mathbb{A}$ is formally defined in terms of a timed transitions system $TS_\nabla = (\mathcal{S}, \Sigma, \longrightarrow)$ where

- $\mathcal{S} \subseteq (\mathbb{A} \times 2^{\mathcal{A}}) \times (C \to \mathbb{R}_{\geq 0})$ is set of states
- $\Sigma = \mathcal{A} \cup \mathbb{R}_{\geq 0} \cup \mathcal{A}_\nabla \cup \{\Delta\}$ is set of vocabulary, where $\mathcal{A}_\nabla = \{\nabla_A \mid A \subseteq \mathcal{A}\}$ and Δ are the drop and undrop actions as described in the previous chapter.
- $\longrightarrow$ (the transition relation) is defined as in Fig. 2.

Before explaining the transition relation ($\longrightarrow$) in detail, first we need to formalize the notion of *deadline of terms*, which ensures the maximum delay time of a term before a discrete action is forced to take place. Let $I(t)$ be the set of all actions a s.t. a sub term $a(\mathbf{x}).u$ in t occurs out of the scope of another action prefix. $I(t)$ is formally defined as the smallest set satisfying equations in Fig. 1. Let $gd(t, A)$ be the enabling condition in t of all actions *not in* A, that is, t can preform *some action* $a \notin A$ in valuation ρ iff $gd(t, A)$ is satisfied in ρ. It is defined as the weakest predicate satisfying equations in Fig. 1. The deadline of a term t considering only deadlines on actions in $A \subseteq \mathcal{A}$ (i.e. disregarding $\mathcal{A}$–A) is the disjunction of all deadlines imposed on any enabled action $a \in A \cap I(t)$ and originating from t. Formally, $dl(t, A)$ is defined as the weakest predicate satisfying equations in Fig. 1. Constraint $gd(t, A)$ is needed in the definition of dl to guarantee that $dl(t, A) \Rightarrow gd(t, A)$. Precisely, this implication is what ensures *timelock freedom*, i.e. it ensures that when time stops progressing, a transition must be enable in order to guarantee progress of the system model.

The semantics of $\mathbb{A}$ is given in Fig. 2 in a structural way. Most rules are fairly obvious (e.g. GUARD allows the execution of an action only if guard γ is valid in

$$\text{DELAY}\quad \frac{\forall d' < d \quad \rho + d' \models \neg dl(t, \mathcal{A} - D)}{(t, D)\rho \xrightarrow{d} (t, D)\rho + d} \qquad \text{REC}\quad \frac{(t[\mathbf{fi}\ \mathbf{x}Xt/X], D)\rho \xrightarrow{a} (t', D')\rho'}{(\mathbf{fi}\ \mathbf{x}Xt, D)\rho \xrightarrow{a} (t', D')\rho'}$$

$$\text{ACTION}\quad \frac{}{(a(\mathbf{x}).t, D)\rho \xrightarrow{a} (t, \varnothing)\rho\{\mathbf{x}{:=}0\}} \qquad \text{DROP}\quad \frac{}{(t, D)\rho \xrightarrow{\nabla_A} (t, D \cup A)\rho}$$

$$\text{GUARD}\quad \frac{(t, D)\rho \xrightarrow{a} (t', D')\rho' \quad \rho \models \gamma}{(\gamma \to t, D)\rho \xrightarrow{a} (t', D')\rho'} \qquad \text{UNDROP}\quad \frac{}{(t, D)\rho \xrightarrow{\Delta} (t, \varnothing)\rho}$$

$$\text{DEADLINE}\quad \frac{(t, D)\rho \xrightarrow{a} (t', D')\rho'}{(\delta{:}t, D)\rho \xrightarrow{a} (t', D')\rho'} \qquad \text{CHOICE}\quad \frac{(t, D\rho) \xrightarrow{a} (t', D')\rho'}{\begin{array}{c}(t + u, D)\rho \xrightarrow{a} (t', D')\rho' \\ (u + t, D)\rho \xrightarrow{a} (t', D')\rho'\end{array}}$$

Fig. 2. Transitional Semantics of $\mathbb{A}$

the current valuation, or DEADLINE states that deadlines have no effect on discrete actions). In particular notice the resetting of clocks in rule ACTION. DELAY defines the time progress: a state $(t, D)\rho$ can progress d time units if no deadline under consideration is reached within this period (i.e. deadline $dl(t, \mathcal{A}{-}D)$ is false in any valuation between ρ and $\rho{+}d$.) Rules DROP and UNDROP define the effect of the drop and undrop actions respectively. Note that they can be performed unconditionally, changing at any arbitrary moment the deadlines that have to be considered for the progress of time.

The notion of equivalence underlying the algebra $\mathbb{A}$ is defined in the following. We say that two states $p = (t, D)\rho$ and $q = (u, E)\eta$ are ∇-*bisimilar*, notation $p \sim^{\nabla} q$ iff there exists a symmetric relation R (called ∇-*bisimulation*) such that for any $(p, q) \in R$ and $l \in \mathbb{R}_{\geq 0} \cup \mathcal{A} \cup \mathcal{A}_{\nabla} \cup \{\Delta\}$, whenever $p \xrightarrow{l} p'$ then $\exists q' : q \xrightarrow{l} q'$ and $p' R q'$. If $p' R q'$ is changed to $p' \sim^{\nabla} \circ R \circ \sim^{\nabla} q'$, R is a ∇-*bisimulation up to* $\sim^{\nabla}$ ($\circ$ is the usual composition on relations). It is enough to prove the existence of a ∇-bisimulation up to $\sim^{\nabla}$ between p and q to state that they are ∇-bisimilar [19].

2.3 Symbolic Semantics.

The language $\mathbb{A}$ is designed in such a way that there is a direct translation between $\mathbb{A}$ terms and TADs. The symbolic semantics of an $\mathbb{A}$ term t in terms of TAD is defined by $T_t = (\mathbb{A}, t, C, \rightarrowtriangle)$, where $\rightarrowtriangle$ is the smallest relation satisfying the rules in Fig. 3 and $\mathcal{A}$ is the set of actions for TADs.

Example 2. The $\mathbb{A}$ model of the `ssh` server in Example 1 can be interpreted in terms of a TAD using the rules in Fig. 3. The resulting TAD is given in Fig. 4. The derivation is straightforward, the interesting part is how to "push" the deadline (x=2) to both branches of s_1. This is done by applying ACTION, GUARD, CHOICE, and DEADLINE sequentially to both branches of s_1.

Conversely, for a given a TAD $T = (\mathcal{L}, l_0, C, \rightarrowtriangle)$, its equivalent $\mathbb{A}$ term can be derived as follows. Suppose $\mathcal{L} = \{l_0, l_1, \ldots, l_n\}$ with $l_i \leq l_j$ iff $i \leq j$. For each $l \in \mathcal{L}$, let

$$\text{ACTION}\ \frac{}{a(\mathbf{x}).t \xrightarrow{a,\mathbf{tt},\mathbf{ff},\mathbf{x}} t} \qquad \text{CHOICE}\ \frac{t \xrightarrow{a,\gamma,\delta,\mathbf{x}} t'}{(t+u) \xrightarrow{a,\gamma,\delta,\mathbf{x}} t' \quad (u+t) \xrightarrow{a,\gamma,\delta,\mathbf{x}} t'}$$

$$\text{REC}\ \frac{t[\mathbf{fix}\, X t/X] \xrightarrow{a,\gamma,\delta,\mathbf{x}} t'}{\mathbf{fix}\, X t \xrightarrow{a,\gamma,\delta,\mathbf{x}} t'} \qquad \text{DEADLINE}\ \frac{t \xrightarrow{a,\gamma',\delta',\mathbf{x}} t'}{\delta : t \xrightarrow{a,\gamma',\delta' \vee (\delta \wedge \gamma'),\mathbf{x}} t'}$$

$$\text{GUARD}\ \frac{t \xrightarrow{a,\gamma',\delta',\mathbf{x}} t'}{\gamma \to t \xrightarrow{a,\gamma \wedge \gamma',\delta' \wedge \gamma,\mathbf{x}} t'}$$

Fig. 3. Symbolic semantics of $\mathbb{A}$

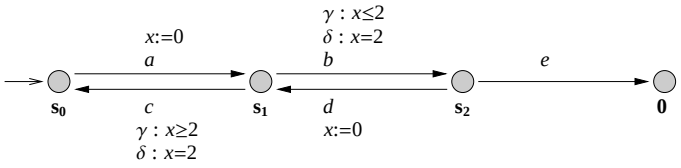

Fig. 4. TAD for `ssh` login procedure

$J_l = \{e \mid e = (l, a_e, \gamma_e, \delta_e, \mathbf{x}_e, l_e) \in \rightarrowtail\}$ and define

$$t_l \stackrel{\text{def}}{=} \mathbf{fix} X_l \Big(\textstyle\sum_{e \in J_l} (\gamma_e \to \delta_e : a_e(\mathbf{x}_e).u_{l_e}) \Big)$$

where

$$u_{l_e} = \begin{cases} X_{l_e} & \text{if } l_e \leq l \\ t_{l_e} & \text{otherwise} \end{cases}$$

Example 3. t_{s_0} is the $\mathbb{A}$ term associated to the TAD of Fig. 4, where:

$$t_{s_0} \equiv \mathbf{fix} X_{s_0}(\mathbf{ff} : \mathbf{tt} \to a(\{x\}).t_{s_1})$$
$$t_{s_1} \equiv \mathbf{fix} X_{s_1}((x{=}2) : (x{\leq}2) \to b(\varnothing).t_{s_2} + (x{=}2) : (x{\geq}2) \to c(\varnothing).X_{s_0}))$$
$$t_{s_2} \equiv \mathbf{fix} X_{s_2}(\mathbf{ff} : \mathbf{tt} \to d(\{x\}).X_{s_1} + \mathbf{ff} : \mathbf{tt} \to e(\varnothing).t_{s_3})$$
$$t_{s_3} \equiv \mathbf{fix} X_{s_3} \mathbf{0}$$

The semantics of T in terms of transition systems is given by the transitional semantics of t_{l_0}. It is routine to show that the semantics of T given in this manner is the same as the one defined in the previous chapter. Moreover, by induction on the proof tree, it is also possible to show that the transitional semantics of t is the same as the two step semantics of t (i.e. interpret t as a TAD T_t and then obtain the transition system from T_t) provided that t is closed (i.e., it does not contain a variable X out of the scope of a $\mathbf{fix}X$.)

3 The Proof System

The proof system of $\mathbb{A}$ is given by the set of axioms and inference rules in Fig. 5 and Fig. 6 respectively. The judgments of the inference system are conditional equations of

S1	$t + \mathbf{0} = t$	U1	$\mathbf{tt} : \mathbf{tt} : t = \mathbf{tt} : t$
S2	$t + t = t$	U2	$\mathbf{tt} : \delta : t = \delta : \mathbf{tt} : t$
S3	$(t + u) + v = t + (u + v)$	U3	$\mathbf{tt} : \gamma \to t = \gamma \to \mathbf{tt} : t$
S4	$t + u = u + t$	U4	$\mathbf{tt} : (t_1 + t_2) = \mathbf{tt} : t_1 + \mathbf{tt} : t_2$
		U5	$\mathbf{tt} : t = \mathbf{tt} : t + t$
DL	$(\delta \wedge \gamma_1) : (\gamma_1 \to a(\mathbf{x}).t + \gamma_2 \to a(\mathbf{y}).u) = \delta : \gamma_1 \to a(\mathbf{x}).t + \gamma_2 \to a(\mathbf{y}).u$		
UR	$\mathbf{fi}\,\mathbf{x}X(t + \delta : \gamma \to X) = \mathbf{fi}\,\mathbf{x}X(t + \delta : \gamma \to t)$		

Fig. 5. The equational axioms

the form

$$\phi \vdash t = u$$

where ϕ is a constraint and t, u are terms. Its intended meaning is: t is equivalent to u whenever ϕ holds. We will abbreviate $\mathbf{tt} \vdash t = u$ as $t = u$ and consider in general two logically equivalent constraints as the same constraint (hence, e.g., $\mathbf{tt} \vdash t = u$ and $(x{+}1 \geq x) \vdash t = u$ are the same judgment.)

Axioms S1-4 are standard summation laws and U1-5 are axioms to manipulate urgent processes. Axiom UR explains in which way unguarded variables in recursion are redundant (notice the difference with Milner's [19] recursion axiom $\mathbf{fix}X(t + X) = \mathbf{fix}Xt$). Axiom DL shows a particularity of the ∇-bisimulation: a deadline on an action has the same impact on another process as long as it is prefixed with the *same action*. Deadlines *cannot* be shifted out of any arbitrary summation. As a simple example, the term $\delta : a.(\mathbf{x}).t + b.(\mathbf{y}).u$ and $\delta : (a.(\mathbf{x}).t + b.(\mathbf{y}).u)$ will only be equivalent if and only if $a = b$. This is precisely what usual bisimulation would allow hence failing to be a congruence.

Each construct in the language has an entry in the set of inference rule of Fig. 6. They show how to use the constructs, and what constraints must be met (if any) before applying the rule. Three additional rules, namely, ABSURD, PARTITION and CONSEQUENCE are also given. They are used to manipulate the condition under which the equation hold. SUBSTITUTION rules handle substitution in the context of choice operator and urgency. Rule ACTION is the specific rule of substitution for action prefix. GUARD does a case analysis on conditions: if

1. t behaves like u when the guard γ holds under ϕ (i.e., when $\phi \wedge \gamma$ holds), and
2. t behaves like $\mathbf{0}$ if this is not the case,

then $\phi \vdash \gamma \to t = u$ can be inferred. The rule DEADLINE is similar to GUARD except that t is required to be urgent when $\phi \wedge \delta$ holds. THINNING states that clocks which are not free in t (denoted by $C(t)$) are redundant in a reset set of a prefix of t. There are two rules for recursion: REC is for folding or unfolding recursion expressions, while UFI states uniqueness of solution of recursive equations provided that the variable of interest occurs only guarded.

EQUIV	$\dfrac{}{t = t} \quad \dfrac{\phi \vdash t = u}{\phi \vdash u = t} \quad \dfrac{\phi \vdash t = u \quad \phi \vdash u = v}{\phi \vdash t = v}$
AXIOM	$\dfrac{}{t = u}$ $t = u$ an axiom instance
SUBSTITUTION	$\dfrac{\phi \vdash t = t'}{\phi \vdash t + u = t' + u} \quad \dfrac{\phi \vdash t = u}{\phi \vdash \mathbf{tt} : t = \mathbf{tt} : u}$
ACTION	$\dfrac{\phi{\downarrow_{\mathbf{x}}}\!\Uparrow \ \vdash t = u}{\phi \vdash a(\mathbf{x}).t = a(\mathbf{x}).u}$
GUARD	$\dfrac{\phi \wedge \gamma \vdash t = u \quad \phi \wedge \neg\gamma \vdash \mathbf{0} = u}{\phi \vdash \gamma \to t = u}$
DEADLINE	$\dfrac{\phi \wedge \neg\delta \vdash t = u \quad \phi \wedge \delta \vdash \mathbf{tt}{:}t = u}{\phi \vdash \delta{:}t = u}$
THINNING	$\dfrac{}{\phi \vdash a(\mathbf{xy}).t = a(\mathbf{x}).t}$ $\mathbf{y} \cap C(t) = \varnothing$
REC	$\dfrac{}{\mathbf{fi}\,\mathbf{x}Xt = t[\mathbf{fi}\,\mathbf{x}Xt/X]}$
UFI	$\dfrac{t = u[t/X]}{t = \mathbf{fi}\,\mathbf{x}Xu}$
PARTITION	$\dfrac{\phi_1 \vdash t = u \quad \phi_2 \vdash t = u}{\phi_1 \vee \phi_2 \vdash t = u}$
CONSEQUENCE	$\dfrac{\psi \vdash t = u}{\phi \vdash t = u}$ $\phi \Rightarrow \psi$
ABSURD	$\dfrac{}{\mathbf{ff} \vdash t = u}$

Fig. 6. The inference rules

4 Properties of the Proof System

This section is presents some selected properties of the proof system, which are used to prove the soundness and completeness. This section is technical and readers only interested in the soundness and completeness of the proof system can safely skip this section when reading for the first time. The proof of some of them is given in appendix A.

Lemma 1. *1. If $\phi \Rightarrow \gamma$ and $\phi \vdash t = u$ then $\phi \vdash \gamma \to t = u$.*
2. If $\phi \Rightarrow \neg\delta$ and $\phi \vdash t = u$ then $\phi \vdash \delta : t = u$.
3. If $\phi \Rightarrow \gamma \wedge \neg\delta$ and $\phi \vdash t = u$ then $\phi \vdash \gamma{\to}\delta{:}t = u$ and $\phi \vdash \delta{:}\gamma{\to}t = u$.
4. $t = t + \phi \to t$.
5. $\gamma_1 \to \gamma_2 \to t = (\gamma_1 \wedge \gamma_2) \to t$
6. $\gamma \to (t_1 + t_2) = \gamma \to t_1 + \gamma \to t_2$.
7. $\gamma_1 \to t + \gamma_2 \to t = (\gamma_1 \vee \gamma_2) \to t$.
8. $\delta_1 : \delta_2 : t = (\delta_1 \vee \delta_2) : t$.

9. $\delta : (t_1 + t_2) = \delta : t_1 + \delta : t_2$.
10. *If* $\phi \vdash t = u$ *then* $\phi \vdash \delta : t = \delta : u$ *for any* δ.
11. $\delta_1 : t + \delta_2 : t = (\delta_1 \vee \delta_2) : t$.
12. $\delta : \gamma \to t = (\delta \wedge \gamma) : \gamma \to t$.
13. $\delta : \gamma \to t = \gamma \to \delta : t$.
14. $\phi \vdash t = u \;\Rightarrow\; \phi \vdash \delta : t = \delta : u$.
15. $\mathbf{ff} \to t = \mathbf{0}$.
16. $\mathbf{tt} : \mathbf{0} = \mathbf{0}$.
17. *From [17]:* $\phi \vdash a(\mathbf{x}).t = a(\mathbf{x}).\phi\downarrow_{\mathbf{x}}\!\Uparrow \to t$.
18. *From [17]: Suppose* Ψ *is a* ϕ*-partition and* $\psi \vdash t = u$ *for each* $\psi \in \Psi$*, then* $\phi \vdash t = u$.

The proof of Lemma 1.10 – 1.15 are straight forward application of the above lemmas, and their proofs are omitted.

The following Lemma helps to gather summands that only differ in their guards and deadlines

Lemma 2. *Let* δ_1, δ_2, γ_1, *and* γ_2 *be predicates, then the equation*

$$\delta_1 : \gamma_1 \to t + \delta_2 : \gamma_2 \to t = \Big((\delta_1 \wedge \gamma_1) \vee (\delta_2 \wedge \gamma_2)\Big) : (\gamma_1 \vee \gamma_2) \to t$$

is provable. In particular if $\delta_1 \Rightarrow \gamma_1$ *and* $\delta_2 \Rightarrow \gamma_2$ *then*

$$\delta_1 : \gamma_1 \to t + \delta_2 : \gamma_2 \to t = (\delta_1 \vee \delta_2) : (\gamma_1 \vee \gamma_2) \to t$$

Proof. First we shall prove for the case when one of the deadlines (say δ_2) is false. That is we need to prove

$$\delta_1 : \gamma_1 \to t + \gamma_2 \to t = (\delta_1 \wedge \gamma_1) : (\gamma_1 \vee \gamma_2) \to t \tag{2}$$

by DEADLINE we need to prove that

$$\neg\delta_1 \vdash \delta_1 : \gamma_1 \to t + \gamma_2 \to t = (\gamma_1 \vee \gamma_2) \to t \tag{3}$$

$$\delta_1 \vdash \delta_1 : \gamma_1 \to t + \gamma_2 \to t = \mathbf{tt} : (\gamma_1 \vee \gamma_2) \to t \tag{4}$$

Using DEADLINE, ABSURD and EQUIV we can easly prove that

$$\neg\delta_1 \vdash \delta_1 : \gamma_1 \to t = \gamma_1 \to t$$

Next, we can add $\gamma_2 \to t$ on both sides using SUBSTITUTION. The right hand side equation will be equal to $(\gamma_1 \vee \gamma_2) \to t$ using Lemma 1.7, which proves equation (3).

In order to prove (4) we use GUARD, Lemma 1.12 and Lemma 1.13 to decompose the problem into the following equations.

$$(\delta_1 \wedge \gamma_1) \wedge (\gamma_1 \vee \gamma_2) \vdash \delta_1 : \gamma_1 \to t + \gamma_2 \to t = \mathbf{tt} : t \tag{5}$$

$$(\delta_1 \wedge \gamma_1) \wedge \neg(\gamma_1 \vee \gamma_2) \vdash \delta_1 : \gamma_1 \to t + \gamma_2 \to t = \mathbf{0} \tag{6}$$

$\Rightarrow$ {By DEADLINE, ABSURD and EQUIV}

$(\delta_1 \wedge \gamma_1) \vdash \delta_1 : t = \mathbf{tt} : t$

$\Rightarrow$ {Applying Lemma 1.1, Lemma 1.13 and EQUIV}

$$
\begin{aligned}
&(\delta_1 \wedge \gamma_1) \vdash \delta_1 \to \gamma_1 : t = \mathbf{tt} : t \\
\Rightarrow\ &\{\text{By SUBSTITUTION}\} \\
&(\delta_1 \wedge \gamma_1) \vdash \delta_1 \to \gamma_1 : t + \gamma_2 \to t = \mathbf{tt} : t + \gamma_2 \to t \\
\Rightarrow\ &\{\text{By Lemma 1.11 and EQUIV}\} \\
&(\delta_1 \wedge \gamma_1) \vdash \delta_1 \to \gamma_1 : t + \gamma_2 \to t = \mathbf{tt} : t \\
\Rightarrow\ &\{\text{By CONSEQUENCE since } (\delta_1 \wedge \gamma_1) = (\delta_1 \wedge \gamma_1) \wedge (\gamma_1 \wedge \gamma_2)\ \} \\
&(\delta_1 \wedge \gamma_1) \wedge (\gamma_1 \wedge \gamma_2) \vdash \delta_1 \to \gamma_1 : t + \gamma_2 \to t = \mathbf{tt} : t
\end{aligned}
$$

This proves equation (5). Note that $(\delta_1 \wedge \gamma_1) \wedge \neg(\gamma_1 \vee \gamma_2) = \mathbf{ff}$ and by ABSURD and CONSEQUENCE we prove (6), which completes the proof of (4).

The same proof applies for the case when δ_1 is false. Finally we group these two cases and prove the present Lemma as follows.

$$
\begin{aligned}
&(\delta_1 \wedge \gamma_1) : \gamma_1 \to t + (\delta_2 \wedge \gamma_2) : \gamma_2 \to t \\
=\ &\{\text{Applying (4) twice}\} \\
&(\delta_1 \wedge \gamma_1) : \gamma_1 \to t + \gamma_1 \to t + (\delta_2 \wedge \gamma_2) : \gamma_2 \to t + \gamma_2 \to t \\
=\ &\{\text{By S1-S4 and Lemma 1.6}\} \\
&(\delta_1 \wedge \gamma_1) : \gamma_1 \to t + (\gamma_1 \vee \gamma_2) \to t + (\delta_2 \wedge \gamma_2) : \gamma_2 \to t + (\gamma_1 \vee \gamma_2) \to t \\
=\ &\{\text{Applying (4) twice}\} \\
&(\delta_1 \wedge \gamma_1) : (\gamma_1 \vee \gamma_2) \to t + (\delta_2 \wedge \gamma_2) : (\gamma_1 \vee \gamma_2) \to t \\
=\ &\{\text{By Lemma Lemma 1.11}\} \\
&(\delta_1 \wedge \gamma_1) \vee (\delta_2 \wedge \gamma_2) : (\gamma_1 \vee \gamma_2) \to t
\end{aligned}
$$

Lemma 3 is a genaralization of the axiom DL which deadline of an action has the same impact on any number of summands as long as they are prefixed with the same action.

Lemma 3. *Let δ_i, γ_i be predicates forall finite number of i, then the following generalization equation of axiom* DL *is provable.*

$$
\sum_{i=1}^{n} \left(\delta_i : \gamma_i \to a(\mathbf{x}_i).t_i\right) = \bigvee_{i=1}^{n} (\delta_i \wedge \gamma_i) : \left(\sum_{i=1}^{n} \gamma_i \to a(\mathbf{x}_i).t_i\right)
$$

Proof. We will show for the case when $n = 3$. Using the same technique recursively, it is straigthforward to show for arbitrary n.

$$
\begin{aligned}
&\delta_1 : \gamma_1 \to a(\mathbf{x_1}).t_1 + \delta_2 : \gamma_2 \to a(\mathbf{x_2}).t_2 + \delta_3 : \gamma_3 \to a(\mathbf{x_3}).t_3 \\
=\ &\{\text{Applying S1-S4, Lemma 1.2, Lemma 1.11 and Lemma 1.12}\} \\
&(\delta_1 \wedge \gamma_1) : \gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_1 \to a(\mathbf{x_1}).t_1 + (\delta_1 \wedge \gamma_1) : \gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_1 \to a(\mathbf{x_1}).t_1 + \\
&(\delta_2 \wedge \gamma_2) : \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_2 \to a(\mathbf{x_2}).t_2 + (\delta_2 \wedge \gamma_2) : \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_2 \to a(\mathbf{x_2}).t_2 + \\
&(\delta_3 \wedge \gamma_3) : \gamma_3 \to a(\mathbf{x_2}).t_3 + \gamma_3 \to a(\mathbf{x_3}).t_3 + (\delta_3 \wedge \gamma_3) : \gamma_3 \to a(\mathbf{x_2}).t_3 + \gamma_3 \to a(\mathbf{x_2}).t_3 + \\
=\ &\{\text{Applying S1-S4}\} \\
&(\delta_1 \wedge \gamma_1) : \gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_2 \to a(\mathbf{x_2}).t_2 + (\delta_1 \wedge \gamma_1) : \gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_3 \to a(\mathbf{x_3}).t_3 + \\
&(\delta_2 \wedge \gamma_2) : \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_1 \to a(\mathbf{x_1}).t_1 + (\delta_2 \wedge \gamma_2) : \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_3 \to a(\mathbf{x_3}).t_3 +
\end{aligned}
$$

$(\delta_3 \wedge \gamma_3) : \gamma_3 \to a(\mathbf{x_3}).t_3 + \gamma_1 \to a(\mathbf{x_1}).t_1 + (\delta_3 \wedge \gamma_3) : \gamma_3 \to a(\mathbf{x_3}).t_3 + \gamma_2 \to a(\mathbf{x_2}).t_2$

$=$ {Applying DL six times}

$(\delta_1 \wedge \gamma_1) : (\gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_2 \to a(\mathbf{x_2}).t_2) + (\delta_1 \wedge \gamma_1) : (\gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_3 \to a(\mathbf{x_3}).t_3) +$
$(\delta_2 \wedge \gamma_2) : (\gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_1 \to a(\mathbf{x_1}).t_1) + (\delta_2 \wedge \gamma_2) : (\gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_3 \to a(\mathbf{x_3}).t_3) +$
$(\delta_3 \wedge \gamma_3) : (\gamma_3 \to a(\mathbf{x_3}).t_3 + \gamma_1 \to a(\mathbf{x_1}).t_1) + (\delta_3 \wedge \gamma_3) : (\gamma_3 \to a(\mathbf{x_3}).t_3 + \gamma_2 \to a(\mathbf{x_2}).t_2)$

$=$ {Applying Lemma 1.9 and S1-S4 three times }

$(\delta_1 \wedge \gamma_1) : (\gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_3 \to a(\mathbf{x_3}).t_3) +$
$(\delta_2 \wedge \gamma_2) : (\gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_3 \to a(\mathbf{x_3}).t_3) +$
$(\delta_3 \wedge \gamma_3) : (\gamma_1 \to a(\mathbf{x_1}).t_1 + \gamma_2 \to a(\mathbf{x_2}).t_2 + \gamma_3 \to a(\mathbf{x_3}).t_3) +$

$=$ {Applying Lemma 1.11 and S1-S4 twice }

$$\bigvee_{i=1}^{3}(\delta_i \wedge \gamma_i) : \Big(\sum_{i=1}^{3} \gamma_i \to a(\mathbf{x}_i).t_i\Big)$$

5 Soundness of the Proof System

In the previous sections, we provided axioms and inference rules to simplify and manipulate terms in $\mathbb{A}$. In this section we prove the soundness of these inference rules with respect to ∇-bisimulation. Formally, the soundness of the proof system can be stated as follows

Theorem 1. *If $\phi \vdash t{=}u$ and ϕ is $\Uparrow$-closed then $(t, D)\rho \sim^{\nabla} (u, D)\rho$ for any $\rho \models \phi$ and $D \subseteq \mathcal{A}$.*

The usual way to prove soundness is to show that if $\phi \vdash t = u$ and ϕ is $\Uparrow$-closed then $t \sim^{\phi} u$. However as it is already noticed in [17] this approach will not work specially for GUARD and DEADLINE. For example, in order to derive $\phi \vdash \gamma \to t = u$, we need to show $\phi \wedge \gamma \vdash t = u$ and $\phi \wedge \neg\gamma \vdash \mathbf{0} = u$. Note that even if ϕ is $\Uparrow$-closed, $\phi \wedge \gamma$ may not be $\Uparrow$-closed. For this reason we will first define an intermediate bisimulation relation, called *bisimulation up to d* denoted as $\sim_d^{\nabla}$. We start by defining $\sim_d^{\nabla}$ formally.

Definition 1 (∇-bisimulation up to d). *Two states p and q are* ∇-bisimilar up to $\hat{d}$ *for $\hat{d} \in \mathbb{R}_{\geq 0}$, notation $p \sim_{\hat{d}}^{\nabla} q$, if there is a family of symmetric relations $R_d \subseteq \mathcal{S} \times \mathcal{S}$, $0 \leq d \leq \hat{d}$ such that*

1. *$\forall d' \in \mathbb{R}_{\geq 0}, d' < d$, if $(p, q) \in R_d$ and $p \xrightarrow{d'} p'$ then $\exists q' : q \xrightarrow{d'} q'$ and $(p', q') \in R_{d-d'}$.*
2. *$\forall l \in \{\measuredangle\} \cup \mathcal{A}_{\nabla}$, if $(p, q) \in R_d$ and $p \xrightarrow{l} p'$ then $\exists q' : q \xrightarrow{l} q'$ and $(p', q') \in R_d$.*
3. *$\forall a \in \mathcal{A}$, if $(p, q) \in R_d$ and $p \xrightarrow{a} p'$ then $\exists q' : q \xrightarrow{a} q'$ and $p' \sim^{\nabla} q'$.*

Lemma 4. 1. *If $p \sim_d^{\nabla} q$ for all $d \in \mathbb{R}_{\geq 0}$ then $p \sim^{\nabla} q$.*
2. *Let ρ_i and d_i, $0 \leq i \leq n$, be s.t. $\rho_{i+1} = \rho_i + d_i$, $0 \leq i < n$. If $(t, D)\rho_i \sim_{d_i}^{\nabla} (u, D)\rho_i$ for all i such that $0 \leq i \leq n$, then $(t, D)\rho_0 \sim_d^{\nabla} (u, D)\rho_0$ where $d = d_0 + \cdots + d_n$.*
3. *$(t, D)\rho \sim_{\hat{d}}^{\nabla} (u, D)\rho$ implies $(t, D)(\rho + \hat{d}') \sim_{\hat{d}''}^{\nabla} (t, D)(\rho + \hat{d}')$ for any $\hat{d}' \leq \hat{d}$ and $\hat{d}'' \leq \hat{d} - \hat{d}'$.*
4. *$\sim_d^{\nabla}$ is transitive.*

Proof. Proofs of items 1 and 2 proceed as [17, Lemma 4.10]. Proof of item 4 follows standard arguments. For item 3, suppose $\{R_d\}_{d\le\hat{d}}$ witnesses $(t,D)\rho \sim^{\nabla}_{\hat{d}} (u,D)\rho$. First notice that for $((t,D)\rho,(u,D)\rho) \in R_{\hat{d}}$, $(t,D)\rho \xrightarrow{\nabla_{\mathcal{A}}} (t,\mathcal{A})\rho \xrightarrow{\hat{d}'} (t,\mathcal{A})(\rho{+}\hat{d}') \xrightarrow{\Delta} (t,\varnothing)(\rho{+}\hat{d}') \xrightarrow{\nabla_D} (t,D)(\rho{+}\hat{d}')$, (by DROP, DELAY, and UNDROP in Fig. 2) implies, by Def. 1, that $(u,D)\rho \xrightarrow{\nabla_{\mathcal{A}}} (u,\mathcal{A})\rho \xrightarrow{\hat{d}'} (u,\mathcal{A})(\rho{+}\hat{d}') \xrightarrow{\Delta} (u,\varnothing)(\rho{+}\hat{d}') \xrightarrow{\nabla_D} (u,D)(\rho{+}\hat{d}')$ and $((t,D)(\rho{+}\hat{d}'),(u,D)(\rho{+}\hat{d}')) \in R_{\hat{d}-\hat{d}'}$. It is now straightforward to show that $\{R_{(\hat{d}-\hat{d}')+d}\}_{d\le\hat{d}''}$ witnesses $t(\rho+\hat{d}') \sim^{\nabla}_{\hat{d}''} u(\rho{+}\hat{d}')$.

In the following lemmas we state some properties of deadlines and guards, which will be used later to prove soundness.

Lemma 5. *For $\hat{d}, d \in \mathbb{R}_{\ge 0}$ and $D \subseteq \mathcal{A}$*

1. $(\delta : t, D)\rho \sim^{\nabla}_{\hat{d}} (t,D)\rho$ *if for all* $d < \hat{d} : \rho + d \models \neg\delta$
2. $(\delta : t, D)\rho \sim^{\nabla}_{\hat{d}} (\mathbf{tt} : t, D)\rho$ *if for all* $d < \hat{d} : \rho + d \models \delta$
3. $(t,D)\rho \sim^{\nabla}_{\hat{d}} (u,D)\rho$ *implies* $(\mathbf{tt} : t, D)\rho \sim^{\nabla}_{\hat{d}} (\mathbf{tt} : u, D)\rho$
4. $(\gamma \to t, D)\rho \sim^{\nabla}_{\hat{d}} (t,D)\rho$ *if for all* $d \le \hat{d} : \rho + d \models \gamma$
5. $(\gamma \to t, D)\rho \sim^{\nabla}_{\hat{d}} (\mathbf{0}, D)\rho$ *if for all* $d \le \hat{d} : \rho + d \models \neg\gamma$

Proof. It is routine to prove that families $\{R_d \cup R_d^{-1}\}_{0\le d\le\hat{d}}$, respectively defined in the following, satisfy conditions of Def. 1.

1. $R_d = \{((\delta : t, D)\rho, (t,D)\rho) \mid \forall d' < d : \rho + d \models \neg\delta\}$
2. $R_d = \{((\delta : t, D)\rho, (\mathbf{tt} : t, D)\rho) \mid \forall d' < d : \rho + d \models \delta\}$
3. $R_d = \{((\mathbf{tt} : t, D)\rho, (\mathbf{tt} : u, D)\rho) \mid (t,D)\rho \sim^{\nabla}_{d} (u,D)\rho\}$
4. $R_d = \{((\gamma \to t, D)\rho, (t,D)\rho) \mid \forall d' \le d : \rho + d' \models \gamma\}$
5. $R_d = \{((\gamma \to t, D)\rho, (\mathbf{0}, D)\rho) \mid \forall d' \le d : \rho + d' \models \neg\gamma\}$

Soundness is standing on the following lemmas.

Lemma 6. *If $\phi \vdash t = u$ then, $(t,D)\rho \sim^{\nabla}_{\hat{d}} (u,D)\rho$ for all $D \subseteq \mathcal{A}$, ρ, and $d \in \mathbb{R}_{\ge 0}$ such that $\forall d \le \hat{d} : \rho + d \models \phi$.*

Proof. The proof proceeds by induction on the depth of the proof tree. The base case corresponds to all axioms. That is, for every axiom $\phi \vdash t = u$ find a family $\{R_d\}_{d\le\hat{d}}$ witnessing $(t,D)\rho \sim^{\nabla}_{\hat{d}} (u,D)\rho$. This is routine and we omit it.

For the induction step, we consider the inference rules separately. For each rule, we assume that the lemma holds in its premises and prove that it also holds in its conclusion. We only show a few representative cases. In particular, soundness of UFI is proved in Lemma 7.

ACTION: By induction $(t,D)\rho \sim^{\nabla}_{\hat{d}} (u,D)\rho$, for any D, ρ, $\hat{d}$, s.t. $\forall d \le \hat{d} : (\rho{+}d) \models \phi{\downarrow_{\mathbf{x}}}{\Uparrow}$. Since ${\phi\downarrow_{\mathbf{x}}}{\Uparrow}$ is $\Uparrow$-closed, by Lemma 4.1, $(t,D)\rho \sim^{\nabla} (u,D)\rho$. We show that $\{R_d\}_{d\le\hat{d}}$ witnesses $(a(\mathbf{x}).t, D)\rho \sim^{\nabla}_{\hat{d}} (a(\mathbf{x}).u, D)\rho$ for all ρ s.t. $\forall d \le \hat{d} : (\rho{+}d) \models \phi$, where

$$R_d = \{((a(\mathbf{x}).t, D)\rho, (a(\mathbf{x}).u, D)\rho) \mid D \subseteq \mathcal{A} \wedge \forall d' \le \hat{d} - d : \rho{+}d' \models \phi \wedge$$

$$\forall d' \le d : (t, D)(\rho{+}d')\{\mathbf{x}{:=}0\} \sim^{\nabla} (u, D)(\rho{+}d')\{\mathbf{x}{:=}0\}\}.$$

Assume $((a(\mathbf{x}).t, D)\rho, (a(\mathbf{x}).u, D)\rho) \in R_d$. We do case analysis on all the four possible type of transitions.

delay transition: By rule DELAY (Fig. 2), we have (for any $d'' \le d$) that

$$(a(\mathbf{x}).t, D)\rho \xrightarrow{d''} (a(\mathbf{x}).t, D)(\rho{+}d'') \text{ and}$$
$$(a(\mathbf{x}).u, D)\rho \xrightarrow{d''} (a(\mathbf{x}).u, D)(\rho{+}d'')$$

It remains to show that

$$((a(\mathbf{x}).t, D)(\rho{+}d''), (a(\mathbf{x}).u, D)(\rho{+}d'')) \in R_{d-d''}$$

Since $\forall d' \le \hat{d} - d : \rho{+}d' \models \phi$ holds by assumption, $\forall d' \le \hat{d} - d : (\rho{+}d')\{\mathbf{x}{:=}0\} \models \phi{\downarrow_{\mathbf{x}}}{\Uparrow}$ also hods by Def. of$\downarrow_{\mathbf{x}}$ and$\Uparrow$. By induction hypothesis and observation above, $\forall d' \le \hat{d} - d : (t, D)(\rho{+}d')\{\mathbf{x}{:=}0\} \sim^{\nabla} (u, D)(\rho{+}d')\{\mathbf{x}{:=}0\}$. In particular,

$$\forall d''' \le \hat{d} - (d - d'') : \ (\rho{+}d''{+}d''')\{\mathbf{x}{:=}0\} \models \phi{\downarrow_{\mathbf{x}}}{\Uparrow}$$

$\wedge$

$$(t, D)(\rho{+}d''{+}d''')\{\mathbf{x}{:=}0\} \sim^{\nabla} (u, D)(\rho{+}d''{+}d''')\{\mathbf{x}{:=}0\}$$

By Def. of $R_{d-d''}$, we finally have that

$$((a(\mathbf{x}).t, D)(\rho{+}d''), (a(\mathbf{x}).u, D)(\rho{+}d'')) \in R_{d-d''}$$

drop and undrop transition: Let $l \in \{\not{\Delta}\} \cup \mathcal{A}_{\nabla}$. Then, by DROP or UNDROP (Fig. 2), $(a(\mathbf{x}).t, D)\rho \xrightarrow{l} (a(\mathbf{x}).t, E)\rho$ and $(a(\mathbf{x}).u, D)\rho \xrightarrow{l} (a(\mathbf{x}).u, E)\rho$ for any $E \subseteq \mathcal{A}$. Besides, $((a(\mathbf{x}).t, E)\rho, (a(\mathbf{x}).u, E)\rho) \in R_d$, since for all $d' \le \hat{d} - d$, $(\rho{+}d') \models \phi$ implies $(\rho{+}d')\{\mathbf{x}{:=}0\} \models \phi{\downarrow_{\mathbf{x}}}{\Uparrow}$ and by induction

$$(t, E)(\rho{+}d')\{\mathbf{x}{:=}0\} \sim^{\nabla} (u, D)(\rho{+}d')\{\mathbf{x}{:=}0\}$$

discrete transition: By ACTION (Fig.2), $(a(\mathbf{x}).t, D)\rho \xrightarrow{a} (t, \varnothing)\rho\{\mathbf{x}{:=}0\}$ and $(a(\mathbf{x}).u, D)\rho \xrightarrow{a} (u, \varnothing)\rho\{\mathbf{x}{:=}0\}$. Moreover, since $\rho \models \phi$ by assumption, $\rho\{\mathbf{x}{:=}0\} \models \phi{\downarrow_{\mathbf{x}}}{\Uparrow}$, and hence $(t, \varnothing)\rho\{\mathbf{x}{:=}0\} \sim^{\nabla} (u, D)\rho\{\mathbf{x}{:=}0\}$ by induction.

DEADLINE: We need to prove that $(\delta : t, D)\rho \sim^{\nabla}_{d} (u, D)\rho$ for all ρ s.t. $\forall d \le \hat{d} : (\rho + d) \models \phi$. The interval $[\rho, \rho + \hat{d})$ can be divided by regions into finitely many subintervals $[\rho_0, \rho_1)$, $[\rho_1, \rho_1]$, (ρ_1, ρ_2), $\ldots$, $[\rho_n, \rho_n]$, (ρ_n, ρ_{n+1}), where $\rho_0 = \rho$ and $\rho_{i+1} = \rho_i + d_i$ for some $d_0, \ldots, d_n$ s.t. $\sum_{i=0}^{n} d_i = d$ in a way that each point $[\rho_i, \rho_i]$, or interval $(\rho_i, \rho_i{+}d_i)$ is entirely contained in a region (so they are entirely contained in $\phi \wedge \neg\delta$ or in $\phi \wedge \delta$). By Lemma 4.2, it is enough to prove $(\delta : t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$ for all $1 \le i \le n$. We only consider the case of intervals $(\rho_i, \rho_i{+}d_i)$, the others follow in a similar manner.
Case $(\rho_i, \rho_i{+}d_i) \models \phi \wedge \neg\delta$. By Lemma 5.1 $(\delta : t, D)\rho_i \sim^{\nabla}_{d_i} (t, D)\rho_i$. Besides, by induction and Lemma 4.3. $(t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$. Hence, by transitivity of $\sim^{\nabla}_{d_i}$, $(\delta : t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$.

Case $(\rho_i, \rho_i{+}d_i) \models \phi \wedge \delta$. By Lemma 5.2, $(\delta : t, D)\rho_i \sim^{\nabla}_{d_i} (\mathbf{tt} : t, D)\rho_i$ By induction and Lemma 4.3, $(\mathbf{tt} : t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$. Therefore, by transitivity of $\sim^{\nabla}_{d_i}$, we have: $(\delta : t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$.

GUARD: Using similar argument as in DEADLINE, we only need to prove that $(\gamma{\rightarrow}t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$ for all $1 \leq i \leq n$.

Case: $(\rho_i, \rho_i{+}d_i) \models \phi \wedge \gamma$

$\Rightarrow$ {By Lemma 5.4}
$(\gamma{\rightarrow}t, D)\rho_i \sim^{\nabla}_{d_i} (t, D)\rho_i$

$\Rightarrow$ {By transitivity of $\sim^{\nabla}_{d_i}$ since $(t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$, by induction and Lemma 4.3.}
$(\gamma{\rightarrow}t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$

Case: $(\rho_i, \rho_i{+}d_i) \models \phi \wedge \neg\gamma$

$\Rightarrow$ {By Lemma 5.5}
$(\gamma{\rightarrow}t, D)\rho_i \sim^{\nabla}_{d_i} (\mathbf{0}, D)\rho_i$

$\Rightarrow$ {By induction and Lemma 4.3, $(\mathbf{0}, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$. Then, by transitivity of $\sim^{\nabla}_{d_i}$, we have:}
$(\gamma{\rightarrow}t, D)\rho_i \sim^{\nabla}_{d_i} (u, D)\rho_i$

SUBSTITUTION on choice: Suppose $(t, D)\rho \sim^{\nabla}_{\hat{d}} (u, D)\rho$, and suppose $\{R_d\}_{0 \leq d \leq \hat{d}}$ witnesses it. We show that $\{R'_d\}_{0 \leq d \leq \hat{d}}$ with $R'_d = \{((t{+}s, D)\rho, (u{+}s, D)\rho) \mid ((t, D)\rho, (u, D)\rho) \in R_d\}$, witnesses $(t{+}s, D)\rho \sim^{\nabla}_{\hat{d}} (u{+}s, D)\rho$. For all $d \leq \hat{d}$, suppose $((t{+}s, D)\rho, (u{+}s, D)\rho) \in R'_d$. We show the case of delay transition, the other cases are easier. For $d' < d$ we calculate:

$(t + s, D)\rho \xrightarrow{d'} (t + s, D)(\rho{+}d')$
$\Leftrightarrow$ {By definition of DELAY}
$\forall d'' < d' : (\rho{+}d'') \models \neg dl(t + s, D)$
$\Leftrightarrow$ {By definition of dl}
$\forall d'' < d' : (\rho{+}d'') \models \neg(dl(t, D) \vee dl(s, D))$
$\Leftrightarrow$ {Logic}
$\forall d'' < d' : (\rho{+}d'') \models \neg dl(t, D)$ and $\forall d'' < d' : (\rho{+}d'') \models \neg dl(s, D)$
$\Leftrightarrow$ {By definition of DELAY}
$(t, D)\rho \xrightarrow{d'} (t, D)(\rho{+}d')$ and $\forall d'' < d' : (\rho{+}d'') \models \neg dl(s, D)$
$\Rightarrow$ {Since $((t, D)\rho, (u, D)\rho) \in R_d$}
$(u, D)\rho \xrightarrow{d'} (u, D)(\rho{+}d')$, $((t, D)(\rho{+}d'), (u, D)(\rho{+}d')) \in R_{d-d'}$ and
$\forall d'' < d' : (\rho{+}d'') \models \neg dl(s, D)$

$\Leftrightarrow$ $\{$By definition of DELAY and $R'_{d-d'}\}$

$\forall d'' < d' : (\rho{+}d'') \models \neg dl(t, D), \quad \forall d'' < d' : (\rho{+}d'') \models \neg dl(s, D)$ and

$((t + s, D)(\rho{+}d'), (u + s, D)(\rho{+}d')) \in R'_{d-d'}$

$\Leftrightarrow$ {Logic}

$\forall d'' < d' : (\rho{+}d'') \models \neg(dl(u, D) \vee dl(s, D))$ and

$((t + s, D)(\rho{+}d'), (u + s, D)(\rho{+}d')) \in R'_{d-d'}$

$\Leftrightarrow$ {By definition of dl and DELAY}

$(u + s, D)\rho \xrightarrow{d'} (u + s, D)(\rho{+}d')$ and

$((t + s, D)(\rho{+}d'), (u + s, D)(\rho{+}d')) \in R'_{d-d'}$

SUBSTITUTION on urgency: First of all notice that if $(t, D)\rho \sim^{\nabla}_{\hat{d}} (u, D)\rho$ then $\rho \models gd(t, D)$ iff $\rho \models gd(u, D)$. From Lemma 4.3 it follows that $(\rho{+}d) \models gd(t, D)$ iff $(\rho{+}d) \models gd(u, D)$ for all $d < d'$. Call this observation ($\star$).

From this observation notice that $(\rho{+}d) \models gd(t, I(t) - I(u)) \Leftrightarrow \mathbf{ff}$ for all $d < d'$ and hence $(\rho{+}d) \models \neg dl(\mathbf{tt} : t, I(t) - I(u)) \Leftrightarrow dl(\mathbf{tt} : u, I(t) - I(u))$. Symmetrically, it holds for t and u exchanged. So, w.l.o.g., we will suppose that $I(t) = I(u)$.

Now, we proceed in a similar fashion as the previous case. Define $R'_d = \{((\mathbf{tt} : t, D)\rho, (\mathbf{tt} : u, D)\rho) \mid ((t, D)\rho, (u, D)\rho) \in R_d\}$ provided $\{R_d\}_{0\le d\le \hat{d}}$ witnesses $(t, D)\rho \sim^{\nabla}_{\hat{d}} (u, D)\rho$. For all $d \le \hat{d}$, suppose $((\mathbf{tt} : t, D)\rho, (\mathbf{tt} : u, D)\rho) \in R'_d$. We show the case of delay transition, the other cases are easier, and in this case we only consider $I(t) \cap D \neq \varnothing$ (and hence $I(u) \cap D \neq \varnothing$) since the case $I(t) \cap D = \varnothing$ is simpler. For $d' < d$ we calculate:

$(\mathbf{tt} : t, D)\rho \xrightarrow{d'} (\mathbf{tt} : t, D)(\rho{+}d')$

$\Leftrightarrow$ {By definition of DELAY}

$\forall d'' < d' : (\rho{+}d'') \models \neg dl(\mathbf{tt} : t, D)$

$\Leftrightarrow$ {By definition of dl}

$\forall d'' < d' : (\rho{+}d'') \models \neg((\mathbf{tt} \wedge gd(t, D)) \vee dl(t, D))$

$\Leftrightarrow$ {Logic}

$\forall d'' < d' : (\rho{+}d'') \models \neg gd(t, D)$ and $\forall d'' < d' : (\rho{+}d'') \models \neg dl(t, D)$

$\Leftrightarrow$ {By definition of DELAY}

$\forall d'' < d' : (\rho{+}d'') \models \neg gd(t, D)$ and $(t, D)\rho \xrightarrow{d'} (t, D)(\rho{+}d')$

$\Rightarrow$ {Since $((t, D)\rho, (u, D)\rho) \in R_d$ and by observation ($\star$)}

$\forall d'' < d' : (\rho{+}d'') \models \neg gd(u, D)$,

$(u, D)\rho \xrightarrow{d'} (u, D)(\rho{+}d')$, and $((t, D)(\rho{+}d'), (u, D)(\rho{+}d')) \in R_{d-d'}$

$\Leftrightarrow$ $\{$Following the inverse reasoning and by definition of $R'_{d-d'}\}$

$(\mathbf{tt} : u, D)\rho \xrightarrow{d'} (\mathbf{tt} : u, D)(\rho{+}d')$ and

$((\mathbf{tt} : t, D)(\rho{+}d'), (\mathbf{tt} : u, D)(\rho{+}d')) \in R_{d-d'}$

The next lemma states soundness of UFI which amounts to proving that every set of equation has a unique solution.

Lemma 7. *Let terms v_i ($i \in I$) contain at most variables X_i ($i \in I$) which occur only guarded. Then, if*

1. $(t_j, D)\rho \sim^{\nabla} (v_j[t_i/X_i \mid i \in I], D)\rho$ *and*
2. $(u_j, D)\rho \sim^{\nabla} (v_j[u_i/X_i \mid i \in I], D)\rho$

then

$$(t_j, D)\rho \sim^{\nabla} (u_j, D)\rho$$

for all $j \in I$, $D \subseteq \mathcal{A}$, and valuation ρ.

Proof. We show that

$$R = \{((v[\tilde{t}/\tilde{X}], D)\rho, (v[\tilde{u}/\tilde{X}], D)\rho) \mid Vars(v) \subseteq \{X_i \mid i \in I\}\}$$

is a timed bisimulation up to $\sim^{\nabla}$ (we let $[\tilde{s}/\tilde{X}]$ denote $[s_i/X_i \mid i \in I]$). First notice that R is symmetric. The proof of the transfer property proceeds by case analysis on the type of the transition. Cases ∇_A and Δ are straightforward. Case $a \in \mathcal{A}$ follows by induction on the proof tree doing case analysis on the form of v like Proposition 14, Sec. 4.5 [19]. For the delay transition, we first consider the case in which $v \equiv X_j$.

Suppose that $(X_j[\tilde{t}/\tilde{X}], D)\rho \xrightarrow{d} (X_j[\tilde{t}/\tilde{X}], D)(\rho + d)$. Notice that $X_j[\tilde{t}/\tilde{X}] \equiv t_j$ and $(t_j, D)\rho \sim^{\nabla} (v_j[\tilde{t}/\tilde{X}], D)\rho$. Hence $(v_j[\tilde{t}/\tilde{X}], D)\rho \xrightarrow{d} (v_j[\tilde{t}/\tilde{X}], D)(\rho+d)$ and $(X_j[\tilde{t}/\tilde{X}], D)(\rho+d) \sim^{\nabla} (v_j[\tilde{t}/\tilde{X}], D)(\rho + d)$ (†). By DELAY, $\forall d' < d \quad \rho + d' \models \neg dl(v_j[\tilde{t}/\tilde{X}], \mathcal{A} - D)$. Since all X_i are guarded in v_j, $dl(v_j[\tilde{t}/\tilde{X}], \mathcal{A} - D) = dl(v_j[\tilde{u}/\tilde{X}], \mathcal{A} - D)$ and from here $(v_j[\tilde{u}/\tilde{X}], D)\rho \xrightarrow{d} (v_j[\tilde{u}/\tilde{X}], D)(\rho + d)$. Noticing that $v_j[\tilde{u}/\tilde{X}] \equiv u_j \equiv X_j[\tilde{u}/\tilde{X}]$, we have that $(X_j[\tilde{u}/\tilde{X}], D)\rho \xrightarrow{d} (X_j[\tilde{u}/\tilde{X}], D)(\rho + d)$ and $(v_j[\tilde{u}/\tilde{X}], D)(\rho + d) \sim^{\nabla} (X_j[\tilde{u}/\tilde{X}], D)(\rho + d)$ (‡). Using (†) and (‡), conclude that $(X_j[\tilde{t}/\tilde{X}], D)(\rho+d) \sim^{\nabla} \circ R \circ \sim^{\nabla} (X_j[\tilde{u}/\tilde{X}], D)(\rho+d)$. From here and Theorem 1 of [10], $dl(t_j, D) \Leftrightarrow dl(u_j, D)$ for any $j \in I$ and $D \subseteq \mathcal{A}$. Using this fact and induction on the structure of v, it is routine to prove that $dl(v[\tilde{t}/\tilde{X}], D) \Leftrightarrow dl(v[\tilde{u}/\tilde{X}], D)$. Using this equivalence, the proof of the transfer property on the delay transition for an arbitrary v is straightforward.

6 Completeness Proof

In this section, we present the completeness theorem of the proof system, that is, whenever $t \sim^{\phi} u$ then $\phi \vdash t = u$. The proof of the theorem follows the arguments used by Milner [18, 20]. That is, we will first show that any term t can provably satisfy a special kind of equation E, called *standard equation* (Lemma 8). Next we prove that if $t \sim^{\phi} u$ then both t and u provably satisfy a common standard equation E (Lemma 9). Finally from these two results we shall conclude $\phi \vdash t = u$ (Theorem 2).

6.1 Transforming Sets of Equations

First, we formally define equations, standard equations and what it means for a term to provably $\tilde{\phi}$-satisfy an equation.

Definition 2. *An* equation *set*

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

is a finite non-empty indexed set of declarations, where the X_i's are pairwise distinct process variables, and the u_i's are terms.

Definition 3. *Given a vector of conditions $\tilde{\phi} = \{\phi_i \mid i{\in}I\}$ and a vector of terms $\tilde{t} = \{t_i \mid i{\in}I\}$, we say that*

$$\tilde{t} \textit{ provably } \tilde{\phi}\textit{-satisfies a set of equation} E : \{X_i = u_i \mid i{\in}I\}$$

iff, for all $i{\in}I$,

$$\phi_i \vdash t_i = u_i[\tilde{\phi}{\rightarrow}\tilde{t}/\tilde{X}]$$

Alternatively, we say that t provably ϕ-satisfies E, to mean that $\tilde{t}$ provably $\tilde{\phi}$-satisfies E when $\phi = \phi_1$ and $t = t_1$.

Definition 4. *An equation set E is* standard *iff each equation of E is of the form:*

$$X_i \;=\; \sum_{a\in\mathcal{A}} \delta_{ia} : \sum_{k\in K_{ia}} \gamma_{ika} \rightarrow a(\mathbf{x}_{ika}).X_{f(i,k,a)} + \sum_{W\in\mathbf{V}} \delta_{iW} : \gamma_{iW} \rightarrow W \tag{7}$$

where, the vector X_i is disjoint from the set V, for all $a \in \mathcal{A}$, $\delta_{ia} \Rightarrow \bigvee_{k\in K_{ia}} \gamma_{ika}$, and for all $W \in \mathbf{V}$, $\delta_{iW} \Rightarrow \gamma_{iW}$. We call X_i the formal *variables of E, and $W \in \mathbf{V}$ the* free *variables of E. The set of equation E is called* closed *if $\mathbf{V} = \varnothing$.*

For example, $\{X_1 = (x_1{\geq}1) : (x_1{\geq}1) \rightarrow a_1(x_2).X_2 + \mathbf{ff} : \mathbf{tt} \rightarrow W, X_2 = (2{\leq}x_2{<}3) : ((x_2{<}x_1) \rightarrow a_2(x_1).X_1 + (x_2{\geq}x_1) \rightarrow a_2(x_2).X_2)\}$ is a standard set of equation, with formal variable set $\mathbf{X} = \{X_1, X_2\}$ and free variable set $\mathbf{V} = \{W\}$.

Lemma 8. *For any guarded term t with free variables $\mathbf{V}$ there exists a set of standard equations E, with free process variables in $\mathbf{V}$, which is provably $\mathbf{tt}$-satisfied by t. In particular, if t is closed so is E.*

Proof. Like in [20], we proceed by induction on the structure of t. We only report the most relevant cases

Case $t \equiv \mathbf{0}$: It is easy to check that E containing the only equation

$$X = \sum_{a\in\mathcal{A}} \mathbf{ff} : \mathbf{ff} \rightarrow a(\varnothing).X + \sum_{W\in\mathbf{V}} \mathbf{ff} : \mathbf{ff} \rightarrow W$$

is satisfied by $\mathbf{0}$ (recall Lemma 1.15).

Case $t \equiv X$, $X \in \mathbf{V}$: Again by Lemma 1.15 the equation

$$E : \Big\{ X_1 = \sum_{a \in \mathcal{A}} \mathbf{ff} : \mathbf{ff} \to a(\varnothing).X_1 + \sum_{W \in \mathbf{V} - \{X\}} \mathbf{ff} : \mathbf{ff} \to W \;\; + \;\; \mathbf{ff} : \mathbf{tt} \to X \Big\}$$

is satisfied by X.

Case $t \equiv t' + t''$: By induction, t' and t'' satisfy sets of standard equations. Let them be E' and E'', respectively. Define the set of equations E containing all equations in E' and E'' and the new equation

$$\begin{aligned} X_1 = \sum_{a \in \mathcal{A}} (\delta'_{1a} \vee \delta''_{1a}) : \Big(& \sum_{k' \in K'_{1a}} \gamma'_{1k'a} \to a(\mathbf{x}_{1k'a}).X'_{f'(1,k',a)} \\ & + \sum_{k'' \in K''_{1a}} \gamma''_{1k''a} \to a(\mathbf{x}_{1k''a}).X''_{f''(1,k'',a)} \Big) \\ + \sum_{W \in \mathbf{V}} \big(\delta'_W \vee \delta''_W\big) : \big(\gamma'_{iW} \vee \gamma''_{iW}\big) \to W & \end{aligned} \tag{8}$$

provided

$$X'_1 = \sum_{a \in \mathcal{A}} \delta'_{1a} : \sum_{k' \in K'_{1a}} \gamma'_{1k'a} \to a(\mathbf{x}_{1k'a}).X'_{f'(1,k',a)} + \sum_{W \in \mathbf{V}} \delta'_{1W} : \gamma'_{1W} \to W \tag{9}$$

in E' and similarly X''_1 in E''. Call r_1 the right-hand side in equation (8). Similarly, call r'_1 and r''_1 the respective right-hand sides in equations for X'_1 in E' and X''_1 in E'' (see equation (9)). Using Lemma 2 and 3 , the reader should be able to show that $r_1 = r'_1 + r''_1$ from which this case is proved.

Case $t \equiv \mathbf{fix} X t'$: By induction, t' satisfies a set of standard equations E' with free variables in $\mathbf{V}$, $X \in \mathbf{V}$. For every equation $X_i = r'_i$ in E' (definitions for E' are like in (7)) we define a new equation $X_i = r_i$ in E where each r_i is defined from r'_i by appropriately replacing variable X. For the distinguished variable X_1 we define:

$$X_1 = \sum_{a \in \mathcal{A}} \delta_{1a} : \sum_{k \in K_{1a}} \gamma_{1ka} \to a(\mathbf{x}_{1ka}).X_{f(1,k,a)} + \sum_{W \in \mathbf{V} - \{X\}} \delta_{1W} : \gamma_{1W} \to W$$

Call r_1 the right-hand side of the equation. Notice that r_1 is like r'_1 only that it omits the summand '$\mathbf{ff} : \mathbf{ff} \to X$' (since X does not occur unguarded in t', X must be guarded by predicate **ff**). For $1 < i \le |E'|$ we calculate the new equation as follows (calculations make use of Lemma 2 and 3).

$$\begin{aligned} X_i =& r'_i[r_1/X] \\ =& \sum_{a \in \mathcal{A}} \delta_{ia} : \sum_{k \in K_{ia}} \gamma_{ika} \to a(\mathbf{x}_{ika}).X_{f(i,k,a)} \;\; + \sum_{W \in \mathbf{V} - \{X\}} \delta_{iW} : \gamma_{iW} \to W \\ & + \; \delta_{iX} : \gamma_{iX} \to \Big(\sum_{a \in \mathcal{A}} \delta_{1a} : \sum_{k \in K_{1a}} \gamma_{1ka} \to a(\mathbf{x}_{1ka}).X_{f(1,k,a)} \;\; + \sum_{W \in \mathbf{V} - \{X\}} \delta_{1W} : \gamma_{1W} \to W \Big) \\ =& \sum_{a \in \mathcal{A}} \delta_{ia} : \sum_{k \in K_{ia}} \gamma_{ika} \to a(\mathbf{x}_{ika}).X_{f(i,k,a)} \;\; + \sum_{W \in \mathbf{V} - \{X\}} \delta_{iW} : \gamma_{iW} \to W \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{a\in\mathcal{A}}((\delta_{iX}\vee\delta_{1a})\wedge\gamma_{iX}\wedge\bigvee_{k\in K_{1a}}\gamma_{1ka}) : \sum_{k\in K_{1a}}(\gamma_{iX}\wedge\gamma_{1ka})\to a(\mathbf{x}_{1ka}).X_{f(1,k,a)}\\
&+ \sum_{W\in\mathbf{V}-\{X\}}((\delta_{iX}\vee\delta_{1W})\wedge\gamma_{iX}\wedge\gamma_{1W}) : (\gamma_{iX}\wedge\gamma_{1W})\to W\\
=&\sum_{a\in\mathcal{A}}(\delta_{ia}\vee((\delta_{iX}\vee\delta_{1a})\wedge\gamma_{iX}\wedge\bigvee_{k\in K_{1a}}\gamma_{1ka})) :\\
&\qquad\sum_{k\in K_{1a}}(\gamma_{iX}\wedge\gamma_{1ka})\to a(\mathbf{x}_{1ka}).X_{f(1,k,a)} + \sum_{k\in K_{ia}}\gamma_{ika}\to a(\mathbf{x}_{ika}).X_{f(i,k,a)}\\
&+ \sum_{W\in\mathbf{V}-\{X\}}(\delta_{iW}\vee((\delta_{iX}\vee\delta_{1W})\wedge\gamma_{iX}\wedge\gamma_{1W})) : (\gamma_{iW}\vee(\gamma_{iX}\wedge\gamma_{1W}))\to W
\end{aligned}
$$

Let t'_i, $i\in I$, be the set of terms that witnesses that t' $(= t'_1)$ satisfies E'. Noticing the $r_1 + \mathbf{ff} : \mathbf{ff}\to X \equiv r'_1$ the reader should not find difficulties on proving that the set of terms $t_i \equiv t'_i[t/X]$, $i\in I$, witnesses that t $(= t_1)$ satisfies E. (The proof needs REC).

6.2 Completeness of the Proof System for Guarded Terms

Lemma 9. *For closed terms t and u, if $t\sim^{\phi} u$ then there exists ϕ' such that $\phi\Rightarrow\phi'$ and a standard closed equation set E which is provably ϕ'-satisfied by both t and u.*

Proof. Let the set of clock variables of t, u be $\mathbf{x}$, $\mathbf{y}$, respectively, with $\mathbf{x}\cap\mathbf{y} = \varnothing$. According to Lemma 8 let E_1 and E_2 be the standard closed equation sets for which t and u provably **tt**-satisfy, respectively:

$$
\begin{aligned}
&E_1 : \{X_i = \textstyle\sum_{a\in\mathcal{A}}\delta_{ia} : \sum_{k\in K_{ia}}\gamma_{ika}\to a(\mathbf{x}_{ika}).X_{f(i,k,a)} \mid i\in I\}\\
&E_2 : \{Y_j = \textstyle\sum_{a\in\mathcal{A}}\delta'_{ja} : \sum_{l\in L_{ja}}\gamma'_{jla}\to a(\mathbf{x}_{ila}).X_{g(j,l,a)} \mid j\in J\}
\end{aligned}
$$

So there are $\tilde{t} = \{t_i \mid i\in I\}$ and $\tilde{u} = \{u_j \mid j\in J\}$ such that $t_1 = t, u_1 = u$, and

$$t_i = \textstyle\sum_{a\in\mathcal{A}}\delta_{ia} : \sum_{k\in K_{ia}}\gamma_{ika}\to a(\mathbf{x}_{ika}).t_{f(i,k,a)} \tag{10}$$

$$u_j = \textstyle\sum_{a\in\mathcal{A}}\delta'_{ja} : \sum_{l\in L_{ja}}\gamma'_{jla}\to a(\mathbf{x}_{ila}).u_{g(j,l,a)} \tag{11}$$

For each pair of i, j let $\Phi_{ij} = \{\omega\in\mathcal{RC}(\mathbf{xy}) \mid t_i\sim^{\omega\restriction} u_j\}$. Set $\varphi_{ij} = \bigvee\Phi_{ij}$. By the definition of Φ_{ij}, φ_{ij} is the weakest condition over which t_i and u_j are symbolically bisimilar, that is $\psi\Rightarrow\varphi_{ij}$ for any ψ such that $t_i\sim^{\psi} u_j$. In particular, $\phi\Rightarrow\varphi_{11}$. Also, for any $\omega\in\Phi_{ij}$ and $a\in\mathcal{A}$, $\omega\models\delta_{ia}\Leftrightarrow\delta'_{ja}$.

For each $a\in\mathcal{A}$ and $\omega\in\Phi_{ij}$, let

$$I_{ij}^{a\omega} = \{(k,l) \mid \omega\models\gamma_{ika}\wedge\gamma'_{jla} \text{ and } t_{f(i,k,a)}\sim^{\omega\downarrow_{\mathbf{x}_{ik}\mathbf{y}_{jl}}\restriction} u_{g(j,l,a)}\}$$

and define the set E containing equations

$$Z_{ij} = \textstyle\sum_{a\in\mathcal{A}}\delta_{ia} : \sum_{\omega\in\Phi_{ij}}\omega\to\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).Z_{f(i,k,a)g(j,l,a)}$$

We claim that E is provably φ_{11}-satisfied by t (resp. u) when each Z_{ij} is instantiated with t_i (resp. u_j) over φ_{ij}. We only prove the case of t; the case of u proceeds in a similar manner. For each i and j, we have to prove that

$$\varphi_{ij}\vdash t_i = \textstyle\sum_{a\in\mathcal{A}}\delta_{ia} : \sum_{\omega\in\Phi_{ij}}\omega\to\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\varphi_{f(i,k,a)g(j,l,a)}\to t_{f(i,k,a)}$$

Because of (10) and soundness on the one hand, and CHOICE and Lemma 1.10 on the other hand, it suffices to prove the equivalence of each a-summand, that is, it suffices to prove that for all $a \in \mathcal{A}$,

$$\varphi_{ij} \vdash \textstyle\sum_{k\in K_{ia}} \gamma_{ika} \to a(\mathbf{x}_{ika}).t_{f(i,k,a)} = \\ \textstyle\sum_{\omega\in\Phi_{ij}} \omega \to \sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\varphi_{f(i,k,a)g(j,l,a)} \to t_{f(i,k,a)}$$

Since the elements of Φ_{ij} are mutually disjoint, by Lemmas 1.1 and 1.18, it is sufficient to show that, for each $\omega \in \Phi_{ij}$,

$$\omega \vdash \textstyle\sum_{k\in K_{ia}} \gamma_{ika} \to a(\mathbf{x}_{ika}).t_{f(i,k,a)} = \sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\varphi_{f(i,k,a)g(j,l,a)} \to t_{f(i,k,a)}$$

By definition of $I_{ij}^{a\omega}$, we have that $t_{f(i,k,a)} \sim^{\omega\downarrow_{\mathbf{x}_{ika}\mathbf{y}_{jla}}\Uparrow} u_{g(j,l,a)}$. Hence, from the definition of $\Phi_{f(i,k,a)g(j,l,a)}$,

$$\omega\downarrow_{\mathbf{x}_{ika}\mathbf{y}_{jla}}\Uparrow \Rightarrow \varphi_{f(i,k,a)g(j,l,a)} \tag{12}$$

Under the assumption ω holds, we calculate,

$$\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\varphi_{f(i,k,a)g(j,l,a)} \to t_{f(i,k,a)}$$

$=$ {By Lemma 1.17 from left to right}

$$\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\omega\downarrow_{\mathbf{x}_{ik}\mathbf{y}_{jl}}\Uparrow\to \varphi_{f(i,k,a)g(j,l,a)} \to t_{f(i,k,a)}$$

$=$ {Lemma 1.5 and (12)}

$$\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).\omega\downarrow_{\mathbf{x}_{ik}\mathbf{y}_{jl}}\Uparrow\to t_{f(i,k,a)}$$

$=$ {By Lemma 1.17 from right to left}

$$\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}\mathbf{y}_{jla}).t_{f(i,k,a)}$$

$=$ {By THINNING}

$$\sum_{(k,l)\in I_{ij}^{a\omega}} a(\mathbf{x}_{ika}).t_{f(i,k,a)}$$

$=$ {By Lemma 1.1 and SUBSTITUTION, since $\omega \models \gamma_{ika}$}

$$\sum_{(k,l)\in I_{ij}^{a\omega}} \gamma_{ika} \to a(\mathbf{x}_{ika}).t_{f(i,k,a)}$$

$=$ {By claim below[4] using Lemma 1.15 and S1–S4.(†)}

$$\sum_{k\in K_{ia}} \gamma_{ika} \to a(\mathbf{x}_{ika}).t_{f(i,k,a)}$$

It only remains to prove equality (†) and the proof of the lemma will be complete. For this, notice that $\{k \mid (k,l) \in I_{ij}^{a\omega}\} \subseteq K_{ia}$. Therefore, to prove (†), it suffices to show the following claim.

Claim. If $k \in K_{ia} - \{k \mid (k,l) \in I_{ij}^{a\omega}\}$ then $\omega \Rightarrow \neg\gamma_{ika}$.

Proof of claim. By contradiction suppose $\omega \Rightarrow \neg\gamma_{ika}$ is not the case, which is equivalent to say $\omega \Rightarrow \gamma_{ika}$ since ω is a region. Suppose $k \in K_{ia}$. Then $t \xrightarrow{a,\gamma,\delta,\mathbf{x}_{ika}} t_{f(i,k,a)}$ with $\omega \Rightarrow \gamma$. Besides, since $\omega \in \Phi_{ij}$, $t_i \sim^{\omega\Uparrow} u_j$. Then, there is a $(\omega \Uparrow \wedge\gamma)$-partition Φ s.t. for all $\phi \in \Phi$, $u_i \xrightarrow{a,\gamma',\delta',\mathbf{y}'} u'$, $\phi \Rightarrow \gamma'$ and $t_{f(i,k,a)} \sim^{\phi\Uparrow_{\mathbf{x}_{ika}\mathbf{y}'}} u'$. In particular this occurs for some ϕ s.t. $\omega \Rightarrow \phi$. Then, by soundness of equality, there must exist a summand $\gamma'_{jla} \to a(\mathbf{y}_{jla}).u_{g(j,l,a)}$ in (11), with $\mathbf{y}_{jla} = \mathbf{y}'$, $u_{g(j,l,a)} \sim^{\omega\Uparrow_{\mathbf{y}_{jla}}} u'$, and $\omega \Rightarrow (\gamma' \wedge \gamma'_{jla})$. But then $\omega \Rightarrow (\gamma_{ika} \wedge \gamma'_{jla})$, and hence, $(k,l) \in I_{ij}^{a\omega}$, by def. of $I_{ij}^{a\omega}$.

Lemma 10. *If both t and u provably ϕ-satisfy an equation set E then $\phi \vdash t = u$.*

The proof of Lemma 10 proceeds as in Proposition 5.4 [17]. The completeness of the proof system is a direct consequence of Lemma 9 and Lemma 10:

Theorem 2. *For closed terms t and u, if $t \sim^{\phi} u$ then $\phi \vdash t = u$.*

Proof. Since $t \sim^{\phi} u$ and t and u are closed terms, by Lemma 9, exists a set of equations E which is ϕ'-satisfied by t and u and $\phi' \Rightarrow \phi$. By Lemma 10, $\phi' \vdash t = u$. Since $\phi \Rightarrow \phi'$, $\phi \vdash t = u$ by RESTRICTION.

6.3 Completeness of the Proof System for all $\mathbb{A}$

In the following we show completeness for all closed terms. The strategy of proof is similar to [20] and it stands on the following lemma.

Lemma 11. *Let t be a term in which X occurs free and unguarded only outside the scope of a recursion. Then, there are predicates δ and γ and also a term u in which X does not occur unguarded such that $t = u + \delta : \gamma \to X$.*

Proof. The proof proceed by structural induction. For t having the form 0, $a(\mathbf{x}).t'$, $\mathbf{fix}Zt'$ (Z been X or any other variable), or a variable different from X, the lemma holds trivially. For the other cases we proceed as follows.

Case $t \equiv X$: Using axioms S1 and Lemmas 1.1 and 1.2, it is easy to show that $X = 0 + \mathbf{ff} : \mathbf{tt} \to X$.

Case $t \equiv t_1 + t_2$: By induction $t_i = u_i + \delta_i : \gamma_i \to X$, $i = 1, 2$. Using Lemmas 1.12 and 2, and axioms S3 and S4, it is possible to show that $t = u_1 + u_2 + ((\delta_1 \wedge \gamma_1) \vee (\delta_2 \wedge \gamma_2)) : (\gamma_1 \vee \gamma_2) \to X$.

Case $t \equiv \gamma \to t'$: By induction $t' = u' + \delta' : \gamma' \to X$. Then $t = \gamma \to u' + \delta : (\gamma \wedge \gamma') \to X$ by Lemmas 1.6 and 1.5.

Case $t \equiv \delta : t'$: By induction $t' = u' + \delta' : \gamma' \to X$. Then $t = \delta : u' + (\delta \vee \delta') : \gamma' \to X$ by Lemmas 1.9 and 1.8.

[4] The proof of Proposition 5.2 in [17] —which is comparable to our Lemma 9—incorrectly assumes that a similar step in the proof is only a result of axioms S1–S4. This is not the case and cannot be proved without the claim.

Theorem 3. *For every term t there exists a guarded term t' s.t. $t = t'$ is provable.*

Proof. By induction we actually prove that for any t there is a t' s.t.

1. X is guarded in t';
2. no free unguarded occurrence of any variable Y in t' lies within a recursion t'; and
3. $\mathbf{fix}Xt = \mathbf{fix}Xt'$

from which the theorem follows. Suppose that 1, 2, and 3 hold for every u with recursion depth less than that of t. (The case when t contains no recursion follows in a similar manner.) Take a recursion $\mathbf{fix}Yu$ in t which lies within no recursion. By induction, there is a term u' s.t. Y is guarded in u', no free unguarded recursion of any variable lies within a recursion, and $\mathbf{fix}Yu = \mathbf{fix}Yu'$. Hence, no free unguarded occurrence of a variable occurs within a recursion in $u'[\mathbf{fix}Yu'/Y]$.

Let t_1 be the result of simultaneously replacing every top recursion $\mathbf{fix}Yu$ in t by $u'[\mathbf{fix}Yu'/Y]$. Clearly $t_1 = t$. Moreover, no free unguarded occurrence of a variable in t_1 lies within a recursion. By Lemma 11, there are predicates δ and γ, and t_2 in which X only occurs guarded, s.t. $t_1 = t_2 + \delta : \gamma \to X$. Then

$$\mathbf{fix}Xt = \mathbf{fix}Xt_1 = \mathbf{fix}X(t_2 + \delta : \gamma \to X) \overset{\mathsf{UR}}{=} \mathbf{fix}X(t_2 + \delta : \gamma \to t_2)$$

which proves the theorem.

The following result is a consequence of Theorems 2 and 3.

Theorem 4. *For all closed $\mathbb{A}$ terms t and u, if $t \sim^{\nabla} u$ then $\phi \vdash t = u$.*

Conclusion This chapter provides a sound and complete proof system for the coarsest congruence for (finite) timed automata with deadlines that is included in bisimulation.

The result on axiomatization can be easily extended to all $\mathbb{A}$ terms by noticing that ∇-bisimulation for open terms can be characterized either by extending the operational semantics allowing $X \xrightarrow{X} \mathbf{0}$ or by extending the symbolic semantics allowing $X \xrightarrow{\mathbf{tt},\mathbf{ff},X,\varnothing} \mathbf{0}$ for any variable X. The proof follows the lines of [12].

By using standard ideas [19, 1], it would not be difficult to define axioms for static operations like hiding or parallel composition. Some operators have already been axiomatised in [6]. In particular, the following expansion law for parallel composition can be proved sound for the operational rules given in [10] ($\otimes$ is a 4-ary operation that returns a formula):

$$\begin{aligned} t \parallel_B^{\otimes} t' = & \sum_{i\in I, a_i\notin B} \delta_i : \gamma_i \to a_i(\mathbf{x}_i).(t_i \parallel_B^{\otimes} t') \;+ \sum_{j\in J, b_j\notin B} \delta'_j : \gamma'_j \to b_j(\mathbf{y}_j).(t \parallel_B^{\otimes} t'_j) \\ & + \sum_{\substack{i\in I, j\in J,\\ a_i = b_j \in B}} ((\delta_i, \gamma_i) \otimes (\delta'_j, \gamma'_j)) : ((\gamma_i \wedge \gamma'_j) \to a_i(\mathbf{x}_i\mathbf{y}_j).(t_i \parallel_B^{\otimes} t'_j) \end{aligned}$$

where $t = \sum_{i\in I} \delta_i : \gamma_i \to a_i(\mathbf{x}_i).t_i$ and $t' = \sum_{j\in J} \delta'_j : \gamma'_j \to b_j(\mathbf{y}_j).t'_j$.

References

1. L. Aceto, B. Bloom, and F.W. Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111(1):1–52, May 1994.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J.Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.
3. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
4. J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. EATCS Monographs. Springer, 2002.
5. G. Behrmann, A. David, K.G. Larsen, O. Möller, P. Pettersson, and Wang Yi. UPPAAL – present and future. In *Proceedings of 40*th *IEEE Conference on Decision and Control*. IEEE Press, 2001.
6. S. Bornot and J. Sifakis. An algebraic framework for urgency. *Information and Computation*, 163:172–202, 2000.
7. Howard Bowman. Modelling timeouts without timelocks. In *Proceeding of ARTS'99*, LNCS, page 20. Springer-Verlag, 1999.
8. M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. KRONOS: A model-checking tool for real-time systems. In A.J. Hu and M. Vardi, editors, *Proceedings of the 10th CAV*, volume 1427 of *LNCS*, pages 546–550. Springer, 1998.
9. P.R. D'Argenio and E. Brinksma. A calculus for timed automata (Extended abstract). In B. Jonsson and J. Parrow, editors, *Procs. of FTRTFT'96*, Uppsala, Sweden, volume 1135 of *LNCS*, pages 110–129. Springer, 1996.
10. P.R. D'Argenio and B. Gebremichael. The coarsest congruence for timed automata with deadlines contained in bisimulation. In *16th International Conference on Concurrency Theory (CONCUR05)*, volume 3653 of *LNCS*, pages 125–140, San francisco, USA, August 2005.
11. C. J. Fidge and J. J. Zic. An expressive real-time CCS. In *Second Australasian Conference on Parallel and Real-Time Systems (PART'95)*. Fremantle, September 1995.
12. R.J. van Glabbeek. A complete axiomatization for branching bisimulation congruence of finite-state behaviours. In *Proc. MFCS'93*, volume 711 of *LNCS*, pages 473–484. Springer, 1993.
13. Gregor Gössler and Joseph Sifakis. Composition for component-based modeling. *Sci. Comput. Program.*, 55(1-3):161–183, 2005.
14. George T. Heineman and William T. Councill. *Component Based Software Engineering*. Addison-Wesley, 2001.
15. T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
16. Wojtek Kozaczynski and Grady Booch. Guest editors' introduction: Component-based software engineering. *IEEE Software*, 15(5):34–36, 1998.
17. Huimin Lin and Wang Yi. Axiomatizing timed automata. *Acta Informatica*, 38(4):277–305, 2002.
18. R. Milner. A complete inference system for a class of regular behaviours. *J. of Comp. and System Sci.*, 28:439–466, 1984.
19. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
20. R. Milner. A complete axiomatisation for observational congruence of finite-state behaviours. *Information and Computation*, 81(2):227–247, 1989.
21. R. Segala, R. Gawlick, J.F. Søgaard-Andersen, and N.A. Lynch. Liveness in timed and untimed systems. *Information and Computation*, 141(2):119–171, 1998.

22. J. Sifakis and S. Yovine. Compositional specification of timed systems. In *Proceedings of the 13th Annual Symp. on Theoretical Aspects of Computer Science, STACS'96*, volume 1046 of *LNCS*, pages 347–359, Grenoble, France, 1996. Springer.

A Technical details of Section 4

Proof. (of Lemma 1.1)

$$(\phi \Rightarrow \gamma) \Rightarrow \phi \wedge \neg\gamma = \mathbf{ff} \tag{13}$$

$\Rightarrow$ {ABSURD and (13)}

$$\phi \wedge \neg\gamma \vdash \mathbf{0} = u. \tag{14}$$

$\Rightarrow$ {Applying PARTITION and the hypothesis $\phi \vdash t = u$}

$$\phi \wedge \gamma \vdash t = u \tag{15}$$

$\Rightarrow$ {Applying GUARD on (14), (15) }

$$\phi \vdash \gamma \rightarrow t = u.$$

Proof. (of Lemma 1.2)

$$(\phi \Rightarrow \neg\delta) \Rightarrow \phi \wedge \delta = \mathbf{ff} \tag{16}$$

$\Rightarrow$ {ABSURD and (16)}

$$\phi \wedge \delta \vdash \mathbf{tt} : t = u. \tag{17}$$

$\Rightarrow$ {Applying PARTITION and the hypothesis $\phi \vdash t = u$}

$$\phi \wedge \neg\delta \vdash t = u \tag{18}$$

$\Rightarrow$ {Applying DEADLINE on (17), (18) }

$$\phi \vdash \delta : t = u.$$

Proof. (of Lemma 1.3)

$$\phi \Rightarrow (\gamma \wedge \neg\delta) \text{ and } \phi \vdash t = u \tag{19}$$

$\Rightarrow \phi \Rightarrow \gamma$ and $\phi \vdash t = u$

$\Rightarrow$ { By Lemma 1.1}

$\phi \Rightarrow \neg\delta$ and $\phi \vdash \gamma \rightarrow t = u$

$\Rightarrow$ { By Lemma1.2}

$\phi \vdash \delta : \gamma \rightarrow t = u$

$\wedge$ { Similarly by (19)}

$\Rightarrow \phi \Rightarrow \neg\delta$ and $\phi \vdash t = u$

$\Rightarrow$ { By Lemma 1.2}

$\phi \Rightarrow \gamma$ and $\phi \vdash \delta : t = u$

$\Rightarrow$ { By Lemma 1.1}

$\phi \vdash \gamma \rightarrow \delta : t = u$

Proof. (of Lemma 1.4)

$\Rightarrow$ { by Lemma 1.1 and since $\phi \Rightarrow \phi$}

$\phi \vdash \phi \rightarrow t = t$

$\Rightarrow$ { By SUBSTITUTION, S2 and EQUIV}

$\phi \vdash t = t + \phi \rightarrow t$ (20)

$\wedge$ { Since $\neg\phi \wedge \phi = \mathbf{ff}$ and by ABSURD}

$\neg\phi \wedge \phi \vdash t = \mathbf{0}$ (21)

$\Rightarrow$ { By EQUIV}

$\neg\phi \wedge \neg\phi \vdash \mathbf{0} = \mathbf{0}$ (22)

$\Rightarrow$ { Applying GUARD on (21) and (22)}

$\neg\phi \vdash \phi \rightarrow t = \mathbf{0}$ (23)

$\Rightarrow$ { By SUBSTITUTION, S1 and EQUIV}

$\phi \vdash t = t + \phi \rightarrow t$ (24)

$\Rightarrow$ { applying PARTITION on (20) and (24), and EQUIV}

$t = t + \phi \rightarrow t$

Proof. (of Lemma 1.5)

$\Rightarrow$ { by Prop 4.1 [17]}

$\gamma_1 \rightarrow \gamma_2 \rightarrow t = (\gamma_1 \wedge \gamma_2) \rightarrow t$

Proof. (of Lemma 1.6)

$\Rightarrow$ { by (23)}

$\neg\gamma \vdash \gamma \rightarrow t_1 = \mathbf{0}$ and $\neg\gamma \vdash \gamma \rightarrow t_2 = \mathbf{0}$

$\Rightarrow$ { By S1 and EQUIV}

$\neg\gamma \vdash \mathbf{0} = \gamma \rightarrow t_1 + \mathbf{0}$ (25)

AND { By SUBSTITUTION}

$\neg\gamma \vdash \gamma \rightarrow t_1 + \mathbf{0} = \gamma \rightarrow t_1 + \gamma \rightarrow t_2$ (26)

$\Rightarrow$ { by EQUIV on (25) and (26)}

$\neg\gamma \vdash \mathbf{0} = \gamma \rightarrow t_1 + \gamma \rightarrow t_2$ (27)

AND {By Lemma 1.1}

$\gamma \vdash \gamma \rightarrow t_1 = t_1$ and $\gamma \vdash \gamma \rightarrow t_2 = t_2$

$\Rightarrow$ { By SUBSTITUTION}

$\gamma \vdash \gamma \rightarrow t_1 + t_2 = t_1 + t_2$ and $\gamma \vdash \gamma \rightarrow t_1 + \gamma \rightarrow t_2 = \gamma \rightarrow t_1 + t_2$

$\Rightarrow$ { By EQUIV}

$\gamma \vdash t_1 + t_2 = \gamma \rightarrow t_1 + \gamma \rightarrow t_2$ (28)

$\Rightarrow$ { Applying GUARD on (27) and (28)}

$\gamma \rightarrow (t_1 + t_2) = \gamma \rightarrow t_1 + \gamma \rightarrow t_2$ (29)

Proof. (of Lemma 1.7)

$\Rightarrow$ { by Lemma 1.4 and since $\phi \Rightarrow \phi$}

$\mathbf{tt} \vdash t = t + \gamma_2 \rightarrow t$

$\Rightarrow$ { since $\gamma_1 \Rightarrow \mathbf{tt}$, by PARTITION}

$\gamma_1 \vdash t = t + \gamma_2 \rightarrow t$ (30)

AND { By Lemma 1.1 and since $\gamma_1 \Rightarrow \gamma_1$}

$\gamma_1 \vdash t = \gamma_1 \rightarrow t$

$\Rightarrow$ { by SUBSTITUTION }

$\gamma_1 \vdash t + \gamma_2 \rightarrow t = \gamma_1 \rightarrow t + \gamma_2 \rightarrow t$ (31)

$\Rightarrow$ { by EQUIV on (30) and (31)}

$\gamma_1 \vdash t = \gamma_1 \rightarrow t + \gamma_2 \rightarrow t$ (32)

AND {Applying the same procedure on γ_2 }

$\gamma_2 \vdash t = \gamma_1 \rightarrow t + \gamma_2 \rightarrow t$ (33)

$\Rightarrow$ { By PARTITION on (32) and (33)}

$(\gamma_1 \vee \gamma_2) = \gamma_1 + t + \gamma_2 \rightarrow t$

Proof. (of Lemma 1.8)

$\Rightarrow$ { by EQUIV and PARTITION}

$\neg(\delta_1 \vee \delta_2) \vdash t = t$

$\Rightarrow$ { By Lemma 1.2 and logics}

$(\neg\delta_1 \wedge \neg\delta_2) \vdash t = (\delta_1 \vee \delta_2) : t$ (34)

AND { since $((\neg\delta_1 \wedge \delta_2) \wedge \neg(\delta_1 \vee \delta_2)) =$ **ff** and by ABSURD}

$((\neg\delta_1 \wedge \delta_2) \wedge \neg(\delta_1 \vee \delta_2)) \vdash \mathbf{tt} : t = t$ (35)

AND { by EQUIV }

$((\neg\delta_1 \wedge \delta_2) \wedge \neg(\delta_1 \vee \delta_2)) \vdash \mathbf{tt} : t = \mathbf{tt} : t$ (36)

$\Rightarrow$ { Applying DEADLINE on (35) and (36)}

$\neg\delta_1 \wedge \delta_2 \vdash \mathbf{tt} : t = (\delta_1 \vee \delta_2) : t$ (37)

$\Rightarrow$ { Applying DEADLINE on (34) and (37)}

$\neg\delta_1 \vdash \delta_2 : t = (\delta_1 \vee \delta_2) : t$ (38)

AND {since $(\delta_1 \wedge \neg(\delta_1 \vee \delta_2)) =$ **ff** then by ABSURD }

$(\delta_1 \wedge \neg(\delta_1 \vee \delta_2)) \vdash \mathbf{tt} : \delta_2 : t = t$ (39)

AND { by EQUIV and PARTITION}

$(\delta_1 \wedge \neg\delta_2) \vdash \mathbf{tt} : t = \mathbf{tt} : t$ (40)

AND {By D1, EQUIV and PARTITION }

$(\delta_1 \wedge \delta_2) \vdash \mathbf{tt} : \mathbf{tt} : t = \mathbf{tt} : t$ (41)

$\Rightarrow$ { Applying DEADLINE on (40) and (41)}

$\delta_1 \wedge (\delta_1 \vee \delta_2) \vdash \delta_2 : \mathbf{tt} : t = \mathbf{tt} : t$

$\Rightarrow$ { by D2}

$\delta_1 \wedge (\delta_1 \vee \delta_2) \vdash \mathbf{tt} : \delta_2 : t = \mathbf{tt} : t$ (42)

$\Rightarrow$ { Applying DEADLINE on (39) and (42) and EQUIV}

$\delta_1 \vdash \mathbf{tt} : \delta_2 : t = (\delta_1 \vee \delta_2) : t$ (43)

$\Rightarrow$ { Applying DEADLINE on (38) and (43) and EQUIV}

$\mathbf{tt} \vdash \delta_1 : \delta_2 : t = (\delta_1 \vee \delta_2) : t$

Proof. (of Lemma 1.9) First we prove the following small lemma

If $\delta \Rightarrow \phi$ then $\phi \vdash \delta : t = \mathbf{tt} : t$ (44)

$(\delta \Rightarrow \phi) \Rightarrow ((\phi \wedge \neg\delta) = \mathbf{ff})$

$\Rightarrow$ { By ABSURD}

$(\phi \wedge \neg\delta) \vdash t = \mathbf{tt} : t$ (45)

AND { By EQUIV}

$(\phi \wedge \delta) \vdash \mathbf{tt} : t = \mathbf{tt} : t$ (46)

$\Rightarrow$ { Applying DEADLINE on (45) and (46) }

$\phi \vdash \delta : t = \mathbf{tt} : t$

Now the proof of Lemma 1.9 follows

$\Rightarrow$ { by Lemma 1.2}

$\neg\delta \vdash t_1 = \delta : t_1$

$\Rightarrow$ { By SUBSTITUTION}

$\neg\delta \vdash t_1 + t_2 = \delta : t_1 + t_2$ (47)

AND { By Lemma 1.2 and SUBSTITUTION as above}

$\neg\delta \vdash \delta : t_1 + t_2 = \delta : t_1 + \delta : t_2$ (48)

$\Rightarrow$ { By EQUIV of(47) and (48) }

$\neg\delta \vdash t_1 + t_2 = \delta : t_1 + \delta : t_2$ (49)

AND { by (44) and EQUIV}

$\delta \vdash \mathbf{tt} : t_1 = \delta : t_1$

$\Rightarrow$ { Applying SUBSTITUTION}

$\delta \vdash \mathbf{tt} : t_1 + \mathbf{tt} : t_2 = \delta : t_1 + \mathbf{tt} : t_2$ (50)

$\Rightarrow$ { Again (44) and EQUIV}

$\delta \vdash \mathbf{tt} : t_2 = \delta : t_2$

$\Rightarrow$ { Applying SUBSTITUTION}

$\delta \vdash \delta : t_1 + \mathbf{tt} : t_2 = \delta : t_1 + \delta : t_2$ (51)

$\Rightarrow$ { by EQUIV of (50) and (51)}

$\delta \vdash \mathbf{tt} : t_1 + \mathbf{tt} : t_2 = \delta : t_1 + \delta : t_2$ (52)

$\Rightarrow$ {By D4 }

$\delta \vdash \mathbf{tt} : (t_1 + t_2) = \delta : t_1 + \delta : t_2$ (53)

$\Rightarrow$ { Applying DEADLINE on (49) and (53)}

$\mathbf{tt} : (t_1 + t_2) = \delta : t_1 + \delta : t_2$

Proof. (of Lemma 1.16)

$\Rightarrow$ { By Lemma 1.15}

$\mathbf{ff} \rightarrow t = \mathbf{0}$

$\Rightarrow$ { by SUBSTITUTION }

$\mathbf{tt} : \mathbf{ff} \rightarrow t = \mathbf{tt} : \mathbf{0}$

$\Rightarrow$ { by U3 and EQUIV}

$\mathbf{ff} \rightarrow \mathbf{tt} : t = \mathbf{tt} : \mathbf{0}$

$\Rightarrow$ { By Lemma 1.15 and EQUIV}

$\mathbf{0} = \mathbf{tt} : \mathbf{0}$