# Correctness Proof of the Standardized Algorithm for ABR Conformance

Jean-François Monin and Francis Klay

France Télécom CNET DTL/MSV, Technopôle Anticipa
2 av. Pierre Marzin, 22307 Lannion, France
JeanFrancois.Monin@cnet.francetelecom.fr

**Abstract.** Conformance control for ATM cells is based on a real-time reactive algorithm which delivers a value depending on inputs from the network. This value must always agree with a well defined theoretical value. We present here the correctness proof of the algorithm standardized for the ATM transfer capability called ABR. The proof turned out a key argument during the standardization process of ABR.

## 1 Introduction

We want to present in this paper an unusual (at least to our knowledge) application of formal methods in telecommunications, though it is closely related to a protocol. There is now quite a long tradition in using formal languages in this area, even standardized ones. They are based on communicating (extended) finite state machines (e.g. Estelle, SDL, Promela) or process algebra (e.g. Lotos). Verification based on model checking [16, 8] or simulation [14] has also been successfully employed. Typically, you model the protocol at hand, using one of the above formalisms, and then you try to verify that bad things like unexpected messages, deadlocks and so on never happen. To this effect you may use temporal logic formulas or observers and automated verification tools. This approach turns out very useful because the global behavior of a system made of several concurrent components is difficult to grasp.

In the problem we deal with here, complexity does not lie in parallelism or message interleaving, but in a single sequential, short, real-time and reactive algorithm. This algorithms runs on a key device for an ATM Transfer Capability called ABR (see below). It handles a small scheduler and delivers a value which depends on inputs from the network. We essentially have to prove that the value delivered by the device always agrees with (more precisely: is not smaller than) a theoretical value whose computation is not feasible under realistic assumptions. The correctness proof presented here has been a key argument in the standardization process of ABR.

The technique used is basically the calculus of weakest preconditions. However real time comes into the picture: not only the mathematical expression of the problem involves functions of the time, but invariants themselves involve such functions: a scheduler predicts values in the future. In order to make the result as convincing as possible (a must in the context of standardization) we do not hesitate to make proof steps explicit and before anything we start from a very simple and declarative specification $\mathcal{S}_\mathrm{d}$. At a later stage, we describe the state space of the device under study, as well as associated invariants. Unfortunately specification $\mathcal{S}_\mathrm{d}$ is technically not well suited to the correctness proof. A bit of theory has to be developed, in order to get an equivalent but more tractable computational specification $\mathcal{S}_\mathrm{c}$. The invariant to be proved is then stated in terms of $\mathcal{S}_\mathrm{c}$ and the proof can be carried out thanks to preliminary lemmas related to $\mathcal{S}_\mathrm{c}$. This paper aims at giving the details of this work. Note that the specification and the whole proof have been completely formalized with Coq [5], an automated proof assistant based on type theory [15].

The rest of this paper is organized as follows. Section 2 describes the problem as well as the stakes for telecommunications. Section 3 states the specifications called $\mathcal{S}_\mathrm{d}$ and $\mathcal{S}_\mathrm{c}$ above and explains how to get the latter from the former. Section 4 describes the state space with its invariant and sketches the main steps of the proof (the standardized algorithm and technical details of its correctness proof are respectively given in appendix A and B). We end with concluding remarks and related work in section 5.

## 2 Context and Motivation

### 2.1 Conformance Control in ATM

In an ATM (Asynchronous Transfer Mode) network, data packets (cells) sent by a user must not exceed a rate which is defined by a contract negotiated between the user and the network. Several modes for using an ATM network, called "ATM Transfer Capabilities" (ATCs) have been defined. Each ATC may be seen as a generic contract between the user and the network, saying that the network must guarantee the negotiated *quality of service* (QoS), defined by a number of characteristics like maximum cell loss or transfer delay, provided the cells sent by the user conform to the negotiated traffic parameters (for instance, their rate must be bounded by some value). The conformance of cells sent by the user is checked using an algorithm called GCRA (generic control of cell rate algorithm). In this way, the network is protected against users misbehaviors and keeps enough resources for delivering the required QoS to well behaved users.

In fact, a new ATC cannot be accepted (as an international standard) without an efficient conformance control algorithm, and some evidence that this algorithm has the intended behavior.

For the ATC called ABR (Available Bit Rate), considered here, a simple but inefficient algorithm had been proposed in a first stage. Reasonably efficient algorithms proposed later turned out to be fairly complicated. This situation has been settled when one of them, due to Christophe Rabadan, has been proved correct in relation to the simple one: this algorithm is now part of the I.371.1 standard [13]. The first version of the proof was hand written. The main invariants discovered during this process are included in I.371.1. Later on, the proof has been completely formalized and mechanically verified with Coq [5].

### 2.2 The Case of ABR

In some of the most recently defined ATCs, like ABR, the allowed cell rate (ACR) may vary during the same session, depending on the current congestion state of the network. Such ATCs are designed for irregular sources, that need high cell rates from time to time, but that may reduce their cell rate when the network is busy. A servo-mechanism is then proposed in order to let the user know whether he can send data or not. This mechanism has to be well defined, in order to have a clear traffic contract between user and network. The key is an adaptation of the public algorithm for checking conformance of cells.

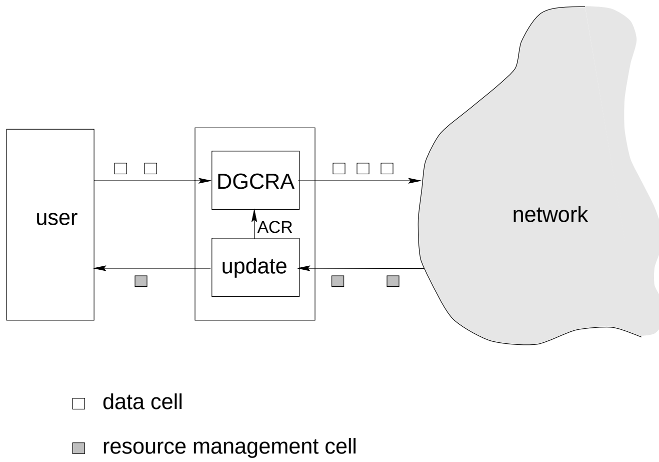


**Fig. 1.** conformance control

An abstract view of the protocol ABR is given in fig. 1 (actually, resource management (RM) cells are sent by the user, but only their transmission from the network to the user is relevant here; details are available in [17]). The conformance control algorithm for ABR has two parts. The first one is called DGCRA (dynamic GCRA). It just checks that the rate of data cells emitted by the user is not higher than a value which is approximately Acr, the allowed cell rate. Excess cells may be discarded by DGCRA. Note that, in the case of ABR, Acr depends on time: its value has to be known each time a new data cell comes from the user. This part is quite simple and is not addressed here. The complexity lies in the computation of $\mathrm{Acr}(t)$ ("update" in fig. 1), which depends on the sequence of values $(\mathrm{ER}_n)$ carried by resource management cells coming from the network. By a slight abuse of notation, the cell carrying $\mathrm{ER}_n$ will be called itself $\mathrm{ER}_n$.

Of course, $\mathrm{Acr}(t)$ depends only on cells $\mathrm{ER}_n$ whose arrival time $t_n$ is such that $t_n < t$ (we order resource management cells so that $t_n < t_{n+1}$ for any $n$). In ABR, a resource management cell carries a value of Acr, that should be reached as soon as possible. At first sight, $\mathrm{Acr}(t)$ should then be simply the last $\mathrm{ER}_i$ received at time $t$, i.e. $\mathrm{ER}_i$ with $i = \mathrm{last}(t)$, where $\mathrm{last}(x)$ is the only integer such that $t_i \leq x < t_{i+1}$. Unfortunately, because of electric propagation time and various transmission mechanisms, the user is aware of this expected value only after a delay. Taking the user's reaction time $\tau$ observed by the control device into consideration, that is, the overall round trip time between the control device and the user, $\mathrm{Acr}(t)$ should then be $\mathrm{ER}_i$ with $i = \mathrm{last}(t - \tau)$. But $\tau$ may vary in turn. ITU-T considers that a lower bound $\tau_3$ and an upper bound $\tau_2$ for $\tau$ are established during the negotiation phase of each ABR connection. Hence, a cell arriving from the user at time $t$ on DGCRA may legitimately have been emitted using any rate $\mathrm{ER}_i$ such that $i$ is between $\mathrm{last}(t - \tau_2)$ and $\mathrm{last}(t - \tau_3)$. Any rate less than or equal to any of these values, or, equivalently less than or equal to the maximum of them, should then be allowed. Therefore, $\mathrm{Acr}(t)$ is taken as the maximum of these $\mathrm{ER}_i$.

Actually the standards committee did not give these explanations, but directly specified the set of $\mathrm{ER}_i$ under the equivalent form (2) below.

### 2.3 Effective Computation of Acr($t$)

ITU-T committee considered that a direct computation of $\mathrm{Acr}(t)$ is not feasible at reasonable cost with current technologies: it would amount to compute the maximum of several hundreds integers each time a cell is received from the user. However, it is not difficult to see that $\mathrm{Acr}(t)$ is constant on any interval that contains no value among $\{t_n + \tau \mid \tau = \tau_2 \vee \tau = \tau_3\}$. In other words, $\mathrm{Acr}(t)$ is determined by a sequence of values. It then becomes possible to use a scheduler handling future changes of $\mathrm{Acr}(t)$. This scheduler is updated when a new cell $\mathrm{ER}_n$ is received. Roughly, if $s$ is the current time, $\mathrm{ER}_n$ will be taken into account at time $s + \tau_3$, while $\mathrm{ER}_{n-1}$ will not be taken into account after $s + \tau_2$.

The control conformance algorithm considered here exploits this idea, with the further constraint that only a small amount of memory is allocated to the scheduler. This means that some information is lost. *Filtering is performed in such a way that the actual value of* $\mathrm{Acr}(t)$ *is greater or equal to its theoretical value, as defined above.*

## 3 Ideal ACR

### 3.1 Declarative Specification

The declarative specification ($\mathcal{S}_\mathrm{d}$ in the introduction) of the ideal value or Acr is given by (1) and (2) under assumption (3). We are given a sequence of RM cells ($\mathrm{ER}_i$) whose arrival date are respectively ($t_i$); the desired allowed cell rate at time $t$ is defined by :

$$\mathrm{Acr}(t) = \max\{\mathrm{ER}_i \mid i \in I(t)\}, \tag{1}$$

where $I$ is the interval defined by :

$$i \in I(t) \quad \text{iff} \quad (t - \tau_2 \ < \ t_i \ \leq \ t - \tau_3) \ \vee \ (t_i \ \leq \ t - \tau_2 \ < \ t_{i+1}) \ . \tag{2}$$

The $t_i$ are taken in increasing order :

$$t_1 < t_2 < \ldots t_n < \ldots \tag{3}$$

The following equivalent characterization of $I(t)$ is easier to handle:

$$i \in I(t) \quad \text{iff} \quad t_i + \tau_3 \ \leq \ t \ < \ t_{i+1} + \tau_2 \tag{4}$$

The initial (inefficient) ABR conformance control algorithm was a direct computation of Acr according to (1).

### 3.2 Specification Using only Finite Knowledge

In practice, only finite prefixes of the sequence $(t_i)$ are available. Then we have to take into account that, given a list $l$ of length $\sharp l$ of RM cells $(\mathrm{ER}_i)$ whose arrival date are respectively $(t_i)$, $t_{i+1}$ makes sense only for $i < \sharp l$. However, if we consider that $t_{\sharp l+1} = \infty$, (4) boils down to $\sharp l \in I(t)$ iff $t_{\sharp l} \ \leq \ t - \tau_3$. With this intuition in mind we introduce

$$\mathrm{Approx}(l,t) = \max\{\mathrm{ER}_i \mid i \in I_a(l,t)\}, \tag{5}$$

where $I_a(l,t)$ is similar to $I(t)$:

$$i \in I_a(l,t) \quad \text{iff} \quad \begin{cases} i \in I(t) & \wedge \quad i < \sharp l \\ \vee & \\ t_i \ \leq \ t - \tau_3 & \wedge \quad i = \sharp l \ . \end{cases} \tag{6}$$

The list $l$ we consider depends on time : $l(s)$ contains all indices $i$ such that $t_i \leq s$, where $s$ represents the current time. The following lemma, whose meaning is that it is enough to compute $\mathrm{Approx}(l(s),t)$, is easy to prove :

**Lemma 1.**
*(i) The value of* Approx *at $t$ becomes stable after $t - \tau_3$:*

$$\forall s \geq t - \tau 3, \quad \mathrm{Approx}(l(s),t) = \mathrm{Approx}(l(t - \tau_3),t) \ .$$

*(ii)* $\mathrm{Approx}(l(s),t)$ *is an exact approximation of* $\mathrm{Acr}(t)$ *for* $s \geq t - \tau 3$*:*

$$\forall t, \quad \mathrm{Acr}(t) = \mathrm{Approx}(l(t - \tau_3),t) \ .$$

Unless otherwise stated $s$ will remain implicit in the following : we will note just $l$ instead of $l(s)$. In the same way, variables like `Efi` (see below) actually handled by the algorithm denote a value that also depends on $s$, but will be noted just `Efi`. We also assume without loss of generality that $\mathrm{ER}_0$ is equal to the initial value of Acr; if the algorithm starts at $s_0$, this amounts to stating $t_0 = s_0 - \tau_3$ and for $i > 0$, $t_i > s_0$:

$$t_0 = s_0 - \tau_3 < s_0 < t_1 < t_2 < \ldots t_{\sharp l} \ . \tag{7}$$

Note that hereafter, $\sharp l$ is the number of the last element of $l$.

The will use the following explicit characterization of $I_a(l,t)$:

$$i \in I_a(l,t) \text{ iff } \begin{cases} t_i + \tau_3 \ \le \ t \ < \ t_{i+1} + \tau_2 & \wedge \quad i < \sharp l \\ \vee & \\ t_i + \tau_3 \ \le \ t & \wedge \quad i = \sharp l \ . \end{cases} \tag{8}$$

In particular, for $t' \ge t \ge t_{\sharp l} + \tau_3$, we have $i \in I_a(l,t')$ implies $i \in I_a(l,t)$, hence the following lemma :

**Lemma 2.** Approx$(l,t)$ *is decreasing after* $t_{\sharp l} + \tau_3$*:*

$$\forall t, t', t_{\sharp l} + \tau_3 \ \le \ t \le t' \Rightarrow \mathrm{Approx}(l,t') \le \mathrm{Approx}(l,t) \ .$$

It is also easy to see that for any $t$ greater than $t_{\sharp l} + \tau_2$, the only $i$ of $I_a(l,t)$ is $\sharp l$, hence the following lemma :

**Lemma 3.**

$$\forall t, \quad t_{\sharp l} + \tau_2 \ \le \ t \quad \Rightarrow \quad i \in I_a(l,t) \text{ iff } i = \sharp l, \tag{9}$$

$$\forall t, \quad t_{\sharp l} + \tau_2 \ \le \ t \quad \Rightarrow \quad \mathrm{Approx}(l,t) = \mathrm{ER}_{\sharp l}. \tag{10}$$

### 3.3 Computing Approx in an Incremental Way

Initially (at time $s_0 = t_0 + \tau_3$), we have $l = \langle t_0 \rangle$. The characterization (8) of $I_a$ yields then

$$i \in I_a(\langle t_0 \rangle, t) \quad \text{iff} \quad s_0 \le t \ \wedge \ i = 0 \ .$$

Then we get (as desired):

$$\forall t, \quad s_0 \le t \ \Rightarrow \ \mathrm{Approx}(\langle t_0 \rangle, t) = \mathrm{ER}_0 \ . \tag{11}$$

We now consider the adjunction of a new $t_n$ at the end of $l$. The new list is denoted by $l \frown t_n$, and we have $n = \sharp l + 1 = \sharp(l \frown t_n)$. Three cases have to be considered : $i < n-1$, $i = n-1$ and $i = n$. We can further assume that $t < t_n + \tau_2$, because lemma 3 gives us directly the result if $t_n + \tau_2 \le t$. The assumption $t < t_n + \tau_2$ is especially useful for $i = n-1$. Using (8) we see that

- for $i < n-1$, $\quad i \in I_a(l,t)$ iff $t_i + \tau_3 \ \le \ t \ < \ t_{i+1} + \tau_2$ iff $i \in I_a(l \frown t_n, t)$,
- for $t \ < \ t_n + \tau_2$, $\quad n-1 \in I_a(l,t)$ iff $t_{n-1} + \tau_3 \ \le \ t$ iff $n-1 \in I_a(l \frown t_n, t)$,
- $n \in I_a(l \frown t_n, t)$ iff $t_n + \tau_3 \ \le \ t$.

Hence we can verify that :

- for $t < t_n + \tau_3$, $\quad i \in I_a(l^\frown t_n, t)$ iff $i \in I_a(l, t)$,
- for $t_n + \tau_3 \le t < t_n + \tau_2$, $\quad i \in I_a(l^\frown t_n, t)$ iff $i \in I_a(l, t) \vee i = n$,
- for $t_n + \tau_2 \le t$, $\quad i \in I_a(l^\frown t_n, t)$ iff $i = n$ (by (9)).

Thus we get the following way of computing the value of $\mathrm{Approx}(l^\frown t_n, t)$ from the value of $\mathrm{Approx}(l, t)$ (this what we called $\mathcal{S}_c$ in the introduction):

**Lemma 4.** *The value of* $\mathrm{Approx}(l^\frown t_n, t)$ *is given by :*

| | $t < t_n + \tau_3$ | $t_n + \tau_3 \le t < t_n + \tau_2$ | $t_n + \tau_2 \le t$ |
|---|---|---|---|
| $\mathrm{Approx}(l^\frown t_n, t)$ | $\mathrm{Approx}(l, t)$ | $\max(\mathrm{Approx}(l, t), \mathrm{ER}_n)$ | $\mathrm{ER}_n$ |

.

A simple but useful consequence of lemma 4 is :

**Lemma 5.**

$$\forall\, l, t, t_n, \mathrm{ER}_n, M, \qquad \mathrm{Approx}(l, t) \le M \;\wedge\; \mathrm{ER}_n \le M \quad \Rightarrow \quad \mathrm{Approx}(l^\frown t_n, t) \le M. \qquad (12)$$

## 4 Proof of Algorithm B'

The enhanced algorithm B proposed by CNET (hereafter called B') computes an upper bound of Approx. More precisely, at the current instant $s$, the state $e(s)$ handled by B' defines a function $\mathrm{Ub}(e(s), t)$ greater than $\mathrm{Approx}(l(s), t)$ for any $t \ge s$ and not defined for $t < s$.

Running the algorithm consists of changing the state $e$ into a $e'$. Such a step is called a *transition*. Here we have essentially two kinds of transitions: the first is fired when receiving a new RM cell, the second is fired when the current time reaches the date for a scheduled event.

We basically use standard calculus of weakest preconditions [9] with notations taken from B [1]. Our treatment of time is inspired by timed automata of [3] and the synchrony hypothesis of synchronous languages [11]. We assume that our system reacts more quickly than its environment : state transitions induced by the arrival of a RM cell or due to the scheduler are finished before the arrival of a new RM cell. This assumption depends on the technology used in the real device and can be checked on it. It is then safe to consider that a transition takes no time. Timed automata consider two kinds of transitions: "continuous" ones concerning time evolution (modeled by clocks) and "discrete" ones concerning the state.

Here we just need to assume the existence of an external clock, with an internal value $s$ that can be read but not written by programming means. We model the progress of time by an implicit assignment

$$s := \text{current date},$$

for instance $s := t_k$ when the $k^{\text{th}}$ RM cell is received. The new value of $s$ cannot be smaller than its old value, and we also constrain the new value in a way such that no event arose in the meantime (we assume that the scheduler is reliable). Formally, we consider transitions of the form

$$\langle s, e\rangle \longrightarrow \langle s', e'\rangle$$

with $s \leq s'$ and such that nothing happened between $s$ and $s'$, and where $e'$ is the new state obtained from $s$ after running a transition of algorithm B'. This is made explicit in assumptions $(G_e)$ and $(G_i)$ below. It may happen that an internal event is scheduled at a time $t_k$. In that case, the internal event has to be handled first.

Transitions are modeled by program assignments or "generalized substitutions" in the terminology of B.

### 4.1 Components of the State

The state $e$ is made of 5 variables :

- `ACR`, the current ACR;
- `Efi`, the next ACR if nothing new happens;
- `tfi`, the date at which `Efi` will be active if nothing new happens;
- `Ela`, containing the value of the last known order ($\mathrm{ER}_{\sharp l}$);
- `tla`, the date at which `Ela` will be active if nothing new happens.

As an optimization trick, there is a sixth variable `Emx` whose value is just the maximum of `Efi` and `Ela`.

### 4.2 Transitions

The algorithm reacts either when receiving a new $\mathrm{ER}_n$, i.e. when the current time reaches $t_n$ (this is called an external event in the sequel), or when the current time reaches `tfi` (this is called an internal event in the sequel). Each transition changes the current state; an internal event is scheduled if and only if `tfi` is greater than the current time.

### 4.3 Invariant

Here we want to ensure that B' provides an `ACR` which cannot be less than the ideal value $\mathrm{Acr}(s)$. To this effect we prove that the following property is invariant. The current time is noted $s$.

$$\mathrm{Approx}(l, s) \leq \mathtt{ACR} \qquad (\mathrm{I_{main}})$$

$\mathrm{I_{main}}$ is itself a consequence of the following conjunction.

$$\texttt{Emx} = \max(\texttt{Efi}, \texttt{Ela}) \tag{$\mathrm{I_{max}}$}$$
$$\texttt{Ela} = \mathrm{ER}_{\sharp l} \tag{$\mathrm{I_{Ela}}$}$$
$$\texttt{tfi} \le \texttt{tla} \le t_{\sharp l} + \tau_2 \tag{$\mathrm{I_{fil}}$}$$
$$\texttt{tfi} \le s \;\Rightarrow\; \forall t,\; s \le t \;\Rightarrow\; \mathrm{Approx}(l,t) \le \texttt{ACR} \tag{$\mathrm{I_{tfs}}$}$$
$$\texttt{ACR} < \texttt{Efi} \;\Rightarrow\; \texttt{tfi} \le t_{\sharp l} + \tau_3 \tag{$\mathrm{I_{Et1}}$}$$
$$\texttt{Efi} < \texttt{Ela} \;\Rightarrow\; \texttt{tla} \le t_{\sharp l} + \tau_3 \tag{$\mathrm{I_{Et2}}$}$$
$$\texttt{tfi} = \texttt{tla} \;\Rightarrow\; \texttt{Efi} = \texttt{Ela} \tag{$\mathrm{I_{ttE}}$}$$
$$\forall t \quad s \le t < \texttt{tfi} \;\Rightarrow\; \mathrm{Approx}(l,t) \le \texttt{ACR} \tag{$\mathrm{I_{Ub1}}$}$$
$$\forall t \quad \texttt{tfi} \le t < \texttt{tla} \;\Rightarrow\; \mathrm{Approx}(l,t) \le \texttt{Efi} \tag{$\mathrm{I_{Ub2}}$}$$
$$\forall t \quad \texttt{tla} \le t \;\Rightarrow\; \mathrm{Approx}(l,t) \le \texttt{Ela}\ . \tag{$\mathrm{I_{Ub3}}$}$$

We define

$$\mathrm{Inv} = \mathrm{I_{max}} \wedge \mathrm{I_{Ela}} \wedge \mathrm{I_{fil}} \wedge \mathrm{I_{tfs}} \wedge \mathrm{I_{Et1}} \wedge \mathrm{I_{Et2}} \wedge \mathrm{I_{ttE}} \wedge \mathrm{I_{Ub1}} \wedge \mathrm{I_{Ub2}} \wedge \mathrm{I_{Ub3}}.$$

Invariants $\mathrm{I_{Ub1}}$, $\mathrm{I_{Ub2}}$ and $\mathrm{I_{Ub3}}$ mean that $\mathrm{Approx}(l,t) \le \mathrm{Ub}(e,t)$ for $t \ge s$, where the function $\mathrm{Ub}(e,t)$ is defined by: $\mathrm{Ub}(e,t) = \texttt{ACR}$ for $s \le t < \texttt{tfi}$, $\mathrm{Ub}(e,t) = \texttt{Efi}$ for $\texttt{tfi} \le t < \texttt{tla}$, $\mathrm{Ub}(e,t) = \texttt{Ela}$ for $\texttt{tla} \le t$. In the sequel we use the following consequence of $\mathrm{I_{max}}$, $\mathrm{I_{Ub2}}$ and $\mathrm{I_{Ub3}}$:

$$\forall t \quad \texttt{tfi} \le t \;\Rightarrow\; \mathrm{Approx}(l,t) \le \texttt{Emx}\ . \tag{$\mathrm{I_{Apx}}$}$$

**Lemma 6.** Inv *implies* $\mathrm{I_{main}}$.

*Proof.* We have either $\texttt{tfi} \le s$ or $s < \texttt{tfi}$. Apply respectively $\mathrm{I_{tfs}}$ and $\mathrm{I_{Ub1}}$ with $t = s$.

### 4.4 Initial State

Initially we have $l = \langle t_0 \rangle$, $\sharp l = 0$ and we want to show that Inv is true in the initial state defined by:

$$\texttt{tfi} = \texttt{tla} = s_0, \texttt{ACR} = \texttt{Emx} = \texttt{Efi} = \texttt{Ela} = \mathrm{ER}_0 \quad \text{(initial value of Acr).}$$

Formally, we consider the substitution $\mathsf{S}_0$:

$$\mathsf{S}_0 \;\stackrel{\mathrm{df}}{=}\; l, \texttt{ACR}, \texttt{Emx}, \texttt{Efi}, \texttt{Ela}, \texttt{tfi}, \texttt{tla} := \langle t_0 \rangle, \mathrm{ER}_0, \mathrm{ER}_0, \mathrm{ER}_0, \mathrm{ER}_0, s_0, s_0$$

and we show $[\mathsf{S}_0]$Inv. The proof is very easy.

### 4.5 External Event

Let $s$ be the current time. Let $k$ be an abbreviation for $\sharp l + 1$. We consider a transition from $s$ to $s' = t_k$ (and consistently of $e(s)$ to $e(t_k)$) only if $s \le t_k$

and there is no internal event between $s$ and $t_k$. Formally, the following guard is taken for granted:

$$s \leq t_k \ \wedge \ (\texttt{tfi} \leq s \vee t_k < \texttt{tfi}). \tag{$G_e$}$$

At time $t_k$, the list $l$ becomes then $l \frown t_k$. Formally, the following substitution is always taken into account:

$$\mathsf{T_e} \ \stackrel{\text{df}}{=} \ s, l := t_k, l \frown t_k.$$

The complete pseudo-code and the proof are given is appendixes A and B.

### 4.6 Internal Event

Let $s$ be the current time. We consider a transition from $s$ to $s' = \texttt{tfi}$ (and consistently of $e(s)$ to $e(\texttt{tfi})$) only if $s < \texttt{tfi}$ and there is no external event between $s$ and $\texttt{tfi}$. Formally, the following guard is taken for granted:

$$t_{\sharp l} \leq s \leq \texttt{tfi} \leq t_{\sharp l+1}. \tag{$G_i$}$$

The substitution $\mathsf{T_i} \ \stackrel{\text{df}}{=} \ s := \texttt{tfi}$. is also taken into account. Let $\mathsf{S_i}$ be the substitution

$$\mathsf{S_i} \ \stackrel{\text{df}}{=} \ \mathsf{T_i} \ || \ \texttt{ACR}, \texttt{tfi}, \texttt{Efi}, \texttt{Emx} := \texttt{Efi}, \texttt{tla}, \texttt{Ela}, \texttt{Ela}.$$

In appendix B we show: $\mathrm{Inv} \wedge (G_i) \Rightarrow [\mathsf{S_i}]\mathrm{Inv}$.

### 4.7 Observing Intermediate States

Let $s$ be the current time. We consider a transition from $\langle s, e \rangle$ to $\langle s', e \rangle$ only if $s \leq s'$ and there is no event between $s$ and $s'$. Actually $s'$ may be equal to $t_k$ but must remain less than $\texttt{tfi}$ (when real time reaches $\texttt{tfi}$, $\mathsf{S_i}$ must run). Formally, the following guard is taken for granted:

$$\begin{array}{l} s \leq s' \ \wedge \ (\texttt{tfi} \leq s \vee s' < \texttt{tfi}) \ \wedge \\ (\forall i, \ t_i \leq s \vee s' \leq t_i) \ \wedge \ s = t_{\sharp l+1} \ \Rightarrow \ s' = t_{\sharp l+1}. \end{array} \tag{$G_o$}$$

The transition is modeled by the substitution: $\mathsf{T_o} \ \stackrel{\text{df}}{=} \ s := s'$. In appendix B we show $\mathrm{Inv} \wedge (G_o) \Rightarrow [\mathsf{T_o}]\mathrm{Inv}$.

### 4.8 Main theorem

Our main result is an easy consequence of previous lemmas.

**Theorem 1.** *At any time $s$ we have* $\mathrm{Acr}(s) \leq \texttt{ACR}$.

*Proof.* Using lemma 1 we know that $\mathrm{Acr}(s) = \mathrm{Approx}(l, s)$. As Inv is actually an invariant, lemma 6 yields $\mathrm{Approx}(l, s) \leq \texttt{ACR}$, hence the result.

## 5 Discussion and Related Work

For engineers working in the context of standardization, theorem 1 is much more convincing than the similar theorem involving Approx instead of Acr. However it is clear for us that the computational characterization of Approx (lemma 4) is much more suited for reasoning about B'. In a first attempt, we tried to prove directly the invariant Inv using (1) and (2). This resulted in shallow areas and even holes in the manual proof.

We also submitted the problem of the correctness of B' to other research teams, in order to assess other approaches. It is too early (and beyond the scope of this paper) to compare the results of these works, we just give some hints. Model checking using classical and temporal automata is experimented in the framework of FORMA (`http://www-verimag.imag.fr/FORMA`), a project founded by the French government which aims at experimenting various formal methods on industrial case studies. In the two first attempts, the property to be checked corresponded to theorem 1, but modeling Acr contributed to an explosion of the number of states. Moreover the tools used—UPPAAL [6] and MEC [4]—allowed only fixed numeric values for $\tau_2$, $\tau_3$ and $\mathrm{ER}_i$. Checking the algorithm could be carried through for small values. Later on, good results within two different frameworks have been obtained by L. Fribourg [10] and B. Bérard [7], with specifications based on Approx instead of Acr. In one framework, they used the parameterized temporized automata of Hytech [12], and in the other an automated proof search procedure due to Revesz [18] was extended to timed automata. In both cases $\tau_2$, $\tau_3$, etc. were symbolic parameters and the desired property could be checked without the help of Inv. In our case, Inv has been incrementally constructed while attempting to prove $\mathrm{I}_{\mathrm{tfs}}$ and $\mathrm{I}_{\mathrm{Ub1}}$, following the steps given in appendix B. Note that such calculations are boring and error prone: this is why we felt that the proof should be checked with a proof assistant. Indeed, our experiment with Coq [15] showed that one of the proofs of appendix B was wrong (but could be repaired, fortunately !). A detailed comparison between the approaches mentioned above will be done in a forthcoming paper.

Finally, let us say a word on two attempts using B. At CNET we (with G. Blorec) tried to use this method on this case study two years ago. At first sight B should be well suited, because of our systematic use of substitution calculus. But we failed to handle time and the very notion of scheduler in a nice way; our specification was heavy and many proof obligations could not be discharged. Recently, Abrial worked on this problem using an event oriented variant of B and he succeeded to reconstruct an algorithm different from the one standardized in I.371, but where the design decisions are much clearer [2].

Our current feeling is that specialized procedures or methods can discharge boring and painful parts in the verification process, but are really successful only on "predigested" specifications like $\mathcal{S}_{\mathrm{c}}$, in contrast with $\mathcal{S}_{\mathrm{d}}$. On the other side, general purpose frameworks and tools like type theory and Coq are helpful on the whole process but still require much more interaction from the user on the parts automatically handled by specialized methods. Work is in progress for integrating both kind of techniques in the same tool.

## Acknowledgement

The problem has been submitted by Christophe Rabadan, who is the main author of the algorithm studied in this document. This work has benefited of very fruitful discussions with him and Annie Gravey. Many improvements are also due to the comments of anonymous referees.

## References

1. J.-R. Abrial. *The B-Book: Assigning Programs to Meanings.* Cambridge University Press, 1996.
2. J.-R. Abrial. Développement de l'algorithme ABR. Personal communication, 1999.
3. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.
4. A. Arnold. MEC: A system for constructing and analysing transition systems. In J. Sifakis, editor, *Proceedings of the International Workshop on Automatic Verification Methods for Finite State Systems*, volume 407 of *LNCS*, pages 117–132, Berlin, June 1990. Springer.
5. Bruno Barras and all. The coq proof assistant reference manual : Version 6.1. Technical Report RT-0203, INRIA, 1997.
6. J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and Wang Yi. UPPAAL: a tool suite for the automatic verification of real-time systems. In R. Alur, T. A. Henzinger, and E. D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 232–243. Springer-Verlag, 1996.
7. B. Bérard and L. Fribourg. Automated verification of a parametric real-time program: the ABR conformance protocol. In *CAV'99*, Lecture Notes in Computer Science. Springer-Verlag, 1999. To appear.
8. D. Clark, E. M. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: a practical approach. In *Proc. 10th ACM Symp. on Principles of Programming Languages*, 1983.
9. E. W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, Englewood Cliffs, NJ, 1976.
10. L. Fribourg. A closed-form evaluation for extended timed automata. Research Report LSV-98-2, Lab. Specification and Verification, ENS de Cachan, Cachan, France, March 1998. 17 pages.
11. N. Halbwachs. *Synchronous Programming of Reactive Systems.* Kluwer Academic Publishers, 1993.
12. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *Lecture Notes in Computer Science*, 1254:460–463, 1997.
13. ITU-T. *Traffic control and congestion control in B-ISDN.* Recommendation I.371.1.
14. C. Jard, J.-F. Monin, and R. Groz. Development of Veda, a Prototyping Tool for Distributed Algorithms. *IEEE Transactions on Software Engineering*, 14(3):339–352, march 1988.

15. Jean-François Monin. Proving a real time algorithm for ATM in Coq. In E. Gimenez and C. Paulin-Mohring, editors, *Types for Proofs and Programs*, volume 1512 of *LNCS*, pages 277–293. Springer Verlag, 1998.
16. J. P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In *Proc. 5th Int'l Symp. on Programming*, Lecture Notes in Computer Science, Vol. 137, pages 337–371. SV, Berlin/New York, 1982.
17. Christophe Rabadan. L'ABR et sa conformité. NT DAC/ARP/034, CNET, 1997.
18. P. Z. Revesz. A closed-form evaluation for Datalog queries with integer (gap)-order constraints. *Theoretical Computer Science*, 116(1):117–149, August 1993.

## A Pseudo-code for Algorithm B'

When real time reaches $t_k$ :

**if** $t_k < \texttt{tfi}$ **then**
  **if** $\texttt{Emx} \leq \mathrm{ER}_k$ **then**
    **if** $\texttt{tfi} < t_k + \tau_3$ **then**
      **if** $t_k + \tau_3 < \texttt{tla} \vee \texttt{tfi} = \texttt{tla}$ **then**
        $\texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{tla} := t_k + \tau_3$
      **else**
        $\texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k$
    **else**
      **if** $\texttt{ACR} \leq \mathrm{ER}_k$ **then**
        $\texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{Efi} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{tfi} := t_k + \tau_3 \;||\; \texttt{tla} := t_k + \tau_3$
      **else**
        $\texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{Efi} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{tla} := \texttt{tfi}$
  **else**
    **if** $\mathrm{ER}_k < \texttt{Ela}$ **then**
      $\texttt{Efi} := \texttt{Emx} \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{tla} := t_k + \tau_2$
    **else**
      $\texttt{Efi} := \texttt{Emx} \;||\; \texttt{Ela} := \mathrm{ER}_k$
**else**
  **if** $\texttt{ACR} \leq \mathrm{ER}_k$ **then**
    $\texttt{Efi} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{tfi} := t_k + \tau_3 \;||\; \texttt{tla} := t_k + \tau_3$
  **else**
    $\texttt{Efi} := \mathrm{ER}_k \;||\; \texttt{Ela} := \mathrm{ER}_k \;||\; \texttt{Emx} := \mathrm{ER}_k \;||\; \texttt{tfi} := t_k + \tau_2 \;||\; \texttt{tla} := t_k + \tau_2$

When real time reaches `tfi`:

$\texttt{ACR} := \texttt{Efi} \;||\; \texttt{tfi} := \texttt{tla} \;||\; \texttt{Efi} := \texttt{Ela} \;||\; \texttt{Emx} := \texttt{Ela}$

If $\texttt{tfi} = t_k$, we run the algorithm for `tfi`, then the algorithm for $t_k$.

## B Proof of Algorithm B'

*Remark 1.* Proof obligations concerning the preservation of $\mathrm{I}_{\mathrm{tfs}}$ and of $\mathrm{I}_{\mathrm{Ub1}}$ have the form $[\mathsf{S_n}] \ldots s \le t \Rightarrow \ldots$, where $\mathsf{S_n}$ includes $\mathsf{T_e}$: we then have to prove $\ldots t_k \le t \Rightarrow \ldots$ Under the assumption $(G_e)$, $t_k \le t$ yields in fact $s \le t$, we may apply hypotheses of the form $\ldots s \le t \Rightarrow \ldots$ if Inv holds at time $s$.

*Remark 2.* We consider proof obligations of the form $[\mathsf{S}]\mathrm{Inv}$, where $[\mathsf{S}]$ is a substitution. It is decomposed into $[\mathsf{S}]\mathrm{I}_{\max}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{Ela}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{fil}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{tfs}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{Et1}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{Et2}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{ttE}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{Ub1}}$, $[\mathsf{S}]\mathrm{I}_{\mathrm{Ub2}}$ and $[\mathsf{S}]\mathrm{I}_{\mathrm{Ub3}}$. Some of them are immediate, for instance $\mathtt{Efi} < \mathrm{ER}_k \Rightarrow t_k+\tau_3 \le t_k+\tau_3$ or $\mathtt{Ela} < \mathtt{Ela} \Rightarrow \mathtt{tla} \le t_{\sharp l}+\tau_3$. They are skipped in order to save space.

### Case 1

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \le \mathrm{ER}_k$ **then if** $\mathtt{tfi} < t_k+\tau_3$
  **then if** $t_k+\tau_3 < \mathtt{tla} \vee \mathtt{tfi} = \mathtt{tla}$
  **then** $\mathtt{Emx} := \mathrm{ER}_k \;||\; \mathtt{Ela} := \mathrm{ER}_k \;||\; \mathtt{tla} := t_k+\tau_3$

Let $\mathsf{S_1}$ be the substitution $\mathsf{S_1} \stackrel{\mathrm{df}}{=} \mathsf{T_e} \;||\; \mathtt{Emx}, \mathtt{Ela}, \mathtt{tla} := \mathrm{ER}_k, \mathrm{ER}_k, t_k+\tau_3$. This transition is correct if :

$$\begin{array}{ll}
\mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & \\
\quad t_k < \mathtt{tfi} \quad \wedge & (G_{11}) \\
\quad \mathtt{Emx} \le \mathrm{ER}_k \quad \wedge & (G_{12}) \\
\quad \mathtt{tfi} < t_k+\tau_3 \quad \wedge & (G_{13}) \\
\quad (t_k+\tau_3 < \mathtt{tla} \vee \mathtt{tfi} = \mathtt{tla}) \quad \Rightarrow & (G_{14}) \\
\qquad [\mathsf{S_1}]\mathrm{Inv}. &
\end{array}$$

*Proof.* We assume Inv, $(G_e)$, $(G_{11})$, $(G_{12})$, $(G_{13})$, $(G_{14})$, and we prove $[\mathsf{S_1}]\mathrm{Inv}$.

- $[\mathsf{S_1}]\mathrm{I}_{\max}$, that is $\mathrm{ER}_k = \max(\mathtt{Efi}, \mathrm{ER}_k)$: by $\mathrm{I}_{\max}$ and $(G_{12})$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{fil}}$, that is $\mathtt{tfi} \le t_k+\tau_3 \le t_k+\tau_2$: trivial from $(G_{13})$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{tfs}}$, that is $\mathtt{tfi} \le t_k \;\Rightarrow\; \forall t,\; t_k \le t \;\Rightarrow\; \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$: absurd hypothesis, given $(G_{11})$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{Et1}}$, that is $\mathtt{ACR} < \mathtt{Efi} \Rightarrow \mathtt{tfi} \le t_k+\tau_3$: the conclusion comes from $(G_{13})$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{ttE}}$, that is $\mathtt{tfi} = t_k+\tau_3 \Rightarrow \mathtt{Efi} = \mathrm{ER}_k$: absurd hypothesis, given $(G_{13})$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\; t_k \le t < \mathtt{tfi} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$:
  for $t$ such that $t_k \le t < \mathtt{tfi}$, we get $t < t_k+\tau_3$ by $(G_{13})$, then lemma 4 yields $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l, t)$; we also have $s < t_k < \mathtt{tfi}$ by $(G_e)$, then we can apply $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1), and finally we get
  $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l, t) \le \mathtt{ACR}$.
- $[\mathsf{S_1}]\mathrm{I}_{\mathrm{Ub2}}$, that is $\forall t,\; \mathtt{tfi} \le t < t_k+\tau_3 \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{Efi}$:
  first remark that $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l, t)$ by lemma 4; $(G_{14})$ gives either $t_k+\tau_3 < \mathtt{tla}$, or $\mathtt{tfi} = \mathtt{tla}$;

- in the former case, $\mathrm{Approx}(l,t) \le \mathtt{Efi}$ by $\mathrm{I_{Ub2}}$, hence the result;
- in the latter case, $\mathrm{I_{ttE}}$ yields $\mathtt{Efi} = \mathtt{Ela}$, and $\mathtt{tla} = \mathtt{tfi} \le t$ yields $\mathrm{Approx}(l,t) \le \mathtt{Ela}$ by $\mathrm{I_{Ub3}}$, hence the result.

– $[\mathsf{S}_1]\mathrm{I_{Ub3}}$, that is $\forall t,\ t_k+\tau_3 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathrm{ER}_k$:
we show $\forall t,\ t_k+\tau_3 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$:
for $t$ such that $t_k+\tau_3 \le t$, we have $\mathtt{tfi} \le t$ by $(G_{13})$, then $\mathrm{Approx}(l,t) \le \mathtt{Emx} \le \mathrm{ER}_k$ by $\mathrm{I_{Apx}}$ and $(G_{12})$; lemma 4 yields either $\mathrm{Approx}(l^\frown t_k, t) = \max(\mathrm{Approx}(l,t), \mathrm{ER}_k)$ or $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$; in both cases, we see that $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$.

### Case 2

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \le \mathrm{ER}_k$ **then if** $\mathtt{tfi} < t_k+\tau_3$ **then**
  **if** $t_k+\tau_3 < \mathtt{tla} \vee \mathtt{tfi} = \mathtt{tla}$ **then**
  **else** $\mathtt{Emx} := \mathrm{ER}_k \ \| \ \mathtt{Ela} := \mathrm{ER}_k$

Let $\mathsf{S}_2$ be the substitution $\mathsf{S}_2 \stackrel{\mathrm{df}}{=} \mathsf{T_e} \ \| \ \mathtt{Emx}, \mathtt{Ela} := \mathrm{ER}_k, \mathrm{ER}_k$. This transition is correct if :

$$
\begin{array}{llr}
\mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & & \\
\quad t_k < \mathtt{tfi} \quad \wedge & & (G_{11}) \\
\quad \mathtt{Emx} \le \mathrm{ER}_k \quad \wedge & & (G_{12}) \\
\quad \mathtt{tfi} < t_k+\tau_3 \quad \wedge & & (G_{13}) \\
\quad \mathtt{tla} \le t_k+\tau_3 \quad \wedge & & (G_{24}) \\
\quad \mathtt{tfi} \ne \mathtt{tla} \quad \Rightarrow & & (G_{25}) \\
\qquad [\mathsf{S}_2]\mathrm{Inv}. & &
\end{array}
$$

*Proof.* $[\mathsf{S}_2]\mathrm{I_{max}}$, $[\mathsf{S}_2]\mathrm{I_{tfs}}$ and $[\mathsf{S}_2]\mathrm{I_{Et1}}$, are proved as in case 1.

– $[\mathsf{S}_2]\mathrm{I_{fil}}$, that is $\mathtt{tfi} \le \mathtt{tla} \le t_k+\tau_2$: trivial from $\mathrm{I_{fil}}$ $(G_{24})$.
– $[\mathsf{S}_2]\mathrm{I_{Et2}}$, that is $\mathtt{Efi} < \mathrm{ER}_k \Rightarrow \mathtt{tla} \le t_k+\tau_3$: the conclusion is $(G_{24})$.
– $[\mathsf{S}_2]\mathrm{I_{ttE}}$, that is $\mathtt{tfi} = \mathtt{tla} \Rightarrow \mathtt{Efi} = \mathrm{ER}_k$: absurd hypothesis, given $(G_{25})$.
– $[\mathsf{S}_2]\mathrm{I_{Ub1}}$, that is $\forall t,\ t_k \le t < \mathtt{tfi} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$:
using $(G_{13})$, lemma 4 and $\mathrm{I_{Ub1}}$ (see remark 1), we have $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l,t) \le \mathtt{ACR}$.
– $[\mathsf{S}_2]\mathrm{I_{Ub2}}$, that is $\forall t,\ \mathtt{tfi} \le t < \mathtt{tla} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{Efi}$:
$\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l,t)$ by lemma 4 and $(G_{24})$; $\mathrm{Approx}(l,t) \le \mathtt{Efi}$ by $\mathrm{I_{Ub2}}$, hence the result.
– $[\mathsf{S}_2]\mathrm{I_{Ub3}}$, that is $\forall t,\ \mathtt{tla} \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathrm{ER}_k$:
we have $\mathtt{tfi} \le t$ by $\mathrm{I_{fil}}$, then $\mathrm{Approx}(l,t) \le \mathtt{Emx} \le \mathrm{ER}_k$ by $\mathrm{I_{Apx}}$ and $(G_{12})$; taking $M = \mathrm{ER}_k$ in lemma 5 gives $\mathrm{Approx}(l^\frown t_k, t) \le \mathrm{ER}_k$.

**Case 3**

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \leq \mathrm{ER}_k$ **then if** $\mathtt{tfi} < t_k+\tau_3$ **then**
**else if** $\mathtt{ACR} \leq \mathrm{ER}_k$ **then**
$\mathtt{Emx} := \mathrm{ER}_k \;||\; \mathtt{Efi} := \mathrm{ER}_k \;||\; \mathtt{Ela} := \mathrm{ER}_k \;||\; \mathtt{tfi} := t_k+\tau_3 \;||\; \mathtt{tla} := t_k+\tau_3$

Let $\mathsf{S}_3$ be the substitution

$$\mathsf{S}_3 \;\stackrel{\mathrm{df}}{=}\; \mathsf{T_e} \;||\; \mathtt{Efi},\mathtt{Ela},\mathtt{Emx},\mathtt{tfi},\mathtt{tla} := \mathrm{ER}_k,\mathrm{ER}_k,\mathrm{ER}_k,t_k+\tau_3,t_k+\tau_3.$$

This transition is correct if :

$$\begin{array}{lr}
\mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & \\
\quad t_k < \mathtt{tfi} \quad \wedge & (G_{11}) \\
\quad \mathtt{Emx} \leq \mathrm{ER}_k \quad \wedge & (G_{12}) \\
\quad t_k+\tau_3 \leq \mathtt{tfi} \quad \wedge & (G_{33}) \\
\quad \mathtt{ACR} \leq \mathrm{ER}_k \quad \Rightarrow & (G_{34}) \\
\qquad [\mathsf{S}_3]\mathrm{Inv}. &
\end{array}$$

*Proof.*

– $[\mathsf{S}_3]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \leq t < t_k+\tau_3 \Rightarrow \mathrm{Approx}(l^\frown t_k,t) \leq \mathtt{ACR}$: using lemma 4 and $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1), we have $\mathrm{Approx}(l^\frown t_k,t) = \mathrm{Approx}(l,t) \leq \mathtt{ACR}$.
– $[\mathsf{S}_3]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t,\ t_k+\tau_3 \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k,t) \leq \mathrm{ER}_k$: we show $\forall t,\ t_k+\tau_3 \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k,t) = \mathrm{ER}_k$; we have either $t < \mathtt{tfi}$ or $\mathtt{tfi} \leq t$;
  - in the first case, $\mathrm{Approx}(l,t) \leq \mathtt{ACR} \leq \mathrm{ER}_k$ by $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1; here $t_k \leq t$ comes from $t_k \leq t_k+\tau_3 \leq t$) and $(G_{34})$;
  - in the second case, $\mathrm{Approx}(l,t) \leq \mathtt{Emx} \leq \mathrm{ER}_k$ by $\mathrm{I}_{\mathrm{Apx}}$ and $(G_{12})$;

  hence $\mathrm{Approx}(l,t) \leq \mathrm{ER}_k$ is always true; using lemma 4 we get $\mathrm{Approx}(l^\frown t_k,t) = \mathrm{ER}_k$ for $t$ such that $t_k+\tau_3 \leq t$.

**Case 4**

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \leq \mathrm{ER}_k$
**then if** $\mathtt{tfi} < t_k+\tau_3$ **then**
**else** $\mathtt{Emx} := \mathrm{ER}_k \;||\; \mathtt{Efi} := \mathrm{ER}_k \;||\; \mathtt{Ela} := \mathrm{ER}_k \;||\; \mathtt{tla} := \mathtt{tfi}$

Let $\mathsf{S}_4$ be the substitution

$$\mathsf{S}_4 \;\stackrel{\mathrm{df}}{=}\; \mathsf{T_e} \;||\; \mathtt{Efi},\mathtt{Ela},\mathtt{Emx},\mathtt{tla} := \mathrm{ER}_k,\mathrm{ER}_k,\mathrm{ER}_k,\mathtt{tfi}.$$

This transition is correct if :

$$\begin{array}{lr}
\mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & \\
\quad t_k < \mathtt{tfi} \quad \wedge & (G_{11}) \\
\quad \mathtt{Emx} \leq \mathrm{ER}_k \quad \wedge & (G_{12}) \\
\quad t_k+\tau_3 \leq \mathtt{tfi} \quad \wedge & (G_{33}) \\
\quad \mathrm{ER}_k < \mathtt{ACR} \quad \Rightarrow & (G_{44}) \\
\qquad [\mathsf{S}_4]\mathrm{Inv}. &
\end{array}$$

*Proof.*

- $[\mathsf{S}_4]\mathrm{I}_{\mathrm{fil}}$, that is $\mathtt{tfi} \leq \mathtt{tfi} \leq t_k + \tau_2$:
  we have $\mathtt{tfi} \leq t_{\sharp l} + \tau_2 = t_{k-1} + \tau_2 \leq t_k + \tau_2$ by $\mathrm{I}_{\mathrm{fil}}$, definition of $\sharp l$ and (7).
- $[\mathsf{S}_4]\mathrm{I}_{\mathrm{tfs}}$, that is $\mathtt{tfi} \leq t_k \Rightarrow \forall t,\ t_k \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$: hypothesis absurd, given $(G_{11})$.
- $[\mathsf{S}_4]\mathrm{I}_{\mathrm{Et1}}$, that is $\mathtt{ACR} < \mathrm{ER}_k \Rightarrow \mathtt{tfi} \leq t_k + \tau_3$: absurd hypothesis, given $(G_{44})$.
- $[\mathsf{S}_4]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \leq t < \mathtt{tfi} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$:
  by $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1) and $t < \mathtt{tfi}$, we have $\mathrm{Approx}(l, t) \leq \mathtt{ACR}$; taking $M = \mathtt{ACR}$ in lemma 5 and using $(G_{44})$ yields $\mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$.
- $[\mathsf{S}_4]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t, \mathtt{tfi} \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathrm{ER}_k$: we show $\forall t,\ \mathtt{tfi} \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$; for $t$ such that $\mathtt{tfi} \leq t$, we have $\mathrm{Approx}(l, t) \leq \mathtt{Emx} \leq \mathrm{ER}_k$ by $\mathrm{I}_{\mathrm{Apx}}$ and $(G_{12})$; we have $t_k + \tau_3 \leq \mathtt{tfi} \leq t$ by $(G_{33})$, then lemma 4 yields either $\mathrm{Approx}(l^\frown t_k, t) = \max(\mathrm{Approx}(l, t), \mathrm{ER}_k)$ or $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$; in both cases, we see that $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$.

**Case 5**

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \leq \mathrm{ER}_k$ **then else if** $\mathrm{ER}_k < \mathtt{Ela}$
**then** $\mathtt{Efi} := \mathtt{Emx} \ \|\ \mathtt{Ela} := \mathrm{ER}_k \ \|\ \mathtt{tla} := t_k + \tau_2$

Let $\mathsf{S}_5$ be the substitution

$$\mathsf{S}_5 \ \stackrel{\mathrm{df}}{=}\ \mathsf{T}_\mathsf{e} \ \|\ \mathtt{Efi}, \mathtt{Ela}, \mathtt{tla} := \mathtt{Emx}, \mathrm{ER}_k, t_k + \tau_2.$$

This transition is correct if :

$$\begin{aligned}
&\mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge \\
&\quad t_k < \mathtt{tfi} \quad \wedge && (G_{11})\\
&\quad \mathrm{ER}_k < \mathtt{Emx} \quad \wedge && (G_{52})\\
&\quad \mathrm{ER}_k < \mathtt{Ela} \qquad \Rightarrow && (G_{53})\\
&\qquad [\mathsf{S}_5]\mathrm{Inv}.
\end{aligned}$$

*Proof.* $[\mathsf{S}_5]\mathrm{I}_{\mathrm{fil}}$ and $[\mathsf{S}_5]\mathrm{I}_{\mathrm{tfs}}$ are similar to $[\mathsf{S}_4]\mathrm{I}_{\mathrm{fil}}$ and $[\mathsf{S}_4]\mathrm{I}_{\mathrm{tfs}}$

- $[\mathsf{S}_5]\mathrm{I}_{\max}$, that is $\mathtt{Emx} = \max(\mathtt{Emx}, \mathrm{ER}_k)$: by $(G_{52})$.
- $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Et1}}$, that is $\mathtt{ACR} < \mathtt{Emx} \Rightarrow \mathtt{tfi} \leq t_k + \tau_3$: we have $\mathtt{Efi} < \mathtt{Ela}$ or $\mathtt{Ela} \leq \mathtt{Efi}$;
  - in the first case, $\mathtt{tfi} \leq \mathtt{tla} \leq t_{\sharp l} + \tau_3$ by $\mathrm{I}_{\mathrm{fil}}$ and $\mathrm{I}_{\mathrm{Et2}}$;
  - in the second case, $\mathtt{Emx} = \mathtt{Efi}$, then $\mathtt{tfi} \leq t_{\sharp l} + \tau_3$ by $\mathrm{I}_{\mathrm{Et1}}$ and $\mathrm{I}_{\max}$;

  in both cases, $\mathtt{tfi} \leq t_{k-1} + \tau_3 \leq t_k + \tau_3$ by definition of $\sharp l$ and (7).
- $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Et2}}$, that is $\mathtt{Emx} < \mathrm{ER}_k \Rightarrow t_k + \tau_2 \leq t_k + \tau_3$:
  the hypothesis $\mathtt{Emx} < \mathrm{ER}_k$ is absurd given $(G_{52})$.
- $[\mathsf{S}_5]\mathrm{I}_{\mathrm{ttE}}$, that is $\mathtt{tfi} = t_k + \tau_2 \Rightarrow \mathtt{Emx} = \mathrm{ER}_k$:
  we have $\mathtt{tfi} \leq t_{\sharp l} + \tau_2 = t_{k-1} + \tau_2 < t_k + \tau_2$ by $\mathrm{I}_{\mathrm{fil}}$, definition of $\sharp l$ and (7); then the hypothesis $\mathtt{tfi} = t_k + \tau_2$ is absurd.
- $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \leq t < \mathtt{tfi} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$:
  by $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1) and $t < \mathtt{tfi}$, we have $\mathrm{Approx}(l, t) \leq \mathtt{ACR}$; we also have $\mathtt{tfi} \leq t_k + \tau_3$ or $t_k + \tau_3 < \mathtt{tfi}$;

- in the first case, $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l, t) \le \mathtt{ACR}$ by lemma 4;
- in the second case, $\mathrm{ER}_k < \mathtt{Emx} \le \mathtt{ACR}$ by $(G_{52})$ and contraposition of $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Et1}}$ shown above; taking $M = \mathtt{ACR}$ in lemma 5 yields $\mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$.

– $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Ub2}}$, that is $\forall t,\ \mathtt{tfi} \le t < t_k + \tau_2 \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{Emx}$:
for $t$ such that $\mathtt{tfi} \le t$, we have $\mathrm{Approx}(l, t) \le \mathtt{Emx}$, by $\mathrm{I}_{\mathrm{Apx}}$; taking $M = \mathtt{Emx}$ in lemma 5 and using $(G_{52})$ yields $\mathrm{Approx}(l^\frown t_k, t) \le \mathtt{Emx}$.

– $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t,\ t_k + \tau_2 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathrm{ER}_k$:
by lemma 4, we have $\forall t,\ t_k + \tau_2 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$.

## Case 6

**if** $t_k < \mathtt{tfi}$ **then if** $\mathtt{Emx} \le \mathrm{ER}_k$ **then else if** $\mathrm{ER}_k < \mathtt{Ela}$ **then**
**else** $\mathtt{Efi} := \mathtt{Emx} \ ||\ \mathtt{Ela} := \mathrm{ER}_k$

Let $\mathsf{S}_6$ be the substitution

$$\mathsf{S}_6 \ \stackrel{\mathrm{df}}{=}\ \mathsf{T}_{\mathsf{e}} \ ||\ \mathtt{Efi}, \mathtt{Ela} := \mathtt{Emx}, \mathrm{ER}_k.$$

This transition is correct if :

$$\begin{array}{llr} \mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & & \\ \quad t_k < \mathtt{tfi} \quad \wedge & & (G_{11}) \\ \quad \mathrm{ER}_k < \mathtt{Emx} \quad \wedge & & (G_{52}) \\ \quad \mathtt{Ela} \le \mathrm{ER}_k & \Rightarrow & (G_{63}) \\ \quad\quad [\mathsf{S}_6]\mathrm{Inv}. & & \end{array}$$

*Proof.* $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Et1}}$ and $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Et2}}$ are similar to $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Et1}}$ and $[\mathsf{S}_5]\mathrm{I}_{\mathrm{Et2}}$.

– $[\mathsf{S}_6]\mathrm{I}_{\max}$, that is $\mathtt{Emx} = \max(\mathtt{Emx}, \mathrm{ER}_k)$: by $(G_{52})$.

– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{fil}}$, that is $\mathtt{tfi} \le \mathtt{tla} \le t_k + \tau_2$: we have
$\mathtt{tfi} \le \mathtt{tla} \le t_{\sharp l} + \tau_2 = t_{k-1} + \tau_2 \le t_k + \tau_2$ by $\mathrm{I}_{\mathrm{fil}}$, definition of $\sharp l$ and (7).

– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{tfs}}$, that is $\mathtt{tfi} \le t_k \ \Rightarrow\ \forall t,\ t_k \le t \ \Rightarrow\ \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$:
the hypothesis is absurd given $(G_{11})$.

– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{ttE}}$, that is $\mathtt{tfi} = \mathtt{tla} \Rightarrow \mathtt{Emx} = \mathrm{ER}_k$:
we use a weakened form of $(G_{52})$:

$$\mathrm{ER}_k \le \mathtt{Emx}; \qquad (G_{52'})$$

assuming $\mathtt{tfi} = \mathtt{tla}$, we have $\mathtt{Efi} = \mathtt{Ela} = \mathtt{Emx}$ by $\mathrm{I}_{\max}$, then $\mathtt{Emx} = \mathtt{Ela} \le \mathrm{ER}_k$ by $(G_{63})$; $\mathtt{Emx} \le \mathrm{ER}_k$ and $(G_{52'})$ yields $\mathtt{Emx} = \mathrm{ER}_k$.

– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \le t < \mathtt{tfi} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$:
by $\mathrm{I}_{\mathrm{Ub1}}$ (see remark 1) and $t < \mathtt{tfi}$, we have $\mathrm{Approx}(l, t) \le \mathtt{ACR}$; we also have $\mathtt{tfi} \le t_k + \tau_3$ or $t_k + \tau_3 < \mathtt{tfi}$;
- in the first case, $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l, t) \le \mathtt{ACR}$ by lemma 4;
- in the second case, $\mathrm{ER}_k < \mathtt{Emx} \le \mathtt{ACR}$ by $(G_{52})$ and contraposition of $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Et1}}$ shown above; taking $M = \mathtt{ACR}$ in lemma 5 yields $\mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$.

– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Ub2}}$, that is $\forall t,\ \mathtt{tfi} \leq t < \mathtt{tla} \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{Emx}$:
  for $t$ such that $\mathtt{tfi} \leq t$, we have $\mathrm{Approx}(l,t) \leq \mathtt{Emx}$, by $\mathrm{I}_{\mathrm{Apx}}$; taking $M = \mathtt{Emx}$ in lemma 5 and using $(G_{52})$ yields $\mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{Emx}$.
– $[\mathsf{S}_6]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t,\ \mathtt{tla} \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathrm{ER}_k$:
  for $t$ such that $\mathtt{tla} \leq t$, we have $\mathrm{Approx}(l,t) \leq \mathtt{Ela} \leq \mathrm{ER}_k$ by $\mathrm{I}_{\mathrm{Ub3}}$ and $(G_{63})$; taking $M = \mathrm{ER}_k$ in lemma 5 yields $\mathrm{Approx}(l^\frown t_k, t) \leq \mathrm{ER}_k$.
  Note that if we can ensure $t_k + \tau_3 \leq \mathtt{tla} \leq t$, we can show
  $\forall t,\ \mathtt{tla} \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$.

### Case 7

**if** $t_k < \mathtt{tfi}$ **then else if** $\mathtt{ACR} \leq \mathrm{ER}_k$ **then**
$\mathtt{Efi} := \mathrm{ER}_k \ \|\ \mathtt{Ela} := \mathrm{ER}_k \ \|\ \mathtt{Emx} := \mathrm{ER}_k \ \|\ \mathtt{tfi} := t_k + \tau_3 \ \|\ \mathtt{tla} := t_k + \tau_3$

Let $\mathsf{S}_7$ be the substitution

$$\mathsf{S}_7 \ \stackrel{\mathrm{df}}{=}\ \mathsf{T}_\mathsf{e} \ \|\ \mathtt{Efi}, \mathtt{Ela}, \mathtt{Emx}, \mathtt{tfi}, \mathtt{tla} := \mathrm{ER}_k, \mathrm{ER}_k, \mathrm{ER}_k, t_k + \tau_3, t_k + \tau_3.$$

This transition is correct if :

$$\begin{array}{llr} \mathrm{Inv} \quad \wedge \quad (G_e) \quad \wedge & & \\ \quad \mathtt{tfi} \leq t_k \quad \wedge & & (G_{71}) \\ \quad \mathtt{ACR} \leq \mathrm{ER}_k & \Rightarrow & (G_{72}) \\ \qquad [\mathsf{S}_7]\mathrm{Inv}. & & \end{array}$$

*Proof.*

– $[\mathsf{S}_3]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \leq t < t_k + \tau_3 \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$:
  we remark that $\mathtt{tfi} \leq s$ by $(G_e)$ and $(G_{71})$, then $\mathrm{Approx}(l,t) \leq \mathtt{ACR}$ by $\mathrm{I}_{\mathrm{tfs}}$ (see remark 1); using lemma 4 we have $\mathrm{Approx}(l^\frown t_k, t) = \mathrm{Approx}(l,t)$, hence $\mathrm{Approx}(l^\frown t_k, t) \leq \mathtt{ACR}$.
– $[\mathsf{S}_3]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t,\ t_k + \tau_3 \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \leq \mathrm{ER}_k$:
  we show $\forall t,\ t_k + \tau_3 \leq t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$; we remark that $\mathtt{tfi} \leq s$ by $(G_e)$ and $(G_{71})$, then $\mathrm{Approx}(l,t) \leq \mathtt{ACR} \leq \mathrm{ER}_k$ by $\mathrm{I}_{\mathrm{tfs}}$ (see remark 1; here $t_k < t$ comes from $t_k < t_k + \tau_3 \leq t$) and $(G_{72})$; using the assumption $t_k + \tau_3 \leq t$ and lemma 4, we know that $\mathrm{Approx}(l^\frown t_k, t)$ is either equal to $\max(\mathrm{Approx}(l,t), \mathrm{ER}_k)$ or to $\mathrm{ER}_k$, that is, in both cases, to $\mathrm{ER}_k$.

### Case 8

**if** $t_k < \mathtt{tfi}$ **then else if** $\mathtt{ACR} \leq \mathrm{ER}_k$ **then else**
$\mathtt{Efi} := \mathrm{ER}_k \ \|\ \mathtt{Ela} := \mathrm{ER}_k \ \|\ \mathtt{Emx} := \mathrm{ER}_k \ \|\ \mathtt{tfi} := t_k + \tau_2 \ \|\ \mathtt{tla} := t_k + \tau_2$

Let $\mathsf{S}_8$ be the substitution

$$\mathsf{S}_8 \ \stackrel{\mathrm{df}}{=}\ \mathsf{T}_\mathsf{e} \ \|\ \mathtt{Efi}, \mathtt{Ela}, \mathtt{Emx}, \mathtt{tfi}, \mathtt{tla} := \mathrm{ER}_k, \mathrm{ER}_k, \mathrm{ER}_k, t_k + \tau_2, t_k + \tau_2.$$

This transition is correct if :

$$\begin{aligned} &\text{Inv} \quad \wedge \quad (G_e) \quad \wedge \\ &\qquad \mathtt{tfi} \le t_k \quad \wedge && (G_{71}) \\ &\qquad \mathrm{ER}_k < \mathtt{ACR} \qquad \Rightarrow && (G_{82}) \\ &\qquad\quad [\mathsf{S}_8]\text{Inv}. \end{aligned}$$

*Proof.*

- $[\mathsf{S}_8]\mathrm{I}_{\mathrm{tfs}}$ and $[\mathsf{S}_8]\mathrm{I}_{\mathrm{Ub2}}$ are similar to $[\mathsf{S}_7]\mathrm{I}_{\mathrm{tfs}}$ and $[\mathsf{S}_7]\mathrm{I}_{\mathrm{Ub2}}$, replacing $\tau_3$ with $\tau_2$.
- $[\mathsf{S}_8]\mathrm{I}_{\mathrm{Et1}}$, that is $\mathtt{ACR} < \mathrm{ER}_k \Rightarrow t_k+\tau_2 \le t_k+\tau_3$: the hypothesis $\mathtt{ACR} < \mathrm{ER}_k$ is absurd given $(G_{82})$.
- $[\mathsf{S}_8]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t,\ t_k \le t < t_k+\tau_2 \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$: for $t$ such that $t_k \le t$, we have $\mathtt{tfi} \le s$ by $(G_e)$ and $(G_{71})$, then $\mathrm{Approx}(l,t) \le \mathtt{ACR}$ by $\mathrm{I}_{\mathrm{tfs}}$ (see remark 1); taking $M = \mathtt{ACR}$ in lemma 5 and using $(G_{82})$ yields $\mathrm{Approx}(l^\frown t_k, t) \le \mathtt{ACR}$.
- $[\mathsf{S}_8]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t,\ t_k+\tau_2 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) \le \mathrm{ER}_k$: by lemma 4, we have $\forall t,\ t_k+\tau_2 \le t \Rightarrow \mathrm{Approx}(l^\frown t_k, t) = \mathrm{ER}_k$.

**Internal Event.** We prove: $\text{Inv} \wedge (G_i) \Rightarrow [\mathsf{S}_\mathsf{i}]\text{Inv}$.

*Proof.*

- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{Ela}}$, that is $\mathtt{Ela} = \mathrm{ER}_{\sharp l}$: from $\mathrm{I}_{\mathrm{Ela}}$.
- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{fil}}$, that is $\mathtt{tla} \le \mathtt{tla} \le t_{\sharp l}+\tau_2$: we have $\mathtt{tla} \le t_{\sharp l}+\tau_2$ by $\mathrm{I}_{\mathrm{fil}}$.
- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{tfs}}$, that is $\mathtt{tla} \le \mathtt{tfi} \ \Rightarrow\ \forall t,\ \mathtt{tfi} \le t \ \Rightarrow\ \mathrm{Approx}(l,t) \le \mathtt{Efi}$: $\mathtt{tla} \le \mathtt{tfi}$ and $\mathrm{I}_{\mathrm{fil}}$ yields $\mathtt{tfi} = \mathtt{tla}$, then $\mathtt{Efi} = \mathtt{Ela}$ by $\mathrm{I}_{\mathrm{ttE}}$; we then have to show that for $t$ such that $\mathtt{tla} \le t$, we have $\mathrm{Approx}(l,t) \le \mathtt{Ela}$: we just apply $\mathrm{I}_{\mathrm{Ub3}}$.
- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{Et1}}$, that is $\mathtt{Efi} < \mathtt{Ela} \Rightarrow \mathtt{tla} \le t_{\sharp l}+\tau_3$: it is just $\mathrm{I}_{\mathrm{Et2}}$.
- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t\ \mathtt{tfi} \le t < \mathtt{tla} \ \Rightarrow\ \mathrm{Approx}(l,t) \le \mathtt{Efi}$: it is just $\mathrm{I}_{\mathrm{Ub2}}$.
- $[\mathsf{S}_\mathsf{i}]\mathrm{I}_{\mathrm{Ub3}}$, that is $\forall t\ \mathtt{tla} \le t \ \Rightarrow\ \mathrm{Approx}(l,t) \le \mathtt{Ela}$: for $t$ such that $\mathtt{tla} \le t$, we have $s \le \mathtt{tla} \le t$ by $(G_i)$; we can then apply $\mathrm{I}_{\mathrm{Ub3}}$, which yields $\mathrm{Approx}(l,t) \le \mathtt{Ela}$.

**Observation Event.** We prove: $\text{Inv} \wedge (G_o) \Rightarrow [\mathsf{T}_\mathsf{o}]\text{Inv}$.

*Proof.*

- $[\mathsf{T}_\mathsf{o}]\mathrm{I}_{\mathrm{tfs}}$, that is $\mathtt{tfi} \le s' \ \Rightarrow\ \forall t,\ s' \le t \ \Rightarrow\ \mathrm{Approx}(l,t) \le \mathtt{ACR}$: from $\mathtt{tfi} \le s'$ and $(G_o)$ we get $\mathtt{tfi} \le s$; for $t$ such that $s' \le t$, we have $s \le t$ by $(G_o)$ then $\mathrm{Approx}(l,t) \le \mathtt{ACR}$ by $\mathrm{I}_{\mathrm{tfs}}$.
- $[\mathsf{T}_\mathsf{o}]\mathrm{I}_{\mathrm{Ub1}}$, that is $\forall t\ s' \le t < \mathtt{tfi} \ \Rightarrow\ \mathrm{Approx}(l,t) \le \mathtt{ACR}$: for $t$ such that $s' \le t < \mathtt{tfi}$ we have $s < s' \le t < \mathtt{tfi}$ by $(G_o)$, then $\mathrm{Approx}(l,t) \le \mathtt{ACR}$ by $\mathrm{I}_{\mathrm{Ub1}}$.