Compositional Invariant Generation for Timed Systems

L. Aştefănoaei, S. Ben Rayana, S. Bensalem, M. Bozga, J. Combaz

Verimag Research Report n^o TR-2013-5

July 11, 2013

Unité Mixte de Recherche 5104 CNRS - Grenoble INP - UJF

Centre Equation
2, avenue de VIGNATE
F-38610 GIERES
tel : +33 456 52 03 40
fax : +33 456 52 03 50
http://www-verimag.imag.fr

Compositional Invariant Generation for Timed Systems

L. Aştefănoaei, S. Ben Rayana, S. Bensalem, M. Bozga, J. Combaz

July 11, 2013

Abstract

In this paper we approach the state-explosion problem when model-checking timed systems with a great number of components. Our solution consists in adapting a rule for compositionally verifying systems of extended finite state machines [3] to timed systems. The main difficulty is in the lack of information about the relations between local timings. We propose to strengthen the verification rule with inequalities between local timings which we show to be invariants of the global system, thus the soundness of the new verification rule is preserved.

Keywords: compositional verification, timed automata, invariants

Reviewers: M. Bozga

Notes: Work supported by the European Integrated Project 257414 ASCENS (`www.ascens-ist.eu`).

How to cite this report:

```
@techreport {TR-2013-5,
  title = {Compositional Invariant Generation for Timed Systems},
  author = {L. Aştefănoaei, S. Ben Rayana, S. Bensalem, M. Bozga, J. Combaz},
  institution = {{Verimag} Research Report},
  number = {TR-2013-5},
  year = {}
}
```

1 Introduction

Motivation: We approach the state-explosion problem when model-checking timed systems with a great number of components. Our solution consists in adapting a rule for compositionally verifying systems of extended finite state machines [3] to timed systems.
Context: When it comes to formalising a particular verification problem, there are several options for modelling behaviour and expressing properties, be it LTS, TA, hybrid systems on the part of behaviour, or be it a specific logic on the part of properties. All of them have certain common concepts which can be factored out and abstracted into a *generic framework* s.t. each class can then be seen as an instantiation.
A Generic Approach for Compositional Verification: Given a generic framework (GF) consisting of:

1. an *operational level* to characterise the *behaviour* of systems in terms of the behaviour of the constituting *components* B_i interacting via a coordination mechanism γ, denoted as $\|_\gamma B_i$

2. a *logical level* to characterise *properties* of the system, usually denoted by Φ, of components, $CI(B_i)$, and of coordination, $II(\gamma)$,

verify if a given (usually safety) property is satisfied by the whole system in a *compositional* manner by means of a rule like:

$$\frac{\begin{array}{c} CI(B_1) \in Inv(B_1) \quad \ldots \quad CI(B_n) \in Inv(B_n) \quad II(\gamma) \in Inv(\|_\gamma B_i) \\ \vdash \Big(\bigwedge_i CI(B_i) \wedge II(\gamma) \rightarrow \Phi \Big) \end{array}}{\|_\gamma B_i \models \Box \Phi} \text{ (VR).}$$

GF is abstract in the sense that at the operational level components are understood as state machines, i.e., their behaviour is given in terms of *states* and *state transformers*, i.e., transitions s.t. one can further define a notion of *state successor* and a notion of *reachable set*. The properties we work with at the logical level are understood as invariants (properties that hold in every reachable state). The logical framework is assumed to be decidable such that $CI(B_i)$, $II(\gamma)$ can either be effectively computed or proved to be invariants. In the rule (VR), $Inv(B_i)$ denotes the set of invariants of component B_i, the symbol $\vdash$ is used to underline that the logical implication can be effectively proved (for instance with an SMT solver) and the notation $B \models \Box\Phi$ means that the predicate Φ holds in every state of B. We recall that in general $\|_\gamma B_i$ has a state space which is too big (possibly infinite) to directly exhaustively check in a feasible manner that Φ holds in every state. This is why it is important to be able to apply a rule such as (VR) which reduces the verification problem to components and interaction model separately. We note that though $CI(B_i)$, $II(\gamma)$ are invariants, we prefer to use the notation $CI(B) \in Inv(B)$ to stress that $CI(B)$ can be *effectively* either computed as an invariant or proved to be an invariant.

GF is generic in the sense that different instantiations of GF can be obtained by making explicit (grounding) the operational and the logical levels in GF. One example of instantiated GF is BIP +FOL for which the verification rule is exploited in the tool D-Finder [3].
Goal: Our goal is to present a different instantiation with respect to timed systems.
Challenge: Though the rule (VR) from GF is, by itself, sufficient to prove interesting safety properties in [3], this is not the case in the context of timed systems, where the rule is quite weak in the sense that it often raises false alarms (the so called "false positives"). This is mainly due to missing relations between the clocks of the system. To illustrate this, we consider an abstract scenario where a "controller" component serves "worker" components one at a time. For simplicity, Figure 1 depicts an instantiation with only one worker $Worker_1$ interacting with $Controller$ by synchronising on ports b_1 and d_1, i.e., the interaction model γ is given as the set $\{(a \mid b_1), (c \mid d_1)\}$. A safety property which the system should fulfil is that whenever $\beta \leq \alpha$, before synchronising on a and b, the time difference between the clock of the controller and that of the worker is less or equal than $\alpha - \beta$, i.e., $Safe = (lc_1 \wedge l_1) \Rightarrow x - y \leq \alpha - \beta$.

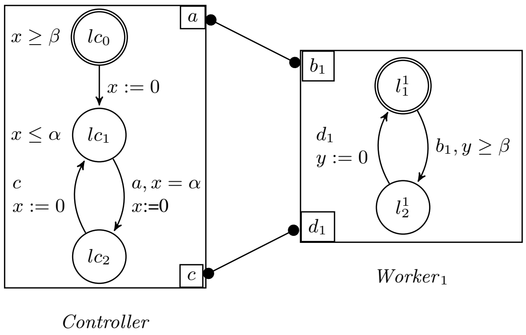

Figure 1: An Abstract Example as Illustration for the Weakness of (VR).

The property *Safe* is, indeed, a global invariant: $Controller \|_\gamma Worker \models \Box Safe$. However, given that the component invariants are[1]:

$$CI(Controller) = lc_0 \vee (lc_1 \wedge x \leq \alpha) \vee lc_2$$
$$CI(Worker_1) = l_1 \vee (l_2 \wedge y \geq \beta)$$

we can show (by hand or with a solver like Z3 [10] or yices [5]) that the conjunction $CI(Controller) \wedge CI(Worker_1) \wedge II(\gamma) \wedge \neg Safe$ is satisfiable, and thus (VR) cannot be used. Intuitively, the problem comes from the fact that the relations between the values of clocks in different components at synchronization time cannot be derived only from the component invariants alone, because what they offer is just a characterisation of the local clocks. In this paper, we propose a solution which makes use of additional clocks to store information about the timings at synchronizations. Furthermore, because of the inherent non-determinism in the interaction model, at a given time, there may be more interactions which can be fired. The order of execution of such competing interactions does not reflect at the component level. We look into solving such conflicting situations by extending our method based on history clocks from component to system level.

2 The Model

2.1 Operational Level

The operational level provides concrete definitions for the notions of **components**, **interaction models**, and **systems**. As mentioned in the introduction, in the framework of the present paper, components are timed automata and systems are compositions of timed automata with respect to interaction models where interactions are sets of ports on which components synchronise. Before detailing these definitions, we note that the timed automata we use are essentially the ones from [12] however sligthly adapted to embrace a uniform notation throughout the paper. We note that we restrict to this particular class because we want to have a common set of examples to compare our compositional approach with model-checking the whole system in Uppaal.

Definition 1 (Timed automaton (TA)). *A TA is a tuple $(L, P, T, \mathcal{X}, \mathtt{tpc})$ where L is a finite set of locations, P a finite set of ports, $T \subseteq L \times (P \times \mathcal{C} \times 2^{\mathcal{X}}) \times L$ is a set of edges labeled with an action, a guard, and a set of clocks to be reset, $\mathcal{X}$ is a finite set of clocks, and $\mathtt{tpc} : L \to \mathcal{C}$ assigns a time progress condition[2] to each location. $\mathcal{C}$ is the set of clock constraints. A clock constraint is defined by the grammar:*

$$C ::= true \mid false \mid x \# ct \mid x - y \# ct \mid C \wedge C$$

[1] We assume that clocks may only have positive values. Thus, for simplicity, we do not show in the component invariants inequalities like $x \geq 0$.

[2] To avoid confusion with invariant properties, we prefer to adopt the terminology of "time progress condition" from [4] instead of "location invariants".

with $x, y \in \mathcal{X}$, $\# \in \{<, \leq, =, \geq, >\}$ and $ct \in \mathbb{Z}$.

Time progress conditions and guards are clock constraints. Time progress conditions are restricted to constraints as $x \leq ct$. Guards are s.t. they are included in the clock invariants, i.e., given an edge $\big(l, (p, g, r), l'\big)$, $\mathsf{tpc}(l) \rightarrow g$ evaluates to true.

Definition 2 (Semantics of a timed automaton)**.** *The semantics of a timed automaton $TA = (L, P, T, X, \mathsf{tpc})$ is given by the LTS $sem(TA) = (Q, \Sigma, \rightarrow)$ where:*

- *$Q \subseteq L \times \mathbf{V}$ denotes the states of TA;*
- *$\rightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times Q$ denotes the transitions according to the rules:*
  - *$(l, \mathbf{v}) \xrightarrow{\delta} (l, \mathbf{v} + \delta)$ if $\big(\forall \delta' \in [0, \delta)\big).(\mathsf{tpc}(l)(\mathbf{v} + \delta'))$ (time progress);*
  - *$(l, \mathbf{v}) \xrightarrow{p} (l', \mathbf{v}')$ if $\big(l, (p, g, r), l'\big) \in T$, $g(\mathbf{v}) \wedge \mathsf{tpc}(l')(\mathbf{v}')$, with $\mathbf{v}' = \mathbf{v}[r]$ (action step).*

$\mathbf{V}$ is the set of all clock valuation functions $\mathbf{v} : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$. For a constraint C, $C(\mathbf{v})$ denotes the evaluation of C in $\mathbf{v}$. The notation $\mathbf{v} + \delta$ represents a new $\mathbf{v}'$ defined as $\mathbf{v}'(x) = \mathbf{v}(x) + \delta$ while $\mathbf{v}[r]$ represents a new $\mathbf{v}'$ defined as:

$$\mathbf{v}'(x) = \begin{cases} \mathbf{v}(x) & x \in X \setminus r \\ 0 & x \in r. \end{cases}$$

Because $sem(TA)$ is usually infinite, the finite symbolic representation that has been proposed instead is the so called the zone graph [12]. The symbolic states in a zone graph are pairs (l, ζ) where l is a location of TA and ζ is a *zone*, a conjunction of clock constraints, or equally, a polyhedron. Given a symbolic state (l, ζ) its successor with respect to a transition t of TA is denoted as $\mathsf{succ}(t, (l, \zeta))$ and defined by means of its timed and its discrete successor:

- $\mathsf{time\_succ}((l, \zeta)) = (l, \nearrow \zeta \cap \mathsf{tpc}(l))$
- $\mathsf{disc\_succ}(t, (l, \zeta)) = (l', (\zeta \cap g)[r] \cap \mathsf{tpc}(l'))$ if $t = \big(l, (\_, g, r), l'\big)$
- $\mathsf{succ}(t, (l, \zeta)) = \mathsf{close}(\mathsf{time\_succ}(\mathsf{disc\_succ}(t, (l, \zeta))), c)$

where $\nearrow$, $[r]$, close are usual operators on zones [12]. We briefly recall their meaning: $\nearrow \zeta$ is the forward diagonal projection of ζ, i.e., it contains any valuation $\mathbf{v}'$ for which there exists a real δ such that $\mathbf{v}' - \delta$ is in ζ; $\zeta[r]$ is the set of all valuations in ζ after applying the resets in r; $\mathsf{close}(\zeta, c)$ is the set of all valuations in ζ where one ignores the constraints with constants greater than c.

Given a TA with transitions T, the set of symbolic states reachable from a given symbolic state s is the set of all possible successors:

$$Reach(s) = \{s\} \cup \bigcup_{\substack{t \in T \\ s' \in \mathsf{succ}(t, s)}} Reach(s').$$

A symbolic execution of a TA starting from a symbolic state s_0 is a sequence of symbolic states $s_0, s_1, \ldots, s_n, \ldots$ such that for any $i > 0$ there exists a transition t for which $s_i \in \mathsf{succ}(t, s_{i-1})$.

Definition 3 (Interaction Model)**.** *Given n components B_i with P_i their sets of ports, an interaction model γ is a set of sets of ports, i.e., $\gamma \subseteq 2^{\cup_i P_i}$.*

Definition 4 (Timed System $\|_\gamma B_i$)**.** *Given n components $B_i = (L_i, P_i, T_i, \mathcal{X}_i, \mathsf{tpc}_i)$ with $P_i \cap P_j = \emptyset$, $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$, for any $i \neq j$, and an interaction model γ, $\|_\gamma B_i$ is the new timed automaton $(L, P, T_\gamma, \mathcal{X}, \mathsf{tpc})$ where $P = \gamma$, $\mathcal{X} = \cup_i \mathcal{X}_i$, $L = \times_i L_i$, $\mathsf{tpc}(\bar{l}) = \cap_i \mathsf{tpc}(l_i)$, T_γ is s.t.:*

- *for any interaction $\alpha \in \gamma$ s.t. $\alpha = \{p_i \mid i \in I\}$ with $I \subseteq \{1, \ldots, n\}$ and $p_i \in P_i$, we have that $\bar{l} \xrightarrow{\alpha, g, r} \bar{l}'$ where $\bar{l} = (l_1, \ldots, l_n)$, $g = \bigcap_{i \in I} g_i$, $r = \bigcup_{i \in I} r_i$, and $\bar{l}'$ is defined as:*

$$\bar{l}'(i) = \begin{cases} l_i' & \text{if } l_i \xrightarrow{p_i, g_i, r_i} l_i' \\ l_i & \text{owise} \end{cases}$$

The semantics of the system is given as TA semantics.

2.2 Logical Level

We recall from the introduction that invariants are state properties which hold in every reachable state and that we use $Inv(B)$ to denote the set of invariants of B. Next, we recall the definition of inductive invariants, which, in contrast to the general definition of invariants, is implementable.

Definition 5 (Inductive Invariant)**.** *Given a component B with initial state s_0 a property I is an inductive invariant of B if $s_0 \models I$ and $s \models I$ implies $s' \models I$ for any $s' \in \mathtt{succ}(t, s)$ and t a transition of B.*

Proposition 1. *If I is an inductive invariant of a component B and the implication $I \rightarrow \Phi$ is a valid formula, then Φ is an invariant of B, $B \models \Box\Phi$.*

Given a system consisting of n components B_i and an interaction model γ, the logical level provides concrete definitions to the notions of **component invariants** $CI(B_i)$, **interaction invariant** $II(\gamma)$. Our choice[3] is to work with component invariants as over-approximations of the state space:

- $CI(B_i) = \bigvee\limits_{(l,\zeta)\in Reach(s^0)} l \wedge \zeta$ where s^0 is the initial state of B_i

and interaction invariants as the minimal models satisfying implications about global locations which can be deduced from interaction models:

- $II(\gamma) = \bigwedge\limits_{L(\gamma)} \bigvee\limits_{l\in L(\gamma)} l$ where $L(\gamma)$ is a trap derived from the interaction model γ.

We note that the above are particular choices of invariants we adopt; this means that the method is generic enough to work with other definitions of invariants as well, for example, for the interaction invariants, one could use linear invariants instead.

Proposition 2. *$CI(B_i)$, $II(\gamma)$ are inductive invariants of of $\|_\gamma B_i$.*

Let GI denote $II(\gamma) \wedge \bigwedge\limits_i CI(B_i)$. Making use of the fact that the conjunction of invariants is an invariant we can show that GI is also a global invariant, and furthermore, that it is inductive.

Proposition 3. *GI is an inductive invariant of $\|_\gamma B_i$.*

As for **system properties**, we are interested in safety properties which we denote by Φ. As an example, we consider the absence of deadlock. We say that a timed system with an interaction model γ is deadlocked when no interaction in γ is enabled. We denote such a property as *DIS*(γ), *DIS*$(\gamma) = \bigwedge\limits_{\alpha\in\gamma} \neg enabled(\alpha)$. Intuitively, a symbolic state $(l, \mathbf{v})$ is enabled if there exists an action successor of $(l, \mathbf{v} + \delta)$. Concretely, we use the enabledeness predicate as it has been defined by means of operations on polyhedra in [13]:

- $enabled(\alpha) = \swarrow (g \cap [r]\mathtt{tpc}(l'))$

where α is an interaction, and g, r, l' refer to a global transition $t = \big(l, (\_, g, \_), l'\big)$ corresponding to α and $[r]\zeta$ is the set of valuations $\mathbf{v}$ such that $\mathbf{v}[r]$ is in ζ.

Recall (VR) from introduction:

$$\frac{\vdash GI \rightarrow \Phi}{\|_\gamma B_i \models \Box\Phi} \ \text{(VR)}$$

Using Prop. 3 and Prop. 1 (VR) can be shown to be sound.

[3] To ease the reading, we abuse notation and use l as a place holder for a state predicate "$at(l)$" which holds in any symbolic state with location l, i.e., its semantics is given by $(l, \zeta) \models at(l)$.

3 A Method for Compositional TA Verification

In the introduction, we gave an intuition about why (VR) in its genericity is weak: the main problem is that the information about the relations between the values of local clocks at synchronisation time is not used. This is a consequence of the fact that the clocks advance at the same rate. The solution we propose consists in equipping components (and later, interactions) with *history clocks* for each port; then, at interaction time, the clocks corresponding to the ports participating in the interaction are reset; finally, new relations between the history clocks together with inequalities on history clocks automatically computed from γ strengthen GI.

3.1 Components with History Clocks

Definition 6 (Component with History Clocks)**.** *Given a component model* $B = (L, P, T, \mathcal{X}, \texttt{tpc})$*, its extension wrt history clocks is a timed automaton* $B^h = (L, P, T^h, \mathcal{X} \cup \mathcal{H}_P, \texttt{tpc})$ *where:*

- $\mathcal{H}_P$ *denotes the set of history clocks associated to ports,* $\mathcal{H}_P = \{h_p \mid p \in P\}$*;*
- $T^h = \big\{\big(l, (p, g, r \cup [h_p := 0]), l'\big) \mid \big(l, (p, g, r), l'\big) \in T\big\}$.

We note that there are no restriction on the initial values of the history clocks.

As an illustration, Figure 2 shows the extension with respect to history clocks of the components from the abstract example in the introduction. The extension preserves the symbolic states of the components, and consequently any invariant of the composition of B_i^h corresponds to an invariant of $\|_\gamma B_i$. For the ease of reading, we abuse notation and use $\exists\mathcal{H}_P$ to stand for $\exists h_{p_1} \exists h_{p_2} \ldots \exists h_{p_n}$ for $P = \{p_1, p_2, \ldots, p_n\}$.

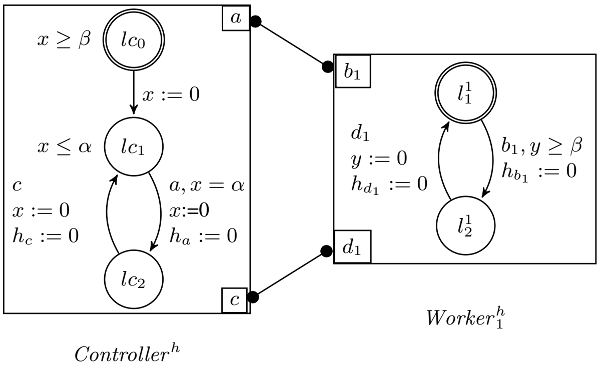

Figure 2: The Abstract Example with History Clocks.

Proposition 4. *Any symbolic execution in* B_i^h *corresponds to a symbolic execution (where all constraints on history clocks are ignored) in* B_i*.*

Corrolary 1. *If* $\|_\gamma B_i^h \models \Box I$ *then* $\|_\gamma B_i \models \Box(\exists\mathcal{H}_P).I$*.*

3.2 Inequalities for Histories

By construction, history clocks are reset when the corresponding ports participate in an interaction. Thus, all other clocks have greater values. This basic but useful observation we exploit in the following definition.

Definition 7 (Interaction Inequalities for History Clocks)**.** *Given an interaction model* γ*, we derive the following interaction inequalities* $\mathcal{E}(\gamma)$*:*

$$\mathcal{E}(\gamma) = \bigvee_{\alpha\in\gamma}\Big(\bigwedge_{p,q\in\alpha} h_p = h_q \leq min_{p'\in P(\gamma\ominus\alpha)} h_{p'} \wedge \mathcal{E}(\gamma\ominus\alpha)\Big)$$

where $\gamma \ominus \alpha = \{\beta \setminus \alpha \mid \beta \in \gamma \wedge \beta \setminus \alpha \neq \emptyset\}$ *and* $P(\gamma)$ *denotes the set of ports in* γ*.*

As an illustration, for the abstract example, $\mathcal{E}(\gamma) \triangleq (h_a = h_{b_1}) \wedge (h_c = h_{d_1})$.

Proposition 5. *$\mathcal{E}(\gamma)$ is an inductive invariant of $\|_\gamma B_i^h$.*

Due to the combination of recursion and disjunction, the formulae obtained by Definition 7 can be large. Much more compact formulae can be obtained by exploiting non-conflicting interactions in γ.

Proposition 6. *If $\gamma = \gamma_1 \cup \gamma_2$, where each γ_i is non-empty and their ports are disjoint, $P(\gamma_1) \cap P(\gamma_2) = \emptyset$, then $\mathcal{E}(\gamma) \equiv \mathcal{E}(\gamma_1) \wedge \mathcal{E}(\gamma_2)$.*

Corrolary 2. *If the interaction model γ has only disjoint interactions, i.e., for any $\alpha_1, \alpha_2 \in \gamma$, $\alpha_1 \cap \alpha_2 = \emptyset$, then $\mathcal{E}(\gamma) \equiv \bigwedge_{\alpha \in \gamma} \Big(\bigwedge_{p,q \in \alpha} h_p = h_q \Big)$.*

3.3 (VR) revisited

We propose to strengthen the global invariant GI as defined in Section 2.2 by replacing $CI(B_i)$ with $CI(B_i^h)$ and by adding $\mathcal{E}(\gamma)$. For ease of reference, we denote the new conjunction $II(\gamma) \wedge_i CI(B_i^h) \wedge \mathcal{E}(\gamma)$ as GI^h. As an illustration, for the abstract example, the invariant properties for the components with history are:

$$CI(Controller^h) = lc_0 \vee (lc_1 \wedge x \leq \alpha \wedge x = h_c \wedge x \leq h_a) \vee (lc_2 \wedge x = h_a \wedge h_c = h_a + \alpha)$$
$$CI(Worker^h) = (l_1 \wedge y \leq h_{b_1} \wedge y = h_{d_1}) \vee (l_2 \wedge h_{b_1} \leq h_{d_1} \wedge y \geq \beta + h_{b_1})$$

Together with the information from $\mathcal{E}(\gamma)$, the conjunction $GI^h \wedge \neg Safe$ reduces to false, i.e., if GI^h is a global inductive invariant then $Safe$ is also an invariant of the system.We conclude that the analysis with the auxiliary information derived with the help of history clocks is more precise in general.

We need to show the soundness of the new $(VR)^h$.

Theorem 1. *The rule $(VR)^h$:*

$$\frac{\vdash (\exists \mathcal{H}_P) GI^h \rightarrow \Phi}{\|_\gamma B_i \models \Box \Phi} \; (VR)^h$$

is sound.

The soundness follows from:

1. GI^h is an inductive invariant of $\|_\gamma B_i^h$ (by Prop. 3 and Prop. 5)
2. $(\exists \mathcal{H}_P) GI^h$ is an inductive invariant of $\|_\gamma B_i$ (by Corr. 1)
3. $\|_\gamma B_i \models \Box \Phi$ (from hypothesis $\vdash (\exists \mathcal{H}_P) GI^h \rightarrow \Phi$, together with item. 2 by Prop. 1)

4 Extension: handling conflicting interactions

In the previous section, we showed, on the abstract example, that by introducing history clocks, the calculated invariant approximates better the global reachable states set of the systems. No false alarm is detected. However, there are scenarios when the technique is weak. This is the case, for example, when interactions are conflicting. To illustrate the problem, we extend the abstract example by setting up N workers. For the ease of reading, let I denote the set $\{i \mid 1 \leq i \leq N\}$. There are N interactions conflicting in port a and N other interactions in port c, i.e., $\gamma = \{(a \mid b_i), (c \mid d_i) \mid i \in I\}$. The equation relating history clocks is:

$$\varepsilon(\gamma) \quad = \quad \bigvee_{k \in I} \Big(h_a = h_{b_k} \leq min_{i \in I \setminus \{k\}} h_{b_i} \Big) \wedge \bigvee_{k \in I} \Big(h_c = h_{d_k} \leq min_{i \in I \setminus \{k\}} h_{d_i} \Big).$$

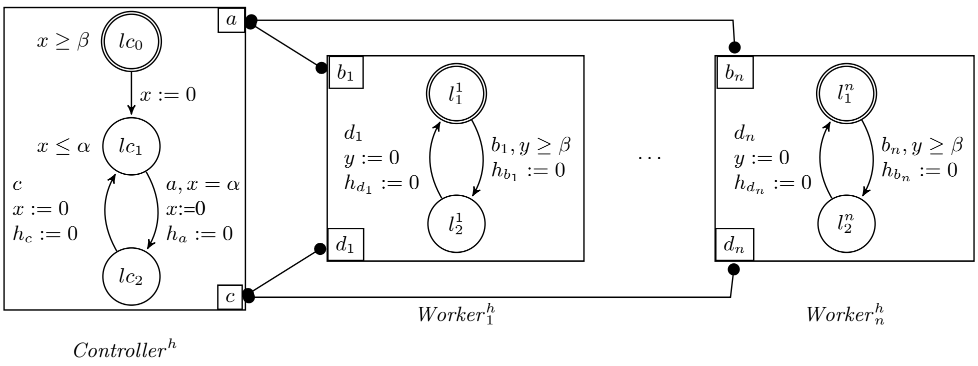

Figure 3: The Abstract Example with N Workers

In the following, we want to check if the system is deadlock-free. We point out that for a precise system, β and α are fixed. By varying those values, we study a family of systems in order to probe the accuracy of the method. We note that, for the system, there is a value β_{limit} such that for every $\beta > \beta_{limit}$, there is a real deadlock state. We may show that $\beta_{limit} = N \times \alpha$. In some way, when the controller attains lc_1 state, at least one worker i should have stayed at least $\beta - \alpha$ time in l_1^i. In fact, the workers are employed sequentially. When a worker makes its loop, all the others remain at $l_1^j, j \neq i$. When $\beta = \beta_{limit}$, the oldest worker in making a loop attains exactly $y_{oldest} = \beta_{limit}$ at l_1^{oldest}, when $x = \alpha$. If $\beta > \beta_{limit}$, its impossible that $y_{oldest} \geq \beta$. Transition b_{oldest} and, subsequently, all b_i are disabled whereas a is urgent ($x = \alpha$). This induces a real deadlock state.

$$\begin{cases} \beta \leq \beta_{limit} & \Rightarrow \quad \text{deadlock freedom} \\ \beta > \beta_{limit} & \Rightarrow \quad \text{real deadlock} \end{cases}$$

We show next that in such a scenario with conflicts the basic method returns false alarms in the case where the real execution is deadlock free, i.e., when $\beta \leq \beta_{limit}$. To do this, we consider the more general[4] safety property $\neg$*DIS*, where *DIS* is as follows:

$$\mathit{DIS} = \bigwedge_{i \in I} \left(\neg (lc_1 \wedge l_1^i \wedge x \leq \alpha \wedge y_i - x \geq \beta - \alpha) \wedge \neg (lc_2 \wedge l_2^i) \right)$$

We recall the subformulae in the invariants for the controller and for workers which are significant for the reasoning:

$$\begin{aligned} CI(Controller^h) &= lc_0 \vee \ (lc_1 \wedge h_c = x \leq h_a \wedge x \leq \alpha) \vee (lc_2 \wedge h_a = x \leq h_c) \\ CI(Worker_i^h) &= \left(l_1^i \wedge y_i = h_{d_i} \leq h_{b_i} \right) \vee \left(l_2^i \wedge y_i = h_{d_i} \geq h_{b_i} + \beta \right) \end{aligned}$$

We can show that for any N there exists a substitution (more precisely, infinitely many) θ which makes true both GI^h and *DIS*. For instance, when N is 3, θ is as follows:

$$\begin{aligned} \theta = \{ lc_1,\ l_1^1,\ l_2^1,\ l_3^1,\ y_3 = \alpha,\ x = \alpha,\ y_1 = y_2 = \frac{3\alpha}{2}, \\ (h_c = h_{d_3} = \alpha) \leq (h_{d_1} = h_{d_2} = \frac{3\alpha}{2}) \leq \\ (h_a = h_{b_3} = 2\alpha) \leq (h_{b_1} = h_{b_2} = 3\alpha) \} \end{aligned}$$

It can be shown that $GI^h\theta$ evaluates to true, i.e.:

$$\Big(CI(Controller^h) \bigwedge_{1 \leq i \leq 3} CI(Worker_i) \wedge II\,(\gamma) \wedge \varepsilon\,(\gamma) \Big) \theta \equiv \top$$

[4] We note that the property *Safe* a priori introduced is, in fact, a subformula of *DIS*.

Further, *DIS* is also satisfied and thus a false deadlock is detected because at l_1^1 we have that $y_3 - x = 0 < \beta - \alpha = \alpha$ and at l_1^i we have that $y_1 = y_2$ and $y_1 - x = \frac{\alpha}{2} < \beta - \alpha = \alpha$.

The above solution is the outcome of the following scenario: if $Worker_3$ has already executed one loop, then the sequence of interactions involved is $(b_3 \mid a)$, $(d_3 \mid c)$. However, with the corresponding equalities and inequalities of $\mathcal{E}(\gamma)$, with the local invariants and the interaction invariant, nothing forbids the following ordering:

$$h_c = h_{d_3} \leq min(h_{d_1}, h_{d_2}) \leq h_{b_3} = h_a \leq min(h_{b_1}, h_{b_2}) \tag{1}$$

The clocks order in Ineqs. (1) does not correspond to any real execution because transition d_2 or d_1 cannot occur between transitions b_3 and d_3. Such a relation cannot be deduced from the component invariants because there is no information about the time difference $(h_{d_2} - h_{d_3})$. As a general remark, d_i cannot occur between b_j and d_j, $j \neq i$. The values $\left|h_{d_j} - h_{d_i}\right|$ are, in fact, bounded below: if we consider the port c, we can check that at least α time units pass between two consecutive occurrences of c transition. We deduce that $\left|h_{d_j} - h_{d_i}\right| \geq \alpha$ for any $i \neq j$. It can be shown that by adding these differences, no false alarms are being raise, i.e., we obtain exactly the same results on deadlock as in the real execution: when a real deadlock exists ($\beta > \beta_{limit}$), we detect it and when the system is deadlock-free, there is no false alarm.

The above reasoning suggests a generic way to strengthen GI^h with information about the differences between the timings of the interactions themselves. To effectively implement this, we attach history clocks and corresponding resets to interactions at the system level:

Definition 8 (Interaction History Clock). *Given a system* $\|_\gamma B_i$*, its extension wrt history clocks for interactions is* $\|_{\gamma^h} B_i^h, B_\gamma$ *where:*

- B_γ *is an auxiliary TA having one location* l *with no invariant, and for each interaction* α *in* γ *a clock* h_α*, i.e.,* $B_\gamma = (\{l^*\}, P_\gamma, T, \mathcal{H}_\gamma, \emptyset)$ *where:*
  - *the set of ports* $P_\gamma = \{p_\alpha \mid \alpha \in \gamma\}$
  - *the set of clocks* $\mathcal{H}_\gamma = \{h_\alpha \mid \alpha \in \gamma\}$
  - $T = \{(l^*, p_\alpha, \top, h_\alpha := 0, l^*)\}$
- $\gamma^h = \{(p_\alpha \mid \alpha) \mid \alpha \in \gamma\}$ *with* $(p_\alpha \mid \alpha)$ *denoting* $\{p_\alpha\} \cup \{p \mid p \in \alpha\}$.

In a similar manner as in Section 3.2, it can be shown that any invariant of $B_\gamma \|_{\gamma^h} B_i^h$ corresponds to an invariant of $\|_\gamma B_i$ by first showing that any execution of $B_\gamma \|_{\gamma^h} B_i^h$ corresponds to an execution of $\|_\gamma B_i$. For the ease of reading, we abuse notation and use $\exists \mathcal{H}_\gamma$ to stand for $\exists h_{\alpha_1} \exists h_{\alpha_2} \ldots \exists h_{\alpha_n}$ for $\gamma = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Proposition 7. *Any execution in* $B_\gamma \|_{\gamma^h} B_i^h$ *corresponds to an execution in* $\|_\gamma B_i$.

Corrolary 3. *If* $B_\gamma \|_{\gamma^h} B_i^h \models \Box\, I$ *then* $\|_\gamma B_i \models \Box\, \exists \mathcal{H}_P \exists \mathcal{H}_\gamma . I$.

Recall that for component history clocks we added inequalities. We extend the def of $\mathcal{E}$ to talk about interaction history clocks.

Definition 9 ($\mathcal{E}^*$).

$$\mathcal{E}^*(\gamma) = \bigwedge_{p \in P} h_p = min_{\alpha \in \gamma_{|p}} h_\alpha.$$

where $\gamma_{|p} = \{\alpha \mid p \in P(\alpha)\}$ *and* δ_p *is the minimum time between two consecutive occurences of* p.

The next proposition makes the relation between $\mathcal{E}(\gamma)$ and $\mathcal{E}^*(\gamma)$ explicit.

Proposition 8. *The following equivalence holds:* $\exists \mathcal{H}_\gamma . \mathcal{E}^*(\gamma) \equiv \mathcal{E}(\gamma)$.

Next, we add "separations" for conflicting ports.

Definition 10 (Separation for interactions). *Given an interaction model γ and a conflicting port p, the induced separation, $\mathcal{S}(\gamma, p)$, is:*

$$\bigwedge_{\substack{\alpha,\beta \in \gamma_{|p} \\ \alpha \neq \beta}} \mid h_\alpha - h_\beta \mid \geq \delta_p.$$

where δ_p is the minimal time elapse between 2 consecutive executions of p in the "parent" component. Further, let $\mathcal{S}(\gamma)$ be $\bigwedge_{p \in P(\gamma)} \mathcal{S}(\gamma, p)$.

Hypothesis 1. *The initial values of the history clocks assoc. with interactions in B_γ is s.t. it satisfies $\mathcal{S}(\gamma)$. This is just a technical convention to simplify the proof.*

Next, we show that the new formulae are in fact inductive invariants.

Proposition 9. *Both $\mathcal{E}^*(\gamma)$, $\mathcal{S}(\gamma)$ are inductive invariants of $B_\gamma \|_{\gamma^h} B_i^h$.*

We strengthen GI^h to:

$$GI^* = II(\gamma) \wedge_i CI(B_i^h) \wedge \mathcal{E}^*(\gamma) \wedge \mathcal{S}(\gamma)$$

and consequently, the new $(VR)^*$ is:

$$\frac{\vdash (\exists \mathcal{H}_\gamma) GI^* \to \Phi}{\|_\gamma B_i \models \Box \Phi} \; (VR)^*$$

Similarly as it has been shown for the basic method in Section 3.3, the soundness of the new rule $(VR)^*$ follows from the fact that GI^* is a global inductive invariant of $\|_{\gamma^h} B_i^h, B_\gamma$. This is, indeed, the case because GI^* is a conjunction of invariants, which themselves are inductive.

Theorem 2. *The rule $(VR)^*$ is sound.*

Remark 1. *By Corollary 3 we have that $\exists \mathcal{H}_P \exists \mathcal{H}_\gamma . GI^*$ is an invariant of $\|_\gamma B_i$. To get some intuition about what information brings such an invariant, we consider an abstraction of the previous example. Let r be a port in a controller component and let r_i be the ports in worker components s.t. r_i interact (and thus conflict) on r. The subformula of GI^* which interests us is the conjunction of $\mathcal{E}^*$ and $\mathcal{S}$. We have:*

$$\mathcal{E}^*(\gamma) = \wedge_i h_{r_i} = h_{r_i|r} \wedge h_r = min_i(h_{r_i|r})$$
$$\mathcal{S}(\gamma) = \bigwedge_{\substack{i,j \\ i \neq j}} \mid h_{r_i|r} - h_{r_j|r} \mid \geq \delta_r$$

and consequently:

$$\exists \mathcal{H}_\gamma . \mathcal{E}^*(\gamma) \wedge \mathcal{S}(\gamma) \equiv \bigwedge_{\substack{i,j \\ i \neq j}} \mid h_{r_i} - h_{r_j} \mid \geq \delta_r .$$

Remark 2. *By definition, separations consider all possible combinations between interactions and this may lead to big formulae. We could, nevertheless, exploit the inherent symmetry in the scenario: real executions correspond to fixing a permutation of the interactions from the non-deterministic γ; we can show that there is an isomorphism between real executions (the controller maps to the controller, and i-th worker maps to j-th worker); thus, in particular scenarios as the one we considered, it is enough to show that the safety prop holds for one a priori chosen ordering.*

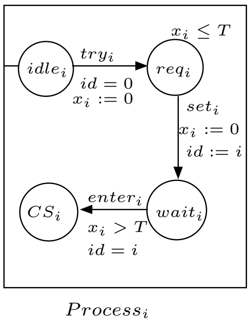

Figure 4: Fischer Protocol- Process automaton

5 Case Studies

5.1 Fischer protocol

A well-known example of real-time systems is the Fischer protocol for mutual exclusion. This example is well-studied in real-time verification context. The system consists of a number of processes sharing a resource. Two or more processes should not share the resource, thus be at the critical state at the same time. In the literature, each process is modeled as a timed automaton, and the assignment of the resource is decribed by a shared variable.

The concept behind Fischer protocol is that each process can affect his own identifier number to the global variable. After T time units, if the global variable is not equal to a different identifier number, the process can enter the critic state and use the resource. The waiting time is constrained by local clocks. Each process P_i with identifier number i has a local clock x_i .

To simplify, we deal with an acyclic version of the protocol, where a process that enters the critical state does not return to the request state again. We propose to check if in the first loop, two different processes can be present in the critical state, which would correspond to a false alarm.

Figure 4 describes the mutual exclusion process.

5.1.1 Fischer protocol model without global variable

To model Fischer protocol in our framework without resorting to the shared variable, we propose an additional component replacing it. The mutual exclusion between two processes is represented in Figure. 5 which contains, in addition, the corresponding history clocks. The figure shows the case of only two concurrent processes. For N processes, the interaction model is

$$\gamma = \big\{(enter_i|eq_i),\ (try_i|eq_0),\ (set_i^{process}, set_i^{id})\ \ i = 1\cdots N\big\}$$

The equations relating the history clocks are calculated using Eq. 7:

- Conflicting interactions on port eq_0 are: $xeq_0|xt_i, i = 1, N$

$$Eq_0 = \bigwedge_{i=1,N} (xeq_0 = xt_i \wedge (xeq_0 \leq xt_j), \forall j \neq i)$$

- There is no conflicting interactions on ports eq_i, set_i^{id}, $set_i^{process}$, and $enter_i$:

$$Eq_{eq_i} = (xeq_i = xen_i)$$
$$Eq_{set_i} = (xs_i = xsp_i)$$

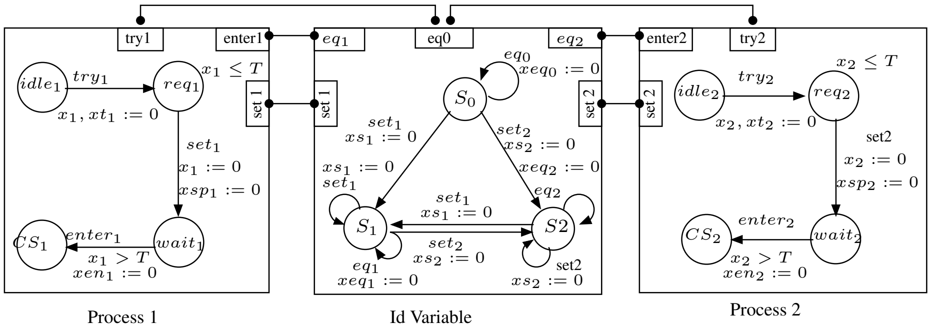

Figure 5: Fischer protocol modeled without the global variable

The equation which we finally derive from γ is

$$\mathcal{E}(\gamma) = Eq_0 \wedge (\bigwedge_{i=1,N} Eq_{eq_i}) \wedge (\bigwedge_{i=1,N} Eq_{set_i})$$

5.1.2 Results

The local invariants of the processes components and the local invariant of the global Variable component are joined to the intercation invariant in order to obtain the global system invariant. Without History clocks, it is equal to:

$$GI = (\bigwedge_{i=1,N} CI_{P_i}) \wedge CI_{Id-Variable}$$

where CI_{P_i} is the component invariant of process P_i and $CI_{Id-Variable}$ is the component invariant of the global variable. If history clocks are considered, the global invariant becomes:

$$GI^h = (\bigwedge_{i=1,N} CI^h_{P_i}) \wedge CI^h_{Id-Variable} \wedge \mathcal{E}(\gamma)$$

This compositional calculation of the global system invariant is used to detect the violation of the required safety property; two processes cannot be in the critical state at the same time:

$$SP = \forall i, j\, (CS_i \wedge CS_j \Rightarrow i = j)$$

The SAT- solver gives the following results:

$$GI \nRightarrow SP$$

$$GI^h \Rightarrow SP$$

We note that, in this case study, there is no use of the interaction invariant. The component invariants and the equality constraints between history clocks are sufficient. We deduce that the proposed method eliminates false alarms. It approximates sufficiently the global reachable states of the system, relatively to this safety property (SP).

5.2 Temperature Control Case Study

As a second case study, we adapt the temperature control example from [3]. There, a BIP model is described where the passing of time and the evolution of temperature are implemented by means of variables. Figure 5.2 shows a semantically equivalent RT-BIP model which replaces tick and temperature variables by clocks. The interaction model is given by $\gamma = \big\{(rest_i \mid heat), (cool_i \mid cool) \mid i \in \{0, 1\}\big\}$, thus the

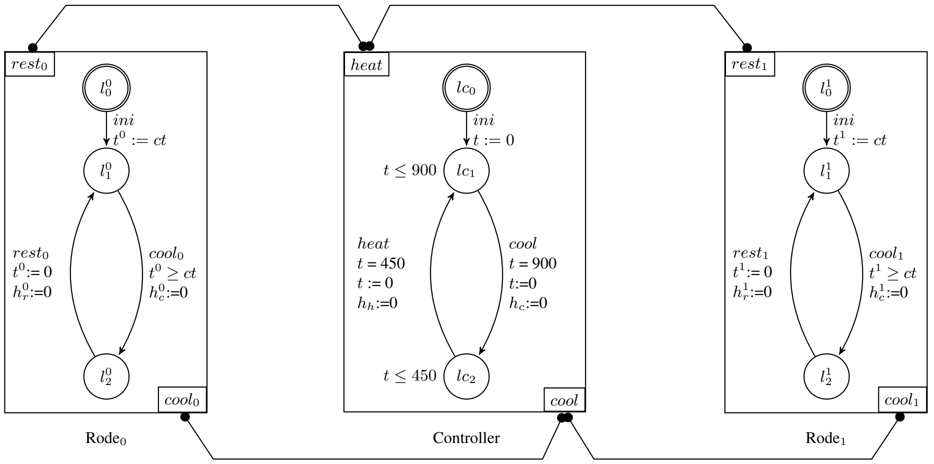

Figure 6: RT-BIP model of TC

corresponding equations we can derive from γ are:

$$\mathcal{E}(\gamma) = ((h_h = h_r^0 \leq h_r^1) \vee (h_h = h_r^1 \leq h_r^0)) \wedge ((h_c = h_c^0 \leq h_c^1) \vee (h_c = h_c^1 \leq h_c^0)).$$

The safety property we are interested in is the absence of deadlock:

$$DIS = \bigwedge_{i \in \{0,1\}} (\neg(l_{c_1} \wedge l_1^i \wedge t \leq 900 \wedge t^i - t \geq ct - 900) \wedge \neg(l_{c_2} \wedge l_2^i \wedge t \leq 450)).$$

For $ct \geq 1800$ we can check that $Controller \| Rode_0 \| Rode_1$ is deadlock free, i.e., $Controller \| Rode_0 \| Rode_1 \not\models \Box DIS$, however this is not the result after applying (VR) when we obtain false alarms:

$$\begin{aligned} CI(Rode_i^h) = {} & (l_1^i \wedge h_c^i \geq t^i = h_r^i \geq 0) \vee (l_1^i \wedge t^i \geq ct \wedge h_c^i = h_r^i \geq t^i - ct) \vee \\ & (l_2^i \wedge t^i - ct \geq h_c^i \geq 0 \wedge h_r^i = t^i) \vee (l_2^i \wedge t^i - ct \geq h_c^i \geq 0 \wedge h_r^i \geq t^i - ct) \\ CI(Controller^h) = {} & (lc_1 \wedge h_h \geq h_c - 450 = t \in [0,900]) \vee (lc_1 \wedge h_h = h_c \geq t \in [0,900]) \vee \\ & (lc_2 \wedge h_c = t \in [0,450] \wedge h_h \geq 900 - t) \vee (lc_2 \wedge h_c = t \in [0,450] \wedge h_h \geq ct - t) \\ \Rightarrow & \\ \exists \theta.(CI(Controller^h) & \wedge CI(Rode_1^h) \wedge CI(Rode_2^h) \wedge II(\gamma) \wedge \mathcal{E}(\gamma) \wedge \boldsymbol{DIS})\theta \equiv \top \end{aligned}$$

where a solution θ is, for example:

$$\theta = \{lc_1, l_1^1, l_1^0, t = 900, t^1 = 901, t^0 = 900, (h_h = h_r^0 = 900) \leq (h_r^1 = 901) \leq (h_c = h_c^0 = 1350) \leq (h_c^1 = 1351)\}$$

which is s.t. it satisfies each $CI(Rode_i^h)$, $CI(Controller^h)$, $\mathcal{E}(\gamma)$, and $II(\gamma)$, *and DIS* because at l_1^i $t^i - t < ct - 900$. This solution is the outcome of the following hypothetical scenario: assume $Rode_0$ has already executed one loop, i.e., the sequence of interactions observed so far is: $(cool_0 \mid cool), (rest_0 \mid heat)$, with the corresponding inequalities in $\mathcal{E}(\gamma)$ being $h_h = h_r^0 \leq h_r^1 \wedge h_c = h_c^0 \leq h_c^1$ and clock constraints:

- $h_h = t = h_c - 450 \in [0,900]$
- $h_c^0 \geq h_r^0 = t^0$
- $h_c^1 \geq h_r^1 = t^1$

and thus nothing forbids an arrangement like:

$$h_h = h_r^0 \leq h_r^1 \leq h_c = h_c^0 \leq h_c^1 \tag{2}$$

which does not, in fact, correspond to any real execution ($rest_1$ cannot be executed between $cool_0$ and $rest_0$). Ineqs. (2) is possible precisely because there is no information about the time difference $h_r^1 - h_r^0$. To synthesise this info, as Marius suggested, remove a conflict, e.g., $heat \mid rest_i$, by "splitting" $heat$ into $heat_i$ s.t. $heat_i \mid rest_i$:

$$CI(Controller^*) = \cdots \wedge (lc_1 \wedge t = h_{h_0} = h_c - 450 \in [0, 900] \wedge \tag{3}$$
$$h_{h_1} \geq t + 1350) \tag{4}$$

this provides enough info to forbid Ineqs. (2):

$$h_r^1 \geq h_r^0 + 1350 \text{ using } (3), h_r^1 = h_{h_1}, h_r^0 = h_{h_0} \text{ from } \mathcal{E}(\gamma)$$
$$h_c \leq 1350 \text{ from } (4)$$
$$1350 + h_r^0 \leq h_r^1 \leq h_c \leq 1350 \text{ using } (1), (4), (5) \tag{5}$$

Ineqs. (5) lead to a contradiction.

The above information can be automatically obtained from the conjunction of $\mathcal{E}^*$ and $\mathcal{S}$ introduced in Section 4:

$$\mathcal{E}^*(\gamma) = \bigwedge_{i \in \{0,1\}} (h_r^i = h_{r_i|h} \wedge h_c^i = h_{c_i|c}) \wedge h_h = min_i(h_{r_i|h}) \wedge h_c = min_i(h_{c_i|c})$$
$$\mathcal{S}(\gamma) = \mid h_{r_1|h} - h_{r_0|h} \mid \geq \delta_h \wedge \mid h_{c_1|c} - h_{c_0|c} \mid \geq \delta_c$$

and thus, by eliminating the quantifiers, $\exists \mathcal{H}_\gamma . \mathcal{E}^*(\gamma) \wedge \mathcal{S}(\gamma)$ is equivalent to:

$$\mid h_r^1 - h_r^0 \mid \geq 1350 \wedge \mid h_c^1 - h_c^0 \mid \geq 1350$$

by using that the time elapse between consecutive *cool* and resp. *heat* is 1350.

5.3 Evaluation

Size (nb rodes)	D^t-Finder	D^t-Finder with Separations (GI*)	Uppaal
2 - 7	cex	true	true
> 7	cex	true	-

Table 1: Comparison between D^t-Finder and Uppaal

In Table 1, $II(\gamma)$ is the linear interaction invariant: $lc_0 + lc_1 + \sum_0^{n-1} lr_2^i = 1$ and GI^* stands for:

$$CI(Controller^*) \wedge_i CI(Rode_i^h) \wedge II(\gamma) \wedge \mathcal{E}(\gamma) \wedge_i h_r^{\pi(i)} - h_r^{\pi(i-1)} \geq 1350.$$

6 Related Work

Formal verification of timed systems encounters state -space-explosion problems, mainly when it comes to timed systems. To formally address this issue, the assume guarantee [9, 8, 11] approach was proposed in order to to deduce global properties of the system based on features of the separate composing subsystems. However, some assumptions should be made and it is a challenge to find the appropriate decomposition and the related assumptions[6]. Yet, it is challenging to offer automated techniques supporting this pattern.

Attempting to reduce the state space explosion, authors in [7] precede composition with an abstraction module. Composition is applied to the abstracted timed automata. Finding a safe abstraction condition makes the method restricted to fully deterministic automata. Compositional logic has also been conducted as part of timed interface theory [1]. It permits to verify if two interfaces are compatible and shows a method to compose them. This framework differs from ours in that we try to automatically calculate a global invariant of the composite system, permitting to approach the global reachable state and check satisfaction of different properties.

The idea of adding new local clocks to automata was proposed in [2], trying to alleviate the reachability checking of timed systems. The idea is to desynchronize local clocks and minimize the exploration of interleaving and independant transitions. Then, resynchronization is carried out through added reference clocks, one in each automaton. These clocks measure the local time that has advanced in each automaton since the start time whereas our history clocks indicate the time that has advanced from every interaction. Besides, we propose to apply our idea to a composition rather than exploration framework.

7 Conclusion and Future Work

Although theoretical methods have been introduced for compositional reasoning, mainly the assume-guarantee approach, they are still unpractical to implement due to the lack of automatism. In this paper, we presented a fully automated technique to generate compositionally global invariants of timed systems and have shown its soundness on several case studies. In the future, we intend to manage data invariants issue, which characterizes typically scheduling scenarios. The proposed method could also be extended to statistical model checking of probabilistic timed systems.

References

[1] L. D. Alfaro, T. A. Henzinger, and M. Stoelinga. Timed interfaces, 2002. 6

[2] J. Bengtsson, B. Jonsson, J. Lilius, and W. Yi. Partial order reductions for timed systems, 1998. 6

[3] S. Bensalem, M. Bozga, J. Sifakis, and T.-H. Nguyen. Compositional verification for component-based systems and application. In *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis*, ATVA '08, pages 64–79, Berlin, Heidelberg, 2008. Springer-Verlag. (document), 1, 1, 5.2

[4] S. Bornot and J. Sifakis. An algebraic framework for urgency. *Information and Computation*, 163:2000, 1998. 2

[5] B. Dutertre and L. de Moura. The Yices SMT solver. Technical report, SRI International, 2006. 1

[6] G. S. A. Jamieson M. Cobleigh and L. A. Clarke. Breaking up is hard to do: An evaluation of automated assume-guarantee reasoning. 2008. 6

[7] H. E. Jensen, K. G. Larsen, and A. Skou. Scaling up uppaal automatic verification of real-time systems using compositionality and abstraction. In *FTRTFT*, pages 19–30, 2000. 6

[8] C. B. Jones. Specification and design of (parallel) programs. pages 321–332, 1983. 6

[9] J. Misra and K. M. Chandy. Proofs of networks of processes. page 4:417–426, 1981. 6

[10] L. Moura and N. Bjørner. Efficient e-matching for smt solvers. In *Proceedings of the 21st international conference on Automated Deduction: Automated Deduction*, CADE-21, pages 183–198, Berlin, Heidelberg, 2007. Springer-Verlag. 1

[11] A. Pnueli. In transition from global to modular temporal reasoning about programs. page 123–144, 1984. 6

[12] S. Tripakis. *The analysis of timed systems in practice*. PhD thesis, Joseph Fourier University, 1998. 2.1, 2.1

[13] S. Tripakis. Verifying progress in timed systems. In *In ARTS'99*, pages 299–314. Springer-Verlag, 1999. 2.2

A Proofs

Calculation of DIS The equation:

$$\mathit{enabled}(\alpha) = \{(l, \mathbf{v}) \mid (\exists l')(\exists \delta \geq 0).\mathsf{tpc}(l)(\mathbf{v}+\delta) \wedge (l, \mathbf{v}+\delta) \xrightarrow{\alpha} (l', \mathbf{v}+\delta)\} \tag{6}$$

is equivalent to:

$$\mathit{enabled}(\alpha) = \swarrow (g \cap [r]\mathsf{tpc}(l')). \tag{7}$$

Proof.

$\mathit{enabled}(\alpha) = \{(l, \mathbf{v}) \mid (\exists l')(\exists \delta \geq 0).\mathsf{tpc}(l)(\mathbf{v}+\delta) \wedge (l, \mathbf{v}+\delta) \xrightarrow{\alpha} (l', \mathbf{v}+\delta)\}$
$\mathit{Eq.}$ (6) $\equiv$
(replacing the trans by its cond and abstracting away the info about locs)
$\{\mathbf{v} \mid (\exists \delta \geq 0).(g \cap [r]\mathsf{tpc}(l'))(\mathbf{v}+\delta) \wedge (\forall 0 \leq \delta' < \delta).\big(\mathsf{tpc}(l)(\mathbf{v}+\delta')\big)\} \equiv$
(using Lemma 3)
$\{\mathbf{v} \mid (\exists \delta \geq 0).(g \cap [r]\mathsf{tpc}(l'))(\mathbf{v}+\delta) \wedge \big(\mathsf{tpc}(l)(\mathbf{v}+\delta)\big)\} \equiv$
$\swarrow \big(g \cap [r]\mathsf{tpc}(l') \cap \mathsf{tpc}(l)\big) \equiv$
(using $g \subseteq \mathsf{tpc}(l)$)
$\mathit{enabled}(\alpha) = \swarrow (g \cap [r]\mathsf{tpc}(l'))$
$\mathit{Eq.}$ (7) (for a global transition $t = \big(l, (\_, g, \_), l'\big)$ corresponding to α)

□

Lemma 3. *If ζ is closed convex and $\zeta(\mathbf{v})$ then* $(\forall 0 \leq \delta' < \delta).\zeta(\mathbf{v}+\delta') \equiv \zeta(\mathbf{v}+\delta)$.

Proof. To ease the reading, we adopt the notation $\mathbf{v} \in \zeta$ instead of $\zeta(\mathbf{v})$.
"$\Rightarrow$":

$\mathbf{v} \in \zeta \wedge (\forall 0 \leq \delta' < \delta).\zeta(\mathbf{v}+\delta') \Rightarrow$
(by choosing $\delta_n = \delta - \dfrac{\delta}{n}$ and using that ζ is closed)
$(\forall n \geq 1).(\mathbf{v}+\delta_n \in \zeta) \Rightarrow \lim\limits_{n \to +\infty}(\mathbf{v}+\delta_n) \in \zeta \Rightarrow \mathbf{v}+\delta \in \zeta$

"$\Leftarrow$":

$\mathbf{v} \in \zeta \wedge (\mathbf{v}+\delta \in \zeta) \Rightarrow$
(by choosing $\mathbf{v}_1 = \mathbf{v}+\delta, \mathbf{v}_2 = \mathbf{v}$ and using that ζ is convex)
$(\forall \lambda \in [0,1)).\big(\lambda(\mathbf{v}+\delta) + (1-\lambda)\mathbf{v} \in \zeta\big) \equiv$
$(\forall \lambda \in [0,1)).(\mathbf{v}+\lambda\delta \in \zeta) \equiv (\forall 0 \leq \delta' < \delta).(\mathbf{v}+\delta' \in \zeta)$

□

Proof of Proposition 4. Let X be the set of clocks in B_i and recall that $\zeta_{|X}$ is the zone containing only the constraints in ζ which have variables in X, where $|_X$ is the zone operator for projection on X. It suffices to note that any symbolic state (l, ζ^h) in the reachability set $\mathit{Reach}(s_0^h)$ of B_i^h with initial state $s_0 = (l_0, \zeta_0^h)$ is equivalent with (up to $\mathcal{H}_P$) a symbolic state $(l, \zeta^h_{|X})$ in the reachablity set of B_i, $\mathit{Reach}(s_0)$, with initial state $s_0 = (l, \zeta^h_{0_{|X}})$. □

Proof of Proposition 5. By induction on the length of global execution paths. It suffices to recall that when an interaction α takes place at a global state s, all h_p with $p \in \alpha$ are reset to 0, thus their value at any successor of s are smaller than any h_q, with $q \in P(\gamma) \setminus \alpha$, and consequently smaller than the minimum among h_q. Also, the values of the history clocks not being reset are unchanged, thus satisfy $\mathcal{E}(\gamma \ominus \alpha)$ by induction. □

Proof of Proposition 6. By induction on the number of interactions in γ. In the base case, γ has 2 interaction, each γ_i consists of precisely one interaction α_i.

$$\begin{aligned}
\mathcal{E}(\gamma) = \Big(\bigwedge_{i,j} h_{\alpha_1(i)} = h_{\alpha_1(j)} \wedge h_{\alpha_1(0)} \leq min_k(h_{\alpha_2(k)}) \wedge \mathcal{E}(\{\alpha_2\})\Big) \qquad \vee \\
\Big(\bigwedge_{i,j} h_{\alpha_2(i)} = h_{\alpha_2(j)} \wedge h_{\alpha_2(0)} \leq min_k(h_{\alpha_1(k)}) \wedge \mathcal{E}(\{\alpha_1\})\Big) \qquad \equiv \\
\Big(\text{using } \mathcal{E}(\{\alpha_i\}) \stackrel{\triangle}{=} \Big(\bigwedge_{i,j} h_{\alpha_i(i)} = h_{\alpha_i(j)}\Big)\Big) \\
\mathcal{E}(\{\alpha_1\}) \wedge \mathcal{E}(\{\alpha_2\}) \wedge \big(h_{\alpha_1(0)} \leq min_k(h_{\alpha_2(k)}) \vee h_{\alpha_2(0)} \leq min_k(h_{\alpha_1(k)})\big) \qquad \equiv \\
(\text{using totality of } \leq, h_{\alpha_1}(0) \geq h_{\alpha_2}(0) \vee h_{\alpha_2}(0) \geq h_{\alpha_1}(0)\ ) \\
\mathcal{E}(\gamma_1) \wedge \mathcal{E}(\gamma_2)
\end{aligned}$$

where we used $\alpha(i)$ to denote the i-th port in α.
"P(n) ⇒ P(n+1)": for the ease of reading, we introduce $eq(\alpha)$ and $leq(\alpha)$ to denote $\bigwedge_{i \neq j} h_\alpha(i) = h_\alpha(j)$ and respectively $h_\alpha(0) \leq min_{\beta \neq \alpha k} h_\beta(k)$.

$$\begin{aligned}
\mathcal{E}(\gamma) = \bigvee_{\alpha \in \gamma_1} eq(\alpha) \wedge leq(\alpha) \wedge \mathcal{E}((\gamma_1 \cup \gamma_2) \ominus \alpha) \vee \bigvee_{\alpha \in \gamma_2} eq(\alpha) \wedge leq(\alpha) \wedge \mathcal{E}((\gamma_1 \cup \gamma_2) \ominus \alpha) \qquad \equiv \\
\big(\text{using } \gamma_2 \ominus \alpha = \gamma_2 \text{ for } \alpha \in \gamma_1 (\text{ resp. for } \gamma_2)\text{and by ind. for } \gamma' = (\gamma_1 \ominus \alpha) \cup \gamma_2\big) \\
\mathcal{E}(\gamma_2) \wedge \big(\bigvee_{\alpha \in \gamma_1} eq(\alpha) \wedge leq(\alpha) \wedge \mathcal{E}(\gamma_1 \ominus \alpha)\big) \vee \mathcal{E}(\gamma_1) \wedge \big(\bigvee_{\alpha \in \gamma_2} eq(\alpha) \wedge leq(\alpha) \wedge \mathcal{E}(\gamma_2 \ominus \alpha)\big) \qquad \equiv \\
\big(\text{using } eq(\alpha) \wedge \mathcal{E}(\gamma_1 \ominus \alpha) = \mathcal{E}(\gamma_1) \text{ (and resp. for } \gamma_2\text{) by ind.}\big) \\
\mathcal{E}(\gamma_1) \wedge \mathcal{E}(\gamma_2) \wedge \big(\bigvee_{\alpha \in \gamma_1} leq(\alpha) \vee \bigvee_{\alpha \in \gamma_2} leq(\alpha)\big) \qquad \equiv \\
\mathcal{E}(\gamma_1) \wedge \mathcal{E}(\gamma_2) \wedge \bigvee_{\alpha \in \gamma} leq(\alpha) \qquad \equiv \\
(\text{ using totality of } \leq \text{ and disjointness of } \gamma_i) \\
\mathcal{E}(\gamma_1) \wedge \mathcal{E}(\gamma_2)
\end{aligned}$$

□

Proof of Proposition 7. The reasoning is similar to the one in the proof of Proposition 4. It suffices to note that any reachable state $(\bar{l}, \zeta^h)$ in $\|_{\gamma^h} B_i^h, B_\gamma$ corresponds to a reachable state $(\bar{l} \setminus l^*, \zeta^h_{|X})$ in $\|_\gamma B_i$ where we recall that l^* is the unique location in B_γ and X is the set of clocks in $\|_\gamma B_i$. □

Proof of Proposition 8. By induction on the number of interaction in γ.
In the base case, $\gamma = \{\alpha\}$, we have the following equivalences:

$$\begin{aligned}
\mathcal{E}(\gamma) = \bigwedge_{p,q \in \alpha} h_p = h_q \equiv \exists h_\alpha.\big(\bigwedge_{p \in \alpha} h_p = h_\alpha\big) \equiv \\
\exists \mathcal{H}_\gamma.\big(\bigwedge_{p \in P(\gamma)} h_p = min_{\alpha \in \gamma_{|p}} h_\alpha\big) \equiv \exists \mathcal{H}_\gamma.\mathcal{E}^*(\gamma).
\end{aligned}$$

In the inductive case, we assume that $\exists \mathcal{H}_\gamma . \mathcal{E}^*(\gamma) \equiv \mathcal{E}(\gamma)$ holds for any γ of size smaller than k and we show that it also holds for a γ of size $k+1$. To do this, we fix γ as the set $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, an arbitrary α as α_1. Further, we denote $\alpha_i^* = \alpha_i \setminus \alpha$, for any $i > 1$ and $\gamma' = \gamma \ominus \alpha$, that is, $\gamma' = \{\alpha_2^*, \ldots, \alpha_n^*\}$. Clearly, the size of γ' is less than n. We have the following equivalences:

$\mathcal{E}(\gamma) \equiv$ (assumming it is α the interaction for which the corr. conj makes $\mathcal{E}(\gamma)$ true)

$$\bigwedge_{p,q \in \alpha} h_p = h_q \leq min_{r \in P(\gamma')} h_r \wedge \mathcal{E}(\gamma') \equiv$$

(by the induction hypothesis)

$$\bigwedge_{p,q \in \alpha} h_p = h_q \leq min_{r \in P(\gamma')} h_r \wedge \exists \mathcal{H}_{\gamma'} . \mathcal{E}^*(\gamma') \equiv$$

$$\exists h_\alpha . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha \leq min_{r \in P(\gamma')} h_r \big) \wedge \exists \mathcal{H}_{\gamma'} . \mathcal{E}^*(\gamma') \equiv$$

$$\exists h_\alpha \exists \mathcal{H}_{\gamma'} . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha \leq min_{r \in P(\gamma')} h_r \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma_{|r}} h_\beta \big) \equiv$$

$$\exists h_\alpha \exists \mathcal{H}_{\gamma'} . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha \leq min_{r \in P(\gamma')} min_{\beta \in \gamma'_{|r}} h_\beta \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma'_{|r}} h_\beta \big) \equiv$$

(h_α is $\leq$ than any clock h_β for any β containing an arbitrary r, so it is the min among all $\beta \in \gamma'$)

$$\exists h_\alpha \exists \mathcal{H}_{\gamma'} . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha = min_{\beta \in \gamma'} h_\beta \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma'_{|r}} h_\beta \big) \equiv$$

(by introducing new vars h_{α_i} and corr. eqs)

$$\exists \mathcal{H}_\gamma \exists \mathcal{H}_{\gamma'} . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha = min_{\beta \in \gamma'} h_\beta \wedge \bigwedge_{i \in \{2, \ldots, n\}} h_{\alpha_i^*} = h_{\alpha_i} \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma'_{|r}} h_\beta \big) \equiv$$

(any β in γ' corresponds to a α_i)

$$\exists \mathcal{H}_\gamma . \big(\bigwedge_{p \in \alpha} h_p = h_\alpha \wedge h_\alpha = min_{\beta \in \gamma} h_\beta \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma_{|r}} h_\beta \big) \equiv$$

$$\exists \mathcal{H}_\gamma . \big(\bigwedge_{p \in \alpha} h_p = min_{\beta \in \gamma} h_\beta \wedge \bigwedge_{r \in P(\gamma')} h_r = min_{\beta \in \gamma_{|r}} h_\beta \big) \equiv$$

(using $P(\gamma) = P(\gamma') \cup \alpha$)

$$\exists \mathcal{H}_\gamma . \big(\bigwedge_{r \in P(\gamma)} h_r = min_{\beta \in \gamma_{|r}} h_\beta \big) \equiv \exists \mathcal{H}_\gamma . \mathcal{E}^*(\gamma)$$

□

Proof of Proposition 9. By induction on the length of computations. The base case follows by Hypothesis 1. For the inductive case, let $s = (\bar{l}, \zeta)$ be the state reached after i steps, α be an interaction which can be executed from s, $s' = (\bar{l}', \zeta')$ be the successor state, and p be an arbitrary port. We make a case analysis depending on wether $p \in \alpha$.

1. $p \in \alpha$: then both h_p and h_α have been reset. Consequently, on the one hand, their values at s' are such that $h_p = h_\alpha = min_{\beta \in \gamma_{|p}}$, on the other hand, for any h_β, $\mid h_\beta - h_\alpha \mid$ evaluates to the value of h_β at s and is thus greater or equal than δ_p by induction. The difference $\mid h_\beta - h_{\beta'} \mid$ is preserved for any $\beta' \neq \alpha$, thus, greater or equal than δ_p by induction.

2. $p \notin \alpha$: then it suffices to note that $\alpha \notin \gamma_{|p}$ and thus both $h_p = min_{\beta \in \gamma_{|p}} h_\beta$ and $\mid h_\beta - h_{\beta'} \mid \geq \delta_p$ are preserved from s to s' and hold by induction.

□