Results for compositional timed testing

Boutheina Bannour  
CEA LIST  
Point Courrier 174, 91191  
Gif-sur-Yvette, France  
boutheina.bannour@cea.fr

Christophe Gaston  
CEA LIST  
Point Courrier 174, 91191  
Gif-sur-Yvette, France  
christophe.gaston@cea.fr

Marc Aiguier  
Laboratoire MAS  
Grande Voie des Vignes, 92195 Châtenay-Malabry, France  
marc.aiguier@ecp.fr

Arnault Lapitre  
CEA LIST  
Point Courrier 174, 91191  
Gif-sur-Yvette, France  
arnauld.lapitre@cea.fr

Abstract—Modern industrial systems are often large and distributed. Consequently, building the test harness for them can be technically challenging. A compositional approach attempts to overcome this problem by partitioning the system into smaller parts easier to test separately. And in particular, compositionality helps to avoid as much as possible testing the whole monolithic system thanks to mathematical results which relate the global correctness of the system to the correctness of its constituent parts. In this paper, we present a compositionality result for model-based testing in the setting of the conformance relation tioco which is dedicated to timed systems. We show how to exploit this result in practice by extending a previously defined symbolic testing framework.

Keywords—Model-based testing, timed input output labeled transition systems, conformance relation tioco, compositionality, component-based system, symbolic execution, projections, real-time systems.

I. INTRODUCTION

Building test harness for systems composed of a large number of interconnected units can be a difficult task. This may be due to the fact that monitoring exchanges between subsystems requires too much technological efforts. This situation can typically occur when systems are distributed. The lack of internal instrumentation can prevent some faults from being discovered. In order to have a partial observation of such internal behaviors, one may try to define test cases whose execution results, even though observed only at the system interface level, are informative about those internal behaviors. However, this requires the system to be sufficiently controllable which is often not the case, especially of distributed or asynchronous systems. Moreover, once a fault is observed at the testing interface level, further observability capabilities are often needed to identify which subsystem(s) caused the fault. Another technical limitation is related to side-effecting instrumentation which can be complex enough to modify the system run-time performance. Such influence questions the quality of the observation and hence the accuracy of the testing process.

In order to tackle those issues, compositionality results may be useful. In a compositional framework, a system is seen as an assembly of interacting units whose correctness implies the correctness of the whole system by construction. Thanks to such a result, testing the whole system comes mainly to testing its units. At the system level, one only has to check that the system is assembled in a way that reflects the structuring operators considered in the compositionality result. When testing is used for that purpose, this is the most simple form of what is called integration testing. In this paper, we study such a compositionality result in a model-based testing framework dedicated to timed systems [8], [17], [22]. Models are particular automata, named Timed Input Output Labeled Transition systems (TIOLTS) whose transitions are labeled by inputs, outputs or durations. In this framework, the correctness of a system is modeled by a so-called conformance relation named tioco. TIOLTS can be composed to constitute models of systems. For such system models, we show how to extract the set of possible executions of its units in the context of the system. For each unit, it is characterized, thanks to projection mechanisms, as a TIOLTS whose set of traces is a subset of the one characterized by the stand-alone TIOLTS that models the unit. Our projection techniques are based on those that we introduced in [13] and latter adapted to deal with time in [4], [5]. We show that under a certain hypothesis, if each unit conforms to its TIOLTS obtained by projection then the composition of unit models conforms to the system one. As usual in testing, such a result is more exploitable in its converse expression: if a composition of units does not conform to a system model, then there is at least one unit that does not conform to its TIOLTS obtained by projection. Therefore assuming that the units are correctly composed, any potential fault of the system can be identified by testing its units at the unitary level. The TIOLTS framework is adequate for theoretical reasoning, however as soon as one wants to really model systems, it is not expressive enough. Indeed, it is classical to reason about durations in terms of first-order constraints, which is impossible with TIOLTS. In previous papers [3], [12], we presented a symbolic framework called Timed Input Output Symbolic Transition Systems (TIOSTS) to deal with such constraints, and more generally with data in a symbolic manner. We defined testing algorithms based on TIOSTS. In this paper, we discuss how we can take benefits of our compositionality result while using those algorithm to test systems. This requires to transpose the notion of projection to the symbolic context and to define an algorithm for assessing the satisfaction of the compositionality result hypothesis.

The remaining of the paper is structured as follow: Section II gives preliminaries about typed equational logic and the behavioral model that we will use; Section III presents the timed conformance testing framework which grounds our result; Section IV establishes the compositionality result; Section V shows how to deal with the compositionality result using symbolic techniques; Section VI reviews the state of the art relevant to our contribution; Finally, we conclude and give ongoing and future work in Section VII.
II. Preliminaries

Abstract data type [25]. In order to symbolically denote data, we use a many-sorted first-order logic. A data signature Σ is a couple (S, F, P) where S is a set of sorts and, F and P are respectively sets of function and predicate names with arities in S. For any set of many-sorted variables V = (Vs)šeS, the set of Σ-terms, noted TΣ(V), and Σ-formulas, noted SenΣ(V), are inductively defined as usual. The mathematical interpretation of any signature Σ = (S, F, P) is given by a Σ-model M = (Ms, vM, dM), called a Σ-model, provided with a total function ƒM : Ms × · · · × Msn → Ms for each function name f : s1 . . . sn → s ∈ F and a -ary predicate pM : Ms × · · · × Msn for each predicate name p : s1 . . . sn ∈ P. The evaluation of Σ-terms from a Σ-model M is given by any total function vM : TΣ( ) → M defined as the canonical extension of any interpretation of variables v : V → M. Therefore, we extend any interpretation v over an unary relation M |=v Ω on Ω-formulas as usual. We denote Mv the set of mappings from V to |M|.

LTS [6]. A Labeled Transition Systems (LTS) over a set of labels L is a triple (Q, q0, tr) where Q is a set of states, q0 ∈ Q is the initial state, and Tr ⊆ Q × L × {τ} × Q is a set of transitions. τ is called the internal action. For any transition tr of the form (q, α, q′), source(tr), target(tr) and act(tr) stand for q, q′ and α respectively. Path(α) ⊆ Tr is the set of paths of the LTS and which contains the empty sequence and all sequences t1 · · · tn with n ≥ 1 and such that source(t1) = q0 and for all i, 1 < i ≤ n, source(t1i) = target(t1i−1). For any p in Path(α), the trace of p, denoted trace(p), is act(tr1) · · · act(trn) after filtering τ actions. Traces(α) is the set of traces of all paths in Path(α).

For any two LTS A1 = (Q1, q0, r1) and A2 = (Q2, q0, q2, t2) respectively over L1 and L2, the composition of A1 and A2, denoted A1 || A2, is the LTS (Q, q0, r) over L1 ∪ L2 such that Q = Q1 × Q2, the initial state q0 is (q0, q2) and r is the set whose transitions are defined according to one of the following rules:

Synchronous execution: If (q1, α, q1′) ∈ r1 and (q2, α, q2′) ∈ r2 and α ̸= τ, then ((q1, q2), α, (q1′, q2′)) ∈ r;

Asynchronous execution: For any (q1, α, q1′) ∈ r1 where α ̸∈ L2 or α = τ, then for any q2 ∈ Q2, ((q1, q2), α, (q1′, q2′)) ∈ r; The role of A1 and A2 can be inverted.

In the sequel, we will need to hide actions in a TIO LTS. For any LTS G = (Q, q0, r) over L and for any L′ ⊆ L, we note Hide(L′, G) the LTS (Q, q0, r′) where r′ is such that for any r ∈ r, if act(r) ̸∈ L′ then r ∈ r′, and if act(r) ∈ L′ then (source(r), τ, target(r)) ∈ r′. The hiding operation can obviously be defined at the level of traces. Trivially for any LTS A over L and for any L′ ⊆ L, we have Traces(Hide(L′, A)) = Traces(Hide(L′, Traces(A))).

III. Testing framework for TIO LTS

We are interested in modeling real time with a particular emphasis on the notion of duration. As in [3], [10], [17], [22], we propose to use a set D, also called durations, which is isomorphic to the set of strictly positive real numbers. In this denotation of durations, we abstract away from concrete units (second, millisecond, etc.) and just talk about "unit of time" as a theoretical unit to be put in correspondence with real units at the implementation level. We now introduce the notion of TIO LTS-signature that permits to structure basic notions to define a TIO LTS.

Definition 3.1 (TIO LTS Signature): A TIO LTS-signature Ω is a couple (I, O) where I ∩ O = ∅. I and O are sets whose elements are called inputs and outputs respectively. Let Ω1 = (I1, O2) and Ω2 = (I2, O2) be two TIO LTS-signatures. The sum of Ω1 and Ω2 is the TIO LTS-signature Ω1 ∪ Ω2 = ((I1 ∪ I2), (O1 ∪ O2), (O1 ∪ O2)).

Definition 3.2 (TIO LTS): A Timed Input/Output Labeled Transition System (TIO LTS) over (I, O) is a labeled transition system over I ∪ O ∪ D.

Often in the literature, inputs and outputs are represented by sending and receiving data on communication channels and denoted respectively by cl and cO where c is the channel and α is the data. Hence, by using the notations of Definition 3.2, cl and cO denote the couple (c, α) in I and O respectively.
the equipment life. We observe that the thermostat defines an operational profile for the equipment. We will see how to benefit from this observation in our testing framework.

Semantics of TIO LTSs can be characterized as sequences of actions being either inputs, outputs, or durations. Such sequences, called timed traces, are defined as usual traces of TIO LTSs by permitting to (de)compose all subsequent durations according to all possibilities offered by $D$. Taking into account these additional durations allows the test harness to report elapsed time with any granularity.

**Definition 3.3 (Timed traces of TIO LTSs):** Let $A$ be a TIO LTS over $(I, O)$. The set $TTraces(A)$ of timed traces of $A$ is the smallest set containing $Traces(A)$ and such that for any $a$, $b$, and $c$ in $D$ satisfying $a + b = c$, for any $\sigma, \sigma', \sigma''$ in $(I \cup O \cup D)^*$ and for any $r \in I \cup O \cup D$ we have:

- $\alpha' . c. \alpha'' \in TTraces(A) \Rightarrow \alpha' . a . b . \alpha'' \in TTraces(A)$;
- $\alpha' . a . b . \alpha'' \in TTraces(A) \Rightarrow \alpha' . c . \alpha'' \in TTraces(A)$;
- $TTraces(A)$ is closed under prefix.

A conformance relation is a binary relation over TIO LTSs which defines correctness of a System Under Test (SUT) with respect to its specification. SUT are TIO LTSs with additional properties.

**Definition 3.4 (System Under Test):** A System Under Test is a TIO LTS $(Q, q_0, Tr)$ over $(I, O)$ such that:

- Input enabling: $\forall q \in Q, \forall i \in I: \exists q' \in Q, (q, i, q') \in Tr$;
- Time elapsing: $\forall q, q' \in Q: \forall o \in O, (q, o, q') \notin Tr \Rightarrow \exists d \in D, (q, d, q') \in Tr$.

Input enabling is very classical in testing and only states that from the viewpoint of the tester, SUT can not refuse an input. Time elapsing ensures that the absence of reaction from SUT leads to observing a waiting time during which no output occurs.

**Definition 3.5 (tioco):** Let $A$ be a TIO LTS and $SUT$ be a system under test, both defined over $(I, O)$. $SUT$ conforms to $A$, denoted $SUT \ tioco A$, if and only if for every $\sigma$ in $TTraces(A) \cap TTraces(SUT)$ and every $r$ in $O \cup D$ we have:

$$\sigma . r \in TTraces(SUT) \Rightarrow \sigma . r \in TTraces(A).$$

**Example 3.2 (tioco):** Let us consider the trace $dt? -1 . 1 . 1 . equip\textsc{COOL}$ of some SUT implementing $A_{\text{equip}}$. This SUT is not correct (in the sense of $tioco$) because after the specified trace $dt? -1 . 1$, the non-specified duration 1 can be observed. That is the SUT allows in all a duration of 2 time units before outputting $equip\textsc{COOL}$. However the intended behavior is to have a duration of exactly 1 time unit.

**IV. Compositionality Result**

In the sequel, we need to consider compositions of TIO LTSs. We use the LTS composition operators $\parallel$ (see Section II). The composition of two TIO LTSs $A_1$ and $A_2$ respectively defined over $\Omega_1 = (I_1, O_1)$ and $\Omega_2 = (I_2, O_2)$ such that $I_1 \cap I_2 = O_1 \cap O_2 = \emptyset$, is the TIO LTS $A_1 \parallel A_2$ over $\Omega_1 \cup \Omega_2$ (see Definition 3.1). The constraint of partitioning on inputs and output sets is important because we assume that TIO LTSs interact by synchronization of the output actions occurring in one TIO LTS with the input actions in the other TIO LTS. Note that when a duration occurs in the resultant TIO LTS, it means that both composed TIO LTSs allow time to advance of that duration (since $D$ is a common set of labels).

**Definition 4.1 (Projection):** Let $A_1$ and $A_2$ be two TIO LTSs respectively over $\Omega_1 = (I_1, O_1)$ and $\Omega_2 = (I_2, O_2)$. Moreover let us suppose that $I_1 \cap I_2 = \emptyset$ and $O_1 \cap O_2 = \emptyset$. For any $i, j \in \{1, 2\}$ s.t. $i \neq j$:

- the projection of $A_1 \parallel A_2$ on $\Omega_i$, denoted $(A_1 \parallel A_2)_{\Omega_i}$ is the TIO LTS $\text{Hide}((I_j \cup O_j) \setminus (I_i \cup O_i), A_1 \parallel A_2)$ over $\Omega_i$;

- let $\sigma$ be a timed trace of $A_1 \parallel A_2$, the projection of $\sigma$ on $\Omega_i$, denoted $\sigma_{\Omega_i}$, is the trace in $(I_i \cup O_i \cup D)^*$ which is equal to $\text{Hide}((I_j \cup O_j) \setminus (I_i \cup O_i), \sigma)$.

(See Section II for the definition of the operator $\text{Hide}$.)

**Example 4.1 (TIO LTS composition and projection):** We show in Figure 2 $A_{\text{therm}} \parallel A_{\text{equip}}$ and $(A_{\text{therm}} \parallel A_{\text{equip}})_{\Omega_{\text{equip}}}$. Transformations of Definition 4.1 are indicated using dashed curvy arrows.

**Proposition 4.1:** For any $i \in \{1, 2\}$, the following facts are satisfied:

(i) $TTraces((A_1 \parallel A_2)_{\Omega_i}) \subseteq TTraces(A_i)$;

(ii) $\forall \sigma \in TTraces(A_1 \parallel A_2), \sigma_{\Omega_i} \in TTraces((A_1 \parallel A_2)_{\Omega_i})$;

(iii) $\forall \sigma \in TTraces(A_1 \parallel A_2), \sigma_{\Omega_i} \in TTraces(A_i)$.

For proof see Appendix.

Here, we define a property about system specifications, i.e., a composition of TIO LTSs, that implies our compositionality result for $tioco$ along composition.

**Definition 4.2 (Local consistency):** With notations of Definition 4.1, $A_1 \parallel A_2$ is said to be locally consistent if and only if for any two $\sigma, \sigma' \in TTraces(A_1 \parallel A_2)$ such that there exists $i \in \{1, 2\}$ with $\sigma_{\Omega_i} = \sigma'_{\Omega_i}$, we have:

$$\forall r \in (\Omega_i \cup O_i). (\sigma . r \in TTraces(A_1 \parallel A_2) \Leftrightarrow \sigma' . r \in TTraces(A_1 \parallel A_2)).$$

The local consistency property states that if there exist two traces at the system level which are not distinguishable from the viewpoint of a subsystem, then the system will continue these traces with the same immediate outputs, or allow for the same amounts of time to elapse (in case no immediate output can be produced).
Example 4.2 (Counterexample to local consistency):
Consider the example in Figure 3. Note that we have not introduced durations (except implicit zero durations). Consider the two traces \( \sigma = e_5.a_5 \) and \( \sigma' = a_5 \Gamma \) of \( A_1 \| A_2 \) and the output \( r = b_5 \), we have \( \sigma_1 = \sigma_1' = a_5 \) and \( \sigma_2 = e_5.a_5.b_5 \) is a trace of \( A_1 \| A_2 \). However, \( \sigma'.r = a_5.b_5 \) is not a trace of \( A_1 \| A_2 \). In fact, \( e_5.a_5 \) must occur in order for \( A_1 \| A_2 \) to accept successively \( a_5 \) then \( b_5 \) which is not the case for \( \sigma' \).

Consider the SUTs in Figure 4. Reconsider the composition

![Diagram](image)

Figure 3: \( A_1 \| A_2 \) is not locally consistent

\( A_1 \| A_2 \) in Figure 3c which is not locally consistent for outputs. We suggest two SUTs namely \( SUT_1 \) and \( SUT_2 \) respectively in Figures 4a–4b. Yet as shown in Figure 4c, the composition \( SUT_1 \| SUT_2 \) is not conform to \( A_1 \| A_2 \): after the trace \( a_5 \), \( SUT_1 \| SUT_2 \) reacts with an unspecified output \( b_5 \) since \( a_5.b_5 \notin TT(\sigma_1 \| \sigma_2) \).

![Diagram](image)

Figure 4: Counterexample to compositional testing: \( SUT_1 \ tioco (A_1 \| A_2), \) and \( SUT_2 \ tioco (A_1 \| A_2), \) but \( SUT_1 \| SUT_2 \ tioco (A_1 \| A_2) \) (non conform output \( b_5 \)).

Theorem 4.1 (Compositionality): Let \( A_1 \) and \( SUT_1 \) (respectively \( A_2 \) and \( SUT_2 \)) be two TLIOSTS over \( \Omega_1 \) (respectively \( \Omega_2 \)). Moreover let us suppose that \( I_1 \cap I_2 = \emptyset \) and \( O_1 \cap O_2 = \emptyset \). Let us suppose that \( A_1 \| A_2 \) is locally consistent. Then we have:

\[
SUT_1 \ tioco (A_1 \| A_2), \quad \text{and} \quad SUT_2 \ tioco (A_1 \| A_2), \quad \Rightarrow \quad SUT_1 \| SUT_2 \ tioco (A_1 \| A_2).
\]

For proof see Appendix.

Theorem 4.1 states that we can deduce the conformity of \( SUT_1 \| SUT_2 \) to \( A_1 \| A_2 \) by reasoning on \( SUT_1 \) and \( SUT_2 \). Let us consider the contrapositive of Theorem 4.1 (with notations of Theorem 4.1):

\[
SUT_1 \| SUT_2 \ tioco (A_1 \| A_2) \Rightarrow SUT_1 \ tioco (A_1 \| A_2) \text{ or } SUT_2 \ tioco (A_1 \| A_2).
\]

This means that any non conformance (in the sense of \( tioco \)) of the system with respect to a given specification boils down to a non conformance of one of its subsystems with respect to a particular TIOLTS which is the projection of the specification on the interface of the concerned subsystem. In the next section, we show how to compute this projection in a symbolic framework and how Theorem 4.1 can be used to reason about conformance with respect to the specification of the system (as a composition).

V. SYMBOLIC EXTENSION

When modeling real systems, it is usual to handle durations in the form of constraints which specify upper bounds or lower bounds concerning delays between actions. Obviously, the framework of TLIOSTS is not powerful enough to deal with such issues. For this reason, we propose to use Timed Input Output Symbolic Transition Systems (TIOMTS) [3, 12]. TIOMTSs re-introduced in this paper are very much alike these extensions [2], [23] of Timed Automata [1] where both time and data are represented symbolically. The symbolic query language of the Diversity tool implemented at CEA LIST [21] has been enriched to support TIOMTSs.

The Diversity tool is a symbolic execution engine provided with various testing algorithms [11, 15]. The most recent ones deal with timed constraints [3, 12]. In this section, we discuss how to enrich our symbolic testing framework for timed systems, in order to take benefits of Theorem 4.1.

A. TIOMTS

In the sequel, we suppose that a data signature \( \Sigma = (S,F,P) \) and a \( \Sigma \)-model \( M \) are given. In order to denote durations, we further suppose that \( \Sigma \) contains a particular sort \( time \) such that \( M_{time} = D \). \( F \) and \( P \) contain some classical operations \( + : time \times time \to time, \ - : time \times time \to time, \) or predicates \( <, \leq, \times time \), associated with their usual semantical meanings. Variables of sort \( time \) are called clocks. For a set of clocks \( T \), we note \( S_{time}(T) \subset S_{\Sigma}(T) \) the set of formulas only containing conjunctions of atoms of the form \( z \leq d, \ z < d, \ z < d < z, \ where \ d \ is \ a \ constant \ and \ z \ is \ in \ T \).

TIOMTSs are defined over a TIOMTS signature \( \Gamma = (C, A, T) \) where \( C \) is a set of channels, \( A \) is a set of variables (whose sort is not \( time \)) called attribute variables, and \( T \) is a set of clocks. Moreover \( C \) can be decomposed as \( C^\text{in} \cup C^\text{out} \) where channels of \( C^\text{in} \) (respectively \( C^\text{out} \)) are called input (respectively output) channels. The set of symbolic actions over \( \Gamma \), noted \( Act(\Gamma) \), is \( I(\Gamma) \cup O(\Gamma) \cup \{ t \} \) where \( I(\Gamma) = \{ (c,x) | x \in A, c \in C^\text{in} \} \) and \( O(\Gamma) = \{ (c,t) | t \in T(\Delta), c \in C^\text{out} \} \) are respectively the set of inputs and the set of outputs over \( \Gamma \). As for TIOLTS, we use the notation
c?x and clt to denote the couples \((c, x)\) and \((c, t)\) in \(I(\Gamma)\) and \(O(\Gamma)\) respectively.

**Definition 5.1 (TIOSTS):** A TIOSTS over \(\Gamma = (C, A, T)\) is a triple \((Q, q_0, Tr)\), where \(Q\) is a set of states, \(q_0 \in Q\) is the initial state and \(Tr\) is a set of transitions of the form 
\[q, \phi, \psi, \mathbb{T}, act, \rho, q'\] 
where \(q, q' \in Q\), \(\phi \in \text{Sen}_{time}(T)\), \(\psi \in \text{Sen}_{\mathbb{T}}(A)\), \(T \subseteq T\), \(act \in \text{Act}(\Gamma)\), and \(\rho : A \rightarrow T_{2}(A)\) is a substitution.

When firing a transition \((q, \phi, \psi, \mathbb{T}, act, \rho, q')\), \(\phi\) is a formula constraining the delay at which the action occurs, \(\psi\) is a firing condition on attribute variables, \(\rho\) assigns new values to attribute variables, and clocks in \(\mathbb{T}\) are reset.

**Example 5.1:** Consider the TIOSTS of the thermostat system in Figure 5 (which have inspired the TIOLTs design). As expressed in \(G_{therm}\) the objective is to maintain the temperature within a certain range \((T_{min} \leq t \leq T_{max})\). The thermostat asks the equipment to apply the difference between the sensed temperature and the exceeded threshold (see actions \(\text{dt}!t_{min} \rightarrow t, \text{dt}!t_{max} \rightarrow t\)). The equipment then heats/cools at \(\pm 4\) degrees/time unit (the clock \(c\) of \(G_{equip}\) is used to constrain the occurrences of actions \(\text{equip} \pm 4\)).

**Asynchronous execution:** for any \((q_1, \phi_1, \psi_1, \mathbb{T}_1, act_1, \rho_1, q'_1)\) \(\in Tr_1\) and for any \(q_2 \in Q_2\) where \(act_2\) is not of the form \(c?x\) or \(clt\) with \(c \notin C_2\), we have \(((q_1, q_2), \phi_1, \psi_1, \mathbb{T}_1, act_1, \rho_1, (q'_1, q'_2)) \in Tr\). The role of \(G_1\) and \(G_2\) can be inverted.

Any TIOSTS can be canonically associated with a TIOLTS. The set of timed traces of this TIOLTS represents the semantics of the TIOSTS. Due to the lack of space, we do not give in detail the process here; interested readers can refer to [3]. Instead, we focus on a symbolic computational way of approaching those timed traces. The computation process is based on symbolic execution [16].

**Symbolic states are used for that purpose.**

**Definition 5.3 (Symbolic state):** Let \(F = (\mathcal{F})_{s \in S}\) be a set of variables. A symbolic state for \(G\) over \(F\) (symbolic state for short) is a tuple \((q, \theta, \pi, \vartheta, \lambda)\) where \(q \in Q\), \(\theta \in \text{Sen}_{time}(\mathcal{F}_{time})\), \(\pi \in \text{Sen}_{\mathbb{T}}(\mathcal{F})\), \(\vartheta \in T_{2}(\mathcal{F}_{time})\) and \(\lambda : A \cup T \rightarrow T_{2}(\mathcal{F})\) is an application preserving types.

In Definition 5.3, \(F\) is the set of symbolic parameters used in the symbolic execution process. In the sequel, we suppose given a set \(F\) disjoint of any set of variables introduced in any TIOSTS signature. We note \(S\) the set of all symbolic states over \(F\). For a symbolic state \(\eta = (q, \theta, \pi, \vartheta, \lambda)\), \(q\) denotes the state reached after an execution leading to \(\eta\), \(\theta\) is a constraint on symbolic duration values called time path condition, \(\pi\) is a constraint on symbolic data values called data path condition, \(\vartheta\) denotes the duration from the beginning of the execution leading to \(\eta\), and \(\lambda\) denotes terms over symbolic variables in \(F\) that are assigned to variables of \(A\cup T\). In the sequel, \(\Gamma_F\) stands for the TIOSTS signature \((C, F \setminus \text{time}, \text{time})\). Moreover, for any symbolic state \(\eta = (q, \theta, \pi, \vartheta, \lambda)\), \(q_0, \eta_0, \pi_0, \vartheta_0\) and \(\lambda_0\) stand respectively for \(q, \theta, \pi, \vartheta\) and \(\lambda\), any application \(\lambda : A \cup T \rightarrow T_{2}(\mathcal{F})\) can be extended in a canonical way to \(T_{Z}(A \cup T), \text{Sens}_{\mathbb{T}}(A \cup T)\) and \(\text{Act}(\Gamma)\). All these extensions are also simply denoted by \(\lambda\).

**Definition 5.4 (Symbolic execution of transitions):** Let \(G = (Q, q_0, Tr)\) be a TIOSTS over \(\Gamma\). Let \(tr = (q, \phi, \psi, \mathbb{T}, act, \rho, q')\) be a transition of \(Tr\). Let \(\eta\) be a symbolic state of \(S\) satisfying \(q_0 = q\). We note \(\lambda : A \cup T \rightarrow T_{2}(\mathcal{F})\) the substitution defined as follow:

- if \(act \in O(\Gamma)\) \(\cup \{\tau\}\), then that for all \(x \in A\), \(\lambda(x) = \lambda(x)\);
- if \(act \in I(\Gamma)\) of the form \(c?y\) then \(\lambda(y)\) is a fresh variable of \(F\) and for all \(x \in A \setminus \{y\}\), \(\lambda(x) = \lambda(x)\);
- for all \(w \in T\) we have \(\lambda(w) = \lambda(w) + z\) where \(z \in \mathcal{F}_{time}\) is a new fresh variable.

The symbolic execution of \(tr\) from \(\eta\) is \(st = (\eta, act_F, \eta')\) \(\in S \times \text{Act}(\Gamma_F) \times S\) such that \(q_0' = q'\), \(\eta' = \eta + z\), for all
We give only relevant information contained in the symbolic state $\eta_{133}$ for illustration: $\theta_{133}$ cumulates the sum of durations $z_0 + 2z + z_7 + z_{16} + z_{34} + z_{69}$; the time path condition $\theta_{133}$ is $z_7 < 1$; the data path condition $\pi_{133}$ is $t_2 < T_{min}$ and $T_{min} - t_2 - 4 \leq 0$; $g_{22}$ is assigned with $0$ (equals $z_7$ from $\eta_8$ to $\eta_7$). Actually, the system was heating, then it is being switch off at this level of the execution. Dashed curve arrows show the projection that we will define later in the Subsection V-B. □

With notations of Definition 5.5, we define paths of $SE(G)$ as finite sequences $s_t \ldots s_{tn}$ of transitions of $ST$, such that $source(s_t) = Init$ and for every $i < n$, we have $target(s_t) = source(s_{t+i})$. For any finite path $p = s_t \ldots s_{tn}$, $target(p)$ is $Init$ if $n = 0$ and $target(s_{tn})$ otherwise. We note $Path(SE(G))$ the set of all such paths. For a path $p$, we note $Seq(p)$ the sequence defined as $\varepsilon$ if $p = \varepsilon$, $Seq(p).delay(st)$ if $p$ is of the form $p'.st$ with $act(st) = \tau$ and $Seq(p).delay(st).act(st)$ if $p$ is of the form $p'.st$ with $act(st) \neq \tau$. For any interpretation $\nu \in MF_2$, we extend it canonically to $I(\Gamma_T) \cup O(\Gamma_T)$: for any $c \in C, z \in T$ and $t \in T_4(\mathcal{F})$ we have $\nu(cz) = c\nu(z), \nu(t) = c\nu(t)$. We extend $\nu$ to $Seq(p)$ as follow: if $Seq(p)$ is $\varepsilon$ then $\nu(Seq(p)) = \varepsilon$, and if $Seq(p)$ is of the form $seq.x$ with $x \in I(\Gamma_T) \cup O(\Gamma_T) \cup F_{time}$ then $\nu(Seq(p)) = \nu(seq'.\nu(x))$.

Definition 5.6 (Timed Traces of a TIOSTS): For any $p \in Path(SE(G))$, we define $TTTraces(p)$ as the least set such that:

- for any $\nu$ such that $M \models \nu \theta(target(p)) \wedge \pi(target(p))$ then we have $\nu(Seq(p)) \in TTTraces(p)$;
- for any timed trace of the form $\sigma.\nu.\sigma'$ in $TTTraces(p)$ with $d \in M_{time}$, for any $a, b \in M_{time}$ such that $d = a + b$ we have $\sigma.a.\sigma' \in TTTraces(p)$;
- $TTTraces(p)$ is closed by prefix.

$TTTraces(G) = \{ \bigcup_{p \in Path(SE(G))} TTTraces(p) \}$

The symbolic execution of a TIOSTS characterizes in intention the set of timed traces of the TIOSTS. Therefore it can be used as reference, rather than the TIOSTS itself, in the frame of testing algorithms aiming at assessing the conformance of an SUT to a TIOSTS specification. The $tioco$ relation is adapted as follows:

Let $G$ be a TIOSTS and $SUT$ be a system under test: $SUT$ conforms to $G$ if and only if for any $\sigma \in TTTraces(G) \cap TTTraces(SUT)$, for every output or duration $r$, we have: $\sigma.r \in TTTraces(SUT) \Rightarrow \sigma.r \in TTTraces(G)$.

We defined several testing algorithms for black box testing based on symbolic execution of TIOSTS [3, 11, 12, 15].

All those algorithms take as input: (1) a reference symbolic execution, (2) a so-called test purpose corresponding to distinguished paths in the symbolic execution; those paths represent behaviors that algorithms try to cover while interacting with $SUT$; (3) the system under test itself. The algorithms build (timed) traces by interacting with SUTs and compute different verdicts depending on the fact that the trace is (or is not), a (timed) trace of the test purpose or of the symbolic execution. In section V-B, we discuss how to adapt symbolic execution in order to be able to use those algorithms in a compositional testing approach. We also discuss how to assess the local consistency property which is a necessary condition to benefit from Theorem 4.1.
that those two TIOSTS do not share inputs and outputs. It is sufficient to impose $C_1^\text{in} \cup C_2^\text{in} = \emptyset$ and $C_1^\text{out} \cup C_2^\text{out} = \emptyset$.

Let us now suppose that we want to test a SUT of the form $SUT_1 |\parallel SUT_2$ ($SUT_1$ and $SUT_2$ are two TIOSTS with respect to the specification $G_1|G_2$). Let us further suppose that we want to test $SUT_1 |\parallel SUT_2$ in a compositional manner by testing successively $SUT_1$ and $SUT_2$. The first step consists in defining two reference symbolic executions respectively for $SUT_1$ and $SUT_2$. This is done by adapting Definition 4.1 to the case of TIOSTS. As we consider the composition $G_1|G_2$, $\Gamma_x = (C \cup C_2, F \setminus F_{\text{time}1}, F_{\text{time}2})$, $\mathcal{E}(G_1|G_2)$ can be seen as an LTS over $(I(\Gamma_x) \cup O(\Gamma_x) \cup \{\tau\})$. Let us define for any $i,j \in \{1,2\}$, $\Gamma_x^i = (C \setminus F_{\text{time}j}, F_{\text{time}j})$. Thus, for any $i,j \in \{1,2\}$ s.t. $i \neq j$, the projection of $\mathcal{E}(G_1|G_2)$ on $\Gamma_x^i$ denoted $(G_1|G_2)_{\Gamma_x^i}$, is the LTS $\mathcal{H}ide((I(\Gamma_x^i) \cup O(\Gamma_x^i)) \setminus (I(\Gamma_x^j) \cup O(\Gamma_x^j)), \mathcal{E}(G_1|G_2))$ over $(I(\Gamma_x^i) \cup O(\Gamma_x^i) \cup \{\tau\})$ (see Example 5.2 for illustration). In the same way, we define the projection $\sigma_1$, of any $\sigma \in \mathcal{TTraces}(\mathcal{E}(G_1|G_2))$. Note that trivially, $\Gamma_x^1$, is in $\mathcal{TTraces}(\mathcal{E}(G_1|G_2)_{\Gamma_x^1})$ (symbolic transposition of Fact(ii) of Proposition 4.1). Moreover, due to the lack of space, we restrict ourselves to systems resulting of the composition of two TIOSTS. However, we can directly extend the projection mechanism the case where $G_1$ and $G_2$ are themselves resulting of the composition of more basic TIOSTS and define recursively the projections on those basic TIOSTS.

In the symbolic context, taking benefits of Theorem 4.1 consists in reducing the testing process of $SUT_1 |\parallel SUT_2$ with respect to $G_1|G_2$ to testing $SUT_1$ and $SUT_2$ with respect to $(G_1|G_2)_{\Gamma_x^1}$ and $(G_1|G_2)_{\Gamma_x^2}$ respectively, while ensuring that any fault that can be discovered by testing $SUT_1 |\parallel SUT_2$ can be discovered either by testing $SUT_1$ or $SUT_2$. In order to get this result, we require the composition of the specifications to be locally consistent, which can be formulated as follow in the symbolic context:

For any two $\sigma, \sigma' \in \mathcal{TTraces}(G_1|G_2)$ such that there exists $i \in \{1,2\}$ with $\sigma_\Gamma = \sigma_{\Gamma^i}$, we have for every $r$ an output of $\Omega_1$ or a duration:

$$\sigma.r \in \mathcal{TTraces}(G_1|G_2) \iff \sigma'.r \in \mathcal{TTraces}(G_1|G_2).$$

In the general case, one can not lead that proof because it would require to fully compute $\mathcal{E}(G_1|G_2)$ which may contain infinite paths. Therefore, we propose to test this property. This testing process comes after the two testing phase respectively for $SUT_1$ and $SUT_2$. More precisely, whatever testing algorithm is used, testing $SUT_i$ with respect to $(G_1|G_2)_{\Gamma_x^i}$, implies to interact with $SUT_i$ which results on a set of timed traces, denoted $\mathcal{TTraces}_i$. Any element of $\mathcal{TTraces}_i$ represents one interaction sequence. Our approach consists in examining whether one of those traces permits to show that the local consistency property is not satisfied. For that purpose, we consider all traces in $\mathcal{TTraces}_i \cap \mathcal{TTraces}(G_1|G_2)$. In the sequel, for any path $p \in \text{Path}((G_1|G_2)_{\Gamma_x^i})$, we note $\text{Ext}(p)$ the extensions of $p$ for the set $\{p\} \cup \{p.st \mid p.st \in \text{Path}((G_1|G_2)_{\Gamma_x^j})\}$. If one trace $\mathcal{TTraces}_i \cap \mathcal{TTraces}(G_1|G_2)$ (say $\sigma$) permits to show that the local consistency property is not satisfied, it means that one of its prefix (say $\sigma_p$) can be used as a counterexample to the local consistency property. In other words, if we note $\sigma_\Gamma$ this prefix, we have:

$$(A) \text{ There exists } \sigma, \sigma' \in \mathcal{TTraces}(G_1|G_2) \text{ and } r \text{ an output of } \Omega_1 \text{ or a duration such that:}$$

$$\sigma_\Gamma = \sigma_p \land \sigma_{\Gamma^i} = \sigma_p \land \sigma.r \in \mathcal{TTraces}(G_1|G_2) \land \sigma'.r \notin \mathcal{TTraces}(G_1|G_2).$$

Since $\sigma$ and $\sigma'$ projections on $(G_1|G_2)_{\Gamma_x^i}$ are the same and equal to $\sigma_p$, (from (A) and by the definition of the projection) we have:

$$(B) \text{ There exists two paths } p, p' \in \text{Path}((G_1|G_2)_{\Gamma_x^i}) \text{ such that:}$$

$$- \text{ there exists } \bar{p} \in \text{Ext}(p) \text{ such that } \sigma.p.r \in \mathcal{TTraces}(\bar{p});$$

$$- \text{ for any } \bar{p}' \in \text{Ext}(p') \text{ we have } \sigma.p.r \notin \mathcal{TTraces}(\bar{p}).$$

**Algorithm 1:** Checking local consistency

| Data: This function is applied for all $\sigma \in \mathcal{TTraces}_i \cap \mathcal{TTraces}(G_1|G_2)$. |
|---|
| 1 begin |
| 2 for all prefix of $\sigma$, of the form $\sigma.p.r$ where $r$ is an output of $\Omega_1$ or a duration do |
| 3 for $p \in \text{Path}((G_1|G_2)_{\Gamma_x^i})$ such that $\sigma.p \in \mathcal{TTraces}(p)$ do |
| 4 for $\bar{p} \in \text{Ext}(p)$ do |
| 5 if $(\sigma.p.r \in \mathcal{TTraces}(\bar{p}))$ then |
| 6 break |
| 7 return NOT_VIOLATED |

The objective of Algorithm1 is to check the local consistency property by systematically exploring all prefixes (ending on an output or a duration) of all traces of $\mathcal{TTraces}_i \cap \mathcal{TTraces}(G_1|G_2)$. This is in order to assess whether or not the prefix satisfy the property (B) and thus questions the local consistency property.

**VI. RELATED WORKS**

Model-based testing of components oriented systems has been widely addressed in the literature. Testing components in the context of the system in which they are plugged, has received much more attention [11], [14], [19], [20], [26]. In this paper, we rather explore how to avoid testing the whole system, thanks to results ensuring that any system failure can be detected during unit testing (thus components are supposed to be tested in isolation rather than in context). In this sense, our contribution relates more closely to [7], [9], where authors study compositionality in the settings of conformance relation ioco [24], which is a non-trimmed version of tioco. In particular, the work [7] provides a similar result to Theorem 4.1, by identifying hypothesis ensuring that the conformance (in the sense of ioco) of $SUT_1$ and $SUT_2$ to $A_1$ and $A_2$ respectively, implies the conformance of $SUT_1|SUT_2$ to $A_1|A_2$. The result in [7] can not be trivially extended to tioco. Indeed, it holds thanks to the hypothesis of input completeness of specifications which guarantees that their composition never introduce synchronization deadlocks. Extending this hypothesis so that durations can not introduce such deadlocks needs further research. Moreover, in our approach instead of having as premises that $SUT_1$ and $SUT_2$ conform to...
and $A_2$ respectively, we have $SUT_1$ and $SUT_2$ conform to the projections $(\langle A_1||A_2 \rangle)_{\Omega_1}$ and $(\langle A_1||A_2 \rangle)_{\Omega_2}$ respectively. The disadvantage of our result (versus [7]) is that its exploitation requires to compute the composition of specifications. Thus, our result is not useful to tackle the problem of state explosion due to the composition. In practice, we use our result with high level specifications such as those that we discussed in [4] (given in UML sequence diagram then translated into TIOSTSs). As argued in the introduction, our result is useful to enable testing components earlier, in order to overcome the technological difficulties of instrumenting the global system and to ease its debugging. The main advantage of our approach, with respect to the kind of results in [7], is that $(\langle A_1||A_2 \rangle)_{\Omega_1}$ represents a much more accurate specification than $A_1$, which captures how $SUT_i$ is going to be used in $SUT_1||SUT_2$. Since model-based testing requires extracting a finite number of test cases from the reference specification, using $(\langle A_1||A_2 \rangle)_{\Omega_1}$ as a reference for testing gives the tester more chances to identify faults of $SUT_i$ which cause faults at the system level. Finally, another close work [18] studies the problem of deriving a specification for a third-party component, based on the specification of the system and the environment in which the component is supposed to reside. This study is however again conducted in the framework of ioco rather than tioco. Up to our knowledge, our contribution is the first to provide both a compositionality result in the settings of tioco and discuss how to exploit it in timed symbolic testing.

VII. FUTURE WORKS

We experimented in [5] the derivation of test purposes using projection mechanisms. In [4], we discussed the application of those techniques on an industrial railway use case\(^1\) specified as UML sequence diagrams extended with time constraints, which were translated into TIOSTSs (the translation is given in [5]). Besides, we defined symbolic testing algorithms in [3], [12]. By mixing the two approaches (projection and testing algorithms), we are able to extract from a system specification, test cases for any of its subsystems. The result presented in this contribution allows us to go a step further: we now have a theorem ensuring that any fault that could be discovered at the system level, can be discovered by partitioning the system into a group of subsystems, and by successively testing each subsystem of the group with respect to its derived system specification (by projection). The first hypothesis to this result is the local consistency of the system specification and we have presented a testing algorithm to address it. The second hypothesis, on the system under test, is that all subsystems are connected in a way which implements the mathematical compositionality result in the settings of tioco and discuss how to exploit it in timed symbolic testing.

---

References

16. J.-C. King. A new approach to program testing. Int. Conf. on Reliable software, 1975.
Let us note that any path $TTraces((A_1 || A_2)_{\Omega_1}) \subseteq TTraces(A_1)$ (the case of $A_2$ is symmetric).

Let us remark that for any TIOLTS $A$ and $B$ such that $Traces(A) \subseteq Traces(B)$, it is quite obvious, based on Definition 3.3, that $TTraces(A) \subseteq TTraces(B)$: any timed trace of $TTraces(A)$ can be obtained by applying, in an appropriate order the three items of Definition 3.3, on a simple trace of $Traces(A)$ (which by hypothesis is also a trace of $Traces(B)$). Therefore, $\sigma$ is also in $TTraces(B)$. It is thus sufficient to prove the following property in order to prove fact(i):

\[(P) \text{ for any trace } \sigma \in Traces((A_1 || A_2)_{\Omega_1}), \text{ we have } \sigma \in Traces(A_1).\]

Let us note that any path $p$ of $(A_1 || A_2)_{\Omega_1}$ can be also seen a TIOLTS whose source state is the initial state of $(A_1 || A_2)_{\Omega_1}$ and whose transition set is composed of all the occurrences of transition of $(A_1 || A_2)_{\Omega_1}$ in $p$. In the sequel, we also note $p$ to denote this TIOLTS, and we may apply the projection operation on it. Obviously, for all $p \in Path((A_1 || A_2)_{\Omega_1})$, we have $p_{\Omega_1} \in Path((A_1 || A_2)_{\Omega_1})$. Moreover, if $\sigma \in Traces((A_1 || A_2)_{\Omega_1})$ then there exists a path $p \in Path(A_1 || A_2)$ such that $\sigma = \text{trace}(p_{\Omega_1})$. It is thus sufficient to prove the following property in order to prove $(P)$:

\[\exists p \in Path(A_1 || A_2) \cdot (\sigma = \text{trace}(p_{\Omega_1})) \Rightarrow (\exists p' \in Path(A_2) \cdot (\sigma = \text{trace}(p')))\]

The proof is conducted by induction on the form of $\sigma$:

- The basic case is the one where $\sigma$ is $\varepsilon$, and the property trivially holds: it suffices to consider $p = p' = \varepsilon$, moreover if we note target$(p)$ and target$(p')$ the state reached by those (empty) paths in their respective TIOLTS, we have target$(p) = (q_0^1, q_0^2)$ and target$(p') = q_0^1$.

- Otherwise $\sigma$ is of the form $\sigma'.act$ where $\sigma'$ is a trace and act is an action or a duration. $p$ can be decomposed as $p_\sigma.tr_1 \cdots tr_n$ with the following induction hypothesis:
  - $\sigma' = \text{trace}(p_\sigma), \text{act}(tr_n) = \text{act}$ and for all $i < n$ we have act$(tr_i) = \tau$;
  - there exists $p'_{\sigma'} \in Path(A_1)$ such that $\sigma' = \text{trace}(p_\sigma), \sigma' = \text{trace}(p'_{\sigma'})$ and if we note target$(p_\sigma)$ as $(q_1, q_2)$ then we have target$(p'_{\sigma'}) = q_1$.

From the sequence $tr_1 \cdots tr_n$ we construct the following sequence of transitions $tr'_1 \cdots tr'_n$ of $A_1$:

- for all $i \leq n$ if act$(tr_i)$ is an input or an output that is not introduced in $\Omega_1$ then $tr_i$ is removed from the sequence.

- for all $i \leq n$ such that the previous conditions do not hold, if we note $tr_i$ as $((q_1, q_2), \text{act}, (q'_1, q'_2))$ then $tr_i$ is replaced by $(q_1, \text{act}, q'_1)$.

From the definition of LTS composition it is clear that $p_{\sigma'.tr'_1 \cdots tr'_n}$ is a path of $Path(A_1)$ and $\text{trace}(p_{\sigma'.tr'_1 \cdots tr'_n})$ is $\sigma$. We end the proof by defining $p'$ as $p_{\sigma'.tr'_1 \cdots tr'_n}$.

**We prove fact(ii)** by induction on the form a trace $\sigma \in TTraces(A_1 || A_2)$, we consider the case of $\Omega_1$ (the case of $\Omega_2$ is symmetric). For the same reason than in the proof of fact(i) it is sufficient to consider that $\sigma \in Traces(A_1 || A_2)$.

- The basic case is when $\sigma$ is $\varepsilon$:
  Trivially $\varepsilon$ is in $Traces(A_1 || A_2)$ and in $Traces((A_1 || A_2)_{\Omega_1})$.
- Inductive case:

  We suppose that $\sigma$ is of the form $\sigma'.act$ and $\sigma'_\Omega \in Traces((A_1 || A_2)_{\Omega_1})$. If act is an input or an output that is not introduced in $\Omega_1$, or if act is $\tau$ then we have $\sigma_\Omega = \sigma'_\Omega$ and thus $\sigma_\Omega \in Traces((A_1 || A_2)_{\Omega_1})$. Now, we suppose that act is either an input or an output of $\Omega_1$, or a duration.

  Since $\sigma$ is a trace of $A_1 || A_2$, we deduce that there exists $p \in Path(A_1 || A_2)$ such that $\sigma' = \text{trace}(p)$ and $\sigma'_\Omega = \text{trace}(p_\Omega)$. Moreover, since $\sigma'.act$ is a trace of $A_1 || A_2$ we may well choose $p$ so that it can be extended as $p.tr_1 \cdots tr_n$ with act$(tr_n) = \text{act}$ and for all $i < n$ we have act$(tr_i) = \tau$. Obviously trace$(p.tr_1 \cdots tr_n) = \sigma'.act$. Moreover, since act$(tr_n)$ is either an input or an output of $\Omega_1$, or a duration, it is clear from Definition 4.1 that $(p.tr_1 \cdots tr_n)_{\Omega_1}$ is $p_{\Omega_1}.tr_1 \cdots tr_n$ that is a path of $(A_1 || A_2)_{\Omega_1}$, and whose trace is $\sigma'_\Omega.act$ which is equal to $\sigma_\Omega$ from Definition 4.1.

**Fact(iii)** is a direct consequence of fact(i) and fact(ii).

**Proof:** Theorem 4.1

We suppose that the assertion $SUT_1 \parallel SUT_2$ tioco $A_1 || A_2$ does not hold and show it implies:

- either that $SUT_1$ tioco $(A_1 || A_2)_{\Omega_1}$ does not hold, or that $SUT_2$ tioco $(A_1 || A_2)_{\Omega_2}$ does not hold.

If $SUT_1 \parallel SUT_2$ tioco $A_1 || A_2$ does not hold, it means:

(HYP): There exists a timed trace $\sigma$ and $\tau$ an output of $\Omega_1 \cup \Omega_2$ or a duration such that:

\[\sigma \in TTraces(A_1 || A_2)\]
\[\sigma.\tau \notin TTraces(A_1 || A_2)\]
\[\sigma.\tau \in TTraces(SUT_1 \parallel SUT_2)\]

From (HYP) we deduce that for any $i$ in $\{1, 2\}$ we have:
(A) \((\sigma)_{\Omega_i} \in TTraces((A_1||A_2)_{\Omega_i})\) (from fact(ii)),

(B) \((\sigma.r)_{\Omega_i} \in TTraces(SUT_i)\) (from fact(iii))

Now we prove that there exists \(i \in \{1, 2, \}\) such that,

\[(\sigma.r)_{\Omega_i} \notin TTraces((A_1||A_2)_{\Omega_i}).\]

If we have \((\sigma.r)_{\Omega_i} \in TTraces((A_1||A_2)_{\Omega_i})\) then there exists \(\sigma' \in TTraces(A_1||A_2)\) such that \(\sigma'_{\Omega_i} = (\sigma.r)_{\Omega_i}\).

Now since \(r\) is a duration or an output there is at least one \(i \in \{1, 2\}\) such that \((\sigma.r)_{\Omega_i} = \sigma_{\Omega_i}.r\) (direct consequence of Definition 4.1). In the sequel we choose such an \(i\).

\(\sigma'\) is necessarily of the form \(\sigma''.r\) with \(\sigma''_{\Omega_i} = \sigma_{\Omega_i}\) because this is the only way to ensure that \(\sigma'_{\Omega_i} = (\sigma.r)_{\Omega_i}\).

If such \(\sigma''\) exists then we have two traces \(\sigma''\) and \(\sigma\) such that \(\sigma''_{\Omega_i} = \sigma_{\Omega_i}\) and there exist \(r\) an output of \(\Omega_i\) or a duration such that \(\sigma.r \notin TTraces(A_1||A_2)\) (second item of \((HYP)\)) and \(\sigma'r \in TTraces(A_1||A_2)\) (because \(\sigma''.r\) is \(\sigma''\) which is a timed trace of \(A_1||A_2\)). This contradicts the hypothesis of local consistency concerning \(A_1||A_2\) and thus we deduce:

\[(C) \quad : (\sigma.r)_{\Omega_i} \notin TTraces((A_1||A_2)_{\Omega_i}).\]

Since \(r\) is an output of \(\Omega_i\) or a duration, we can reformulate \((B)\) and \((C)\) respectively as:

(D) \(\sigma_{\Omega_i}.r \in TTraces(SUT_i)\)

(E) \(\sigma_{\Omega_i}.r \notin TTraces((A_1||A_2)_{\Omega_i})\)

Whose direct consequence is that \(SUT_i\) tioco \((A_1||A_2)_{\Omega_i}\) does not hold.