A Temporal Logic for Input Output Symbolic Transition Systems

Marc Aiguier, Christophe Gaston, Pascale Le Gall, Delphine Longuet, Assia Touil

A Temporal logic for Input Output Symbolic Transition Systems

Marc Aiguier[1], Christophe Gaston[2], Pascale Le Gall[1], Delphine Longuet[1] and Assia Touil[1]

[1]Université d'Évry, IBISC CNRS FRE 2890,
Tour Évry2, 523 place des terrasses, F-91000 Évry
{dlonguet,aiguier,legall}@ibisc.univ-evry.fr
fax number: (+33) 1 60 87 37 89

[2]CEA/DRT/LIST/DTSI/SLA Saclay (?)
F-91191 Gif sur Yvette Cedex
gaston@cea.fr

1 Introduction

Many works have been done to mathematically modelize reactive systems and verify their correctness. Reactive systems are open and dynamic systems whose behaviours are represented by (labelled) transition systems. Two kinds of technics are mainly used to verify correctness: model-checking or testing [6, 5]. Most of these works simply deal with system behaviours, independently of other aspects such as data. Thus, properties to be verified are expressed in propositional modal logic. Transition systems have been extended to communications and data in order to tackle communications with system environnement: this has given rise to Input Ouput Symbolic Transition Systems (IOSTS). As far as we know, no logic has been defined, whose interpretation is IOSTS. However, verification technics need logic to express requirements to be verified. In particular, properties verified by testing are either of the form of a set of finite scenarios (often called test purpose) or of given according to a simple logic in order to characterise a class of scenarios such as behavioural patterns in [1]. When dealing with testing for IOSTS, some works have succeeded in considering symbolic test purposes [4, 2, 3]. However, no work has been done to propose a logic to abstractly express properties to test.

This paper is then devoted to define a logic powerful enough to express properties of reactive systems modelized by IOSTS, mixing both data and communication actions with dynamic aspects[1]. For specifying behavior of IOSTS, we may choose to extend any possible modal logic to communications and data (for example, Hennessy-Milner logic, modal fix-point logic, Linear Temporal Logic, Computational Tree Logic, ...). In this paper, we choose CTL* which subsumes both LTL and CTL, to express properties respectively on stats and paths. The reason is that such a temporal logic allows to deal with safety, liveness and fairness properties. Our approach to extend CTL* could also be applied to other modal logics. A basic property that this logic must satisfy is adequacy [**?**], that is two bisimilar IOSTS are elementary equivalent. In this paper, we will go beyond by showing that this logic, in addition to be adequate, preserves properties along synchronized product and refinement of IOSTS.

DONNER LE PLAN DU PAPIER

2 Preliminaries

The data part addresses the functional issues of Input Ouput Symbolic Transition Systems. It will be described with a many-sorted first order logic. As usual, Σ-*terms*,

[1]This work is performed within a national French project STACS in collaboration with the Nuclear Research Center. This project is devoted to automically generate test data sets for Input Output Symbolic Systsems (IOSTS)

noted $T_\Sigma(V)$, and Σ-*formulas*, noted $Sen(\Sigma)$, are inductively built over a *many-sorted first order signature*, noted $\Sigma = (S, \mathcal{F}, R)$, and a set of *many-sorted variables*, noted $V = (V_s)_{s \in S}$. S is a set of sorts and $\mathcal{F}$ and R are respectively sets of function and relation names with arities in S.

The mathematical interpretation of any signature $\Sigma = (S, \mathcal{F}, R)$ is given by a S-set $M = (M_s)_{s \in S}$ provided with a total function $f^{\mathcal{M}} : M_{s_1} \times \cdots \times M_{s_n} \to M_s$ for each function name $f : s_1 \dots s_n \to s \in \mathcal{F}$ and a n-ary relation $r^{\mathcal{M}} : M_{s_1} \times \cdots \times M^{s_n}$ for each predicate name $r : s_1 \dots s_n \in R$. The evaluation of Σ-terms from a Σ-model $\mathcal{M}$ is given by any total function $\sigma^\natural : T_\Sigma(V) \to M$ defined as the canonical extension of any interpretation of variables $\sigma : V \to M$. Therefore, we extend any interpretation σ into an unary relation $\mathcal{M} \models_\sigma$ on Σ-formulas as usual. The validation of Σ-formulas from Σ-models is defined by: $\mathcal{M} \models \varphi$ if and only if for any $\sigma : V \to M$, $\mathcal{M} \models_\sigma \varphi$.

We denote M^V the set of mappings from V to $|\mathcal{M}|$.

3 Input Output Symbolic Transition Systems

3.1 Syntax

Input Output Symbolic Transition Systems (IOSTS) are used for modelling reactive systems. A reactive system is a system which interacts with its environment, represented itself by another IOSTS. Thus, a reactive system is an open system, defined by a IOSTS which can be also decomposed as several communicating IOSTS, each one representing one of its subsystems. Communications consist in sending or receiving messages represenTeed by first-order terms through communication channels. As usual when considering automata, IOSTS describe possible evolutions of system states. Elementary evolutions are represented by a transition relation between states. Each transition between two states is labelled by three elements: communication actions (sending or receipt of messages) or internal actions of the system, guards expressed here with first-order properties, and assignments. As usual, we start by defining the language, so-called signature, on which IOSTS are built:

Definition 3.1 (Signature) *A* signature *is a triple* $\mathscr{L} = (\Sigma, V, \mathcal{C})$ *where:* Σ *is a first-order signature,* V *is a set of variables over* Σ *and* $\mathcal{C}$ *is a set whose elements are called* channel names.

Given a signature $\mathscr{L} = (\Sigma, V, \mathcal{C})$, we can define elements that label transitions. A guard will be a first-order formula built over Σ. An assignment will be defined by a mapping $\delta : V \to T_\Sigma(V)$ preserving sorts (i.e. $\forall s \in S,\ \delta(V_s) \subseteq T_\Sigma(V)_s$) and actions are defined as follows:

$$Act_{\mathscr{L}} = \tau \mid c?x \mid c!t$$

where $c \in \mathcal{C}$, $x \in V$ and $t \in T_\Sigma(V)$. τ means an internal action while $c?x$ and $c!t$ mean, respectively, a receipt on the variable x and sending of the value t trough the channel c.

An IOSTS is then defined as follows:

Definition 3.2 (IOSTS) *Given a signature* $\mathscr{L} = (\Sigma, V, \mathcal{C})$, *an* IOSTS *is a triple* $(\mathbb{Q}, q_0, \mathbb{T})$ *where:*

- $\mathbb{Q}$ *is a set of* states
- $q_0 \in \mathbb{Q}$ *is the initial state*
- $\mathbb{T} \subseteq \mathbb{Q} \times Act_{\mathscr{L}} \times Sen(\Sigma) \times T_\Sigma(V)^V \times \mathbb{Q}$ *is a relation such that each state of* $\mathbb{Q}$ *is reachable*[2] *from* q_0.

Notation 3.1 *Note* $source : \mathbb{T} \to \mathbb{Q}$ *and* $target : \mathbb{T} \to \mathbb{Q}$ *such that for each* $t = (q, act, \varphi, \delta, q') \in \mathbb{T}$, $source(t) = q$ *and* $target(t) = q'$.

Given an IOSTS $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$, *a* path *is a word* $tr_1 \dots tr_n$ *on* $\mathbb{T}$ *such that for each* $1 \leq j < n$, $target(t_j) = source(t_{j+1})$. *Note* $Path(\mathbb{G})$ *the set of paths of* $\mathbb{G}$. *Note* $source^\natural$ *and* $target^\natural$ *the canonical extensions of* $source$ *and* $target$ *on* $Path(\mathbb{G})$.

Note $Path_q(\mathbb{G})$ *the set* $\{pa \in Path(\mathbb{G}) \mid source^\natural(pa) = q\}$.

3.2 Semantics of IOSTS

By their construction, semantics of IOSTS must take into account:

- a first-order structure $\mathcal{M}$ in order to give a mathematical meaning of data
- and a binary relation on states, which naturally are defined by variable interpretation. This relation will be the semantical meaning of transitions, and by relational composition, of paths.

[2] Reachability means: if we note $\mathbb{T}_\mathbb{Q}$ and $\mathbb{T}_\mathbb{Q}^+$ the projection of $\mathbb{T}$ on $\mathbb{Q} \times \mathbb{Q}$ and the transitive closure of $\mathbb{T}_\mathbb{Q}$, respectively, then for each $q \in \mathbb{Q} \setminus \{q_0\}$, $(q_0, q) \in \mathbb{T}_\mathbb{Q}^+$

Intuitively, semantics of paths are defined as the composition of transition semantics which depend both on guard interpretation and variable assignment. The semantics of an IOSTS will then be the set of semantics of all paths issued from the initial state.

Definition 3.3 (Semantics of IOSTS) *Let $\mathscr{L}$ be a signature and let $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$ an IOSTS on $\mathscr{L}$.*

For every $tr = (q, act, \varphi, \delta, q') \in \mathbb{T}$, note $[\![tr]\!] \subseteq M^V \times M^V$ defined by:

$(\nu^i, \nu^f) \in [\![tr]\!]$ iff:

- *$\mathcal{M} \vDash_{\nu^i} \varphi$ and $\nu^f = \nu_a^{i\natural} \circ \delta$ if $act = c?x$ and forall $y \neq x$ in V, $\nu_a^i(y) = \nu^i$*
- *$\mathcal{M} \vDash_{\nu^i} \varphi$ and $\nu^f = \nu^i$ otherwise.*

For every $pa = tr_1 tr_2 \dots tr_n$ in $Path(\mathbb{G})$, $[\![pa]\!] = [\![tr_1]\!].[\![tr_2]\!] \dots [\![tr_n]\!]$ where . is the relational composition[3]*.*

The semantics of $\mathbb{G}$, denoted $[\![\mathbb{G}]\!]$, is defined as follows:

$$[\![\mathbb{G}]\!] = \bigcup_{pa \in Path_{q_0}(\mathbb{G})} [\![pa]\!]$$

3.3 Classical operations on transition systems

3.3.1 Synchronized product

Reactive systems are often described by synchronizing subsystems together. When using IOSTS, composition of subsystems is achieved by the algebraic operation of synchronized product. This modelizes the communication by "rendez-vous". This product is informally defined as follows:

- each transition labelled by a sending through a channel c is synchronized with a transition labelled by a receipt through the same channel c,
- other transitions are asynchronous. In other words, they are fired independently.

Notation 3.2 *Let Σ be a first-order signature. Let $\varphi \in Sen(\Sigma)$. Note $\varphi[x \leftarrow t]$ the formula obtained from φ by replacing each occurrence of the free variable x by the term $t \in T_\Sigma(V)$ (of course, x and t are of the same sort).*

Definition 3.4 (Synchronized product) *Let $\mathscr{L}_1 = (\Sigma, V_1, \mathcal{C}_1)$ and $\mathscr{L}_2 = (\Sigma, V_2, \mathcal{C}_2)$ be two signatures such that $V_1 \cap V_2 = \emptyset$. Note $\mathscr{L} = (\Sigma, V_1 \cup V_2, \mathcal{C}_1 \cup \mathcal{C}_2)$. First, define the triple $(\overline{\mathbb{Q}}, \overline{q}_0, \overline{\mathbb{T}})$ as follows:*

- $\overline{\mathbb{Q}} = \mathbb{Q}_1 \times \mathbb{Q}_2$,
- $\overline{q}_0 = (q_{0_1}, q_{0_2})$
- $\overline{\mathbb{T}} \subseteq \overline{\mathbb{Q}} \times Act_{\mathscr{L}} \times Sen(\Sigma) \times T_\Sigma(V)^V \times \overline{\mathbb{Q}}$ *is the least set (according to theoretical set inclusion) such that:*
  - *if $(q_1, act, \varphi, \delta_1, q_1') \in \mathbb{T}_1$ where $act = \tau$ or is of the form $c?x$ or $c!t$ with $c \notin \mathcal{C}_1 \cap \mathcal{C}_2$, then $((q_1, q_2), act, \varphi, \delta, (q_1', q_2)) \in \overline{\mathbb{T}}$, where $\delta_{|_{V_1}} = \delta_1$ and $\delta_{|_{V_2}} = id_{V_2}$*
  - *if $(q_2, act, \varphi, \delta_2, q_2') \in \mathbb{T}_2$ where $act = \tau$ or is of the form $c?x$ or $c!t$ with $c \notin \mathcal{C}_1 \cap \mathcal{C}_2$, then $((q_1, q_2), act, \varphi, \delta, (q_1, q_2')) \in \overline{\mathbb{T}}$, where $\delta_{|_{V_1}} = id_{V_1}$ and $\delta_{|_{V_2}} = \delta_2$*
  - *if $(q_1, c!t, \varphi_1, \delta_1, q_1') \in \mathbb{T}_1$ and $(q_2, c?x, \varphi_2, \delta_2, q_2') \in \mathbb{T}_2$, then $((q_1, q_2), \tau, \varphi, \delta, (q_1', q_2')) \in \overline{\mathbb{T}}$, where $\varphi = \varphi_1 \wedge \varphi_2[x \leftarrow t]$, $\delta_{|_{V_1}} = \delta_1$ and $\delta_{|_{V_2}} = \delta_2 \circ x \mapsto t$*
  - *if $(q_1, c?x, \varphi_1, \delta_1, q_1') \in \mathbb{T}_1$ and $(q_2, c!t, \varphi_2, \delta_2, q_2') \in \mathbb{T}_2$, then $((q_1, q_2), \tau, \varphi, \delta, (q_1', q_2')) \in \overline{\mathbb{T}}$, where $\varphi = \varphi_1[x \leftarrow t] \wedge \varphi_2$, $\delta_{|_{V_1}} = \delta_1 \circ x \mapsto t$ and $\delta_{|_{V_2}} = \delta_2$.*

In order to satisfy the condition on transitions of Definition 3.2, we must cut down in the set of states $\overline{\mathbb{Q}}$ and only keep states that are reachable from $\overline{q}_0$. Hence, the synchronized product of $\mathbb{G}_1$ and $\mathbb{G}_2$, noted $\mathbb{G}_1 \otimes \mathbb{G}_2$, is the IOSTS $(\mathbb{Q}_\otimes, q_{0_\otimes}, \mathbb{T}_\otimes)$ over $\mathscr{L}$ defined by:

- $\mathbb{Q}_\otimes = \{\overline{q} \in \overline{\mathbb{Q}} | (\overline{q}_o, \overline{q}) \in \overline{\mathbb{T}}_{\overline{Q}}^+\}$
- $q_{0_\otimes} = \overline{q}_0$
- $\mathbb{T}_\otimes = \{(\overline{q}, act, \varphi, \delta, \overline{q}') \in \overline{\mathbb{T}} | (\overline{q}, \overline{q}') \in \mathbb{Q}_\otimes\}$

3.3.2 Bisimulation

Various equivalences have been studied in the litterature that identify transition systems on the basis on their behavior. The classic example is strong bisimulation denoted by $\sim$. For two given IOSTS $\mathbb{G}_1 = (\mathbb{Q}_1, q_1, \mathbb{T}_1)$ and $\mathbb{G}_2 = (\mathbb{Q}_2, q_2, \mathbb{T}_2)$, bisimulation is defined as a relation between the set of states $\mathbb{Q}_1$ and $\mathbb{Q}_2$. As relations between $\mathbb{Q}_1$ and $\mathbb{Q}_2$, they can be characterized as the greatest fixpoint $\nu F_\sim$ of a certain monotonic functional $F_\sim$. This functional operates on the complete lattice of realtions $R \subseteq \mathbb{Q}_1 \times \mathbb{Q}_2$ ordered by set inclusion and is defined by: $q\ F_\sim(R)\ q'$ iff both conditions are satisfied

[3]. is defi ned as follows : $(a, b).(b.c) = (a, c)$

- $\forall tr_1 \in \mathbb{T}_1, source(tr_1) = q \Rightarrow$
$$\exists tr_2 \in \mathbb{T}_2, \begin{cases} source(tr_2) = q' \wedge \\ [\![tr_1]\!] = [\![tr_2]\!] \wedge \\ target(tr_1)\ R\ target(tr_2) \end{cases}$$

- $\forall tr_2 \in \mathbb{T}_2, source(tr_2) = q' \Rightarrow$
$$\exists tr_1 \in \mathbb{T}_1, \begin{cases} source(tr_1) = q \wedge \\ [\![tr_1]\!] = [\![tr_2]\!] \wedge \\ target(tr_1)\ R\ target(tr_2) \end{cases}$$

The two IOSTS $\mathbb{G}_1$ and $\mathbb{G}_2$ are bisimilar, noted $\mathbb{G}_1 \sim \mathbb{G}_2$ if and only if $q_{0_1} \sim q_{0_2}$.

3.4 Refinement

3.4.1 Syntax

I0STS are mathematical abstractions of systems. We can then refine IOSTS in order to be closer and closer to the real implantation of the system. Here, refinement will only concern dynamic behavior of systems, that is transitions and paths. We suppose that data are preserved from an abstract level to a more concrete one [4] First-order signatures are then preserved in both signatures of refined and refining IOSTS. Hence, given a signature $\mathscr{L}_1 = (\Sigma_1, V_1, \mathcal{C}_1)$ and an $IOSTS\ \mathbb{G}_1 = (\mathbb{Q}_1, q_{0_1}, \mathbb{T}_1)$, a refinement of $\mathbb{G}_1$ built over $\mathscr{L}_1 = (\Sigma_1, V_1, \mathcal{C}_1)$ will be an IOSTS $\mathbb{G}_2$ over signature $\mathscr{L}_2 = (\Sigma_2, V_2, \mathcal{C}_2)$ such that $\Sigma_1 = \Sigma_2$, $V_1 \subseteq V_2$, and $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Moreover, both are equipped with the same first-order structure $\mathcal{M}$.

Transition refinement will consist in replacing a transition tr of $\mathbb{G}_1$ by an IOSTS $\mathbb{G}_{tr} = (\mathbb{Q}_{tr}, q_{0_{tr}}, \mathbb{T}_{tr})$. Three conditions have to be imposed on $\mathbb{G}_{tr}$:

1. $source(tr)$ is the initial state of $\mathbb{G}_{tr}\mathbb{G}_t$.
2. $target(tr)$ is reachable from each state of $\mathbb{G}_{tr}$.
3. Finally, each path of $\mathbb{G}_{tr}$ must only contain the action which occurs in tr and no other ones of $\mathscr{L}_1$.

Syntactically, a transition refinement is then defined as follows:

[4] There are many works that have be done on data refinement by using algebraic techiques. A very good suvey on this subject can be found in [**?**]. Here, we do not consider such a refinement in order to be more comprehensive. However, such a refinement combining together data and dynamic behavior refinement can be found in [**?**, **?**].

Definition 3.5 (Syntactical refinement of a transition) *Let $\mathbb{G}$ be an IOSTS over $\mathscr{L} = (\Sigma, V, \mathcal{C})$. Let $tr = (q, act, \varphi, \delta, q') \in \mathbb{T}_1$ be a transition. A* syntactical refinement *of tr is an IOSTS $\mathbb{G}_{tr} = (\mathbb{Q}_{tr}, q_{0_{tr}}, \mathbb{T}_{tr})$ over $\mathscr{L}_{tr} = (\Sigma, V_{tr}, \mathcal{C}_{tr})$ such that:*

- $\mathbb{Q}_{tr} \cap \mathbb{Q}_1 = \{q, q'\}$
- $q_{0_{tr}} = q$
- *for each $q'' \in \mathbb{Q}_{tr}$, there exists $pa \in Path_{q''}(\mathbb{G}_{tr})$ such that $target^\natural(pa) = q'$*
- *for each $pa = tr_1 \dots tr_n \in Path_q(\mathbb{G}_{tr})$ with $target^\natural(pa) = q'$, there exists a unique $1 \leq k \leq n$ such that the action of t_k is act, and for each $1 \leq j \neq k \leq n$, the action of t_j is either τ or uses a channel name in $\mathcal{C}_{tr} \setminus \mathcal{C}$.*

Remark. A transition $tr = (q, act, \varphi, \delta, q')$ can also be considered as an IOSTS $\mathbb{G}_{tr}^{Id} = (\mathbb{Q}_{tr}, q_{0_{tr}}, \mathbb{T}_{tr})$ where $\mathbb{Q}_{tr} = \{q, q'\}$, $q_{0_{tr}} = q$ and $\mathbb{T}_{tr} = \{tr\}$. By Definition 3.5, $\mathbb{G}_{tr}^{Id}$ is a syntactical refinement of itself.

Syntactical refinement of an IOSTS is then defined as follows:

Definition 3.6 (Syntactical refinement of an IOSTS) *A* syntactical refinement *of $\mathbb{G}_1 = (\mathbb{Q}_1, q_1, \mathbb{T}_1)$ is an IOSTS $\mathbb{G}_2 = (\mathbb{Q}_2, q_2, \mathbb{T}_2)$ defined from a $\mathbb{T}_1$-indexed family $(\mathbb{G}_{tr})_{tr \in \mathbb{T}_1}$ where[5] $\mathbb{G}_{tr}$ is a syntactical refinement of tr, as follows:*

- $\mathbb{Q}_2 = \bigcup_{tr \in \mathbb{T}_1} \mathbb{Q}_{tr}$
- $q_{0_2} = q_{0_1}$
- $\mathbb{T}_2 = \bigcup_{tr \in \mathbb{T}_1} \mathbb{T}_{tr}$

A refinement of $\mathbb{G}_1$ is then an IOSTS composed of the refinements of all the transitions of $\mathbb{G}_1$.

Remark. We deduce from Definition 3.5 and Definition 3.6 that $\mathbb{Q}_1 \subseteq \mathbb{Q}_2$ and $\mathbb{T}_1 \subseteq \mathbb{T}_2$.

[5] If $\mathbb{G}_{tr}$ is the IOSTS $\mathbb{G}_{tr}^{Id}$, then it simply means that the corresponding transition tr is not refined.

3.4.2 Correctness

Refinement correctness holds when refinement IOSTS completely preserves dynamic behavior of refined one. Formally, this is expressed as follows:

Definition 3.7 (Refinement correctness) *Let $\mathbb{G}_2$ be a syntactical refinement of $\mathbb{G}_1$. This refinement is* correct *if and only if $U([\![\mathbb{G}_2]\!]) = [\![\mathbb{G}_1]\!]$ where $U([\![\mathbb{G}_2]\!])$ means:*

$$U([\![\mathbb{G}_2]\!]) = \{(\nu^i_{|_{V_1}}, \nu^f_{|_{V_1}})|(\nu^i, \nu^f) \in [\![\mathbb{G}_2]\!]\}$$

Of course, it is not reasonable to refine an IOSTS as a whole in a single step. Large softwares usually require many refinement steps before obtaining efficient programs. This leads to the notion of sequential composition of refinement steps. Usually, composition of enrichment is mainly divided into two concepts: horizontal composition, and vertical composition.

Horizontal composition deals with refinement of subparts of systems when they are structured into "blocks". Here, blocks are IOSTS and structuration is defined by synchronized product. On the contrary, vertical composition deals with many refinement steps, that is it is the transitive closure of correct refinements. In both cases, correctness is preserved. For lack of space, we do not present these results. However, they can be found in [**?**, **?**]

4 A temporal logic for IOSTS

We present in this section a first-order temporal logic $\mathcal{F}$ interpretation of which will be over IOSTS. $\mathcal{F}$ extends CTL^* [**?**] to first-order in order to take into account messages passing in actions by adding the modality $\mathbf{after}[a]$ where a is a finite sequence of actions. $\mathbf{after}[a]\varphi$ roughly means from the current sequence of transitions σ that φ is satisfied for the subsequence of σ that directly follows the sequence a in σ. Observe **after**[a] is the extension to paths of the modality $[a]$ of the standard Hennessy-Milner logic [**?**]. Hence, $\mathcal{F}$ is a branching-time temporal logic where the structure representing all possible executions is *tree-like* rather than linear.

4.1 Syntax

As interpretation of $\mathcal{F}$ is over IOSTS, signatures are the ones of Definition 3.1. Actions are extendes in order to consider finite sequences of actions.

Hence, actions are defined as $Act_{\mathscr{L}}$ for $\mathscr{L}$ a signature, at which we add the production $Act_{\mathscr{L}}; Act_{\mathscr{L}}$. By the associativity property, a is a sequence of elementary actions $a = a_1; \ldots; a_n$ where for each $1 \leq i \leq n$, a_i denotes internal action, receipt or sending.

Definition 4.1 (Formulae) *Let $\mathscr{L} = (\Sigma, V, \mathcal{C})$ be a signature. Formulae are defined as follows:*

$$\begin{aligned} For := \;& Sen(\Sigma)|\ \mathbf{after}[Act_{\mathscr{L}}]For|\alpha For| \\ & For\ \mathbf{U}\ For|\forall For|\exists For|\ \neg For|\ For\beta For \end{aligned}$$

where $\alpha \in \{\mathbf{X}, \mathbf{F}, \mathbf{G}\}$ and $\beta \in \{\vee, \wedge, \Rightarrow\}$

4.2 Semantics

As already said above, formulae are interpreted over IOSTS. Of course, IOSTS and formulae must be built over a same language $\mathscr{L}$. Before giving satisfaction of formulae, we have first to define the notion of term embedding in paths of a given IOSTS. The satisfaction of fromulae of the form $\mathbf{after}[a]\varphi$ will be based on this notion.

Definition 4.2 (Embedding of a term in a path) *Let $a = a_1; \ldots; a_n$ be a term. Let $pa = tr_1 \ldots tr_m \in Path(\mathbb{G})$ be a path where $m \geq n$ and for each $1 \leq i \leq m$, $tr_i = (q_i, act_i, \varphi_i, \delta_i, q'_i)$. a is said* embedded *into pa if and only if there exists a sequence $(i_1, \ldots, i_n)$ where for every $1 \leq j \leq n$ $i_j \in \{1, \ldots, m\}$, $i_j < i_{j+1}$ and $i_n = m$, such that for every $1 \leq l \leq n$, $a_l = act_{i_l}$.*

In IOSTS, only paths starting from the intial state make sense. Therefore, formula satisfaction will only be defined from sequence of actions the source of which is q_0, and variable interpretations. This gives rise to the following definition:

Definition 4.3 (Satisfaction) *Let $\mathscr{L}$ be a signature. Let $\mathbb{G}$ be an IOSTS over $\mathscr{L}$ together with $\mathcal{M}$ as underlying first-order structure. Let φ be a formula over $\mathscr{L}$. Let $\sigma = (tr_0, \ldots, tr_n, \ldots)$ be a sequence of actions of $\mathbb{G}$, so-called* run*, satisfying: $\forall i \in \mathbb{N}$, $target(tr_i) = source(tr_{i+1})$. Let $\nu : V \to \mathcal{M}$ be an interpretation of variables.* $\mathbb{G}$ satisfies *for σ and ν the formula φ, noted $\mathbb{G} \models_{\sigma,\nu} \varphi$ if and only if: for every $i \in \mathbb{N}$, noted $\sigma^i = (tr_i, \ldots, tr_n, \ldots)$ the subsequence of σ.*

- *if $\varphi \in Sen(\Sigma)$, then $\mathbb{G} \models_{\sigma,\nu} \varphi$ iff $\mathcal{M} \models_\nu \varphi$,*

- *if φ is of the form* $\mathbf{after}[a]\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff there exists* $i \in \mathbb{N}$ *such that* a *is embedded in* $pa = (tr_0, \ldots, tr_{i-1})$ *and for every* $(\nu, \nu') \in [\![pa]\!]$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,
- *if φ is of the form* $\mathbf{X}\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff for every* $(\nu, \nu') \in [\![tr_1]\!]$, $\mathbb{G} \models_{\sigma^1,\nu'} \psi$,
- *if φ is of the form* $\mathbf{F}\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff there exists* $i \in \mathbb{N}$ *such that for every* $(\nu, \nu') \in [\![tr_0 \ldots tr_{i-1}]\!]$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,
- *if φ is of the form* $\mathbf{G}\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff for every* $i \in \mathbb{N}$ *and for every* $(\nu, \nu') \in [\![tr_0 \ldots tr_{i-1}]\!]$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,
- *if φ is of the form* $\psi \mathbf{U} \chi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff there exists* $i \in \mathbb{N}$ *such that for every* $(\nu, \nu') \in [\![tr_0 \ldots tr_{i_1}]\!]$ $\mathbb{G} \models_{\sigma^i,\nu'} \chi$ *and for every* $1 \leq k < j$ *and every* $(\nu, \nu') \in [\![tr_0 \ldots tr_{k-1}]\!]$, $\mathbb{G} \models_{\sigma^k,\nu'} \psi$,
- *if φ is of the form* $\forall\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff for every run* σ' *sharing the same initial state with* σ, $\mathbb{G} \models_{\sigma',\nu} \psi$,
- *if φ is of the form* $\exists\psi$*, then* $\mathbb{G} \models_{\sigma,\nu} \varphi$ *iff there exists a run* σ' *sharing the same initial state with* σ, $\mathbb{G} \models_{\sigma',\nu} \psi$,
- *propositional connectives are handled as usual.*

Note $\mathbb{G} \models \varphi$ *if and only if for every run* σ *statrting to* q_0 *and every interpretation* ν $\mathbb{G} \models_{\sigma,\nu} \varphi$.

4.3 Preservation results

In this section, we establish three results which show that $\mathcal{F}$ is well-adapted to express properties on IOSTS. For lack of space, we do not give their proofs. For interested readers, they can be found in [**?**, **?**].

4.3.1 Synchronized product

Synchronized product restricts IOSTS behavior. Therefore, preservation cannot hold for all formulae. It can only hold for a subset of them. Actually, all formulae implicitely dealing with existness quantifiers such as both modalities **F**, **U**, and $\exists$ do not preserve properties along synchronized product. This subset of formulae is defined as follows:

$$\begin{aligned} For' := \; & Sen(\Sigma) | \; \mathbf{after}[Act_{\mathscr{L}}]For' | \alpha For' | \\ & \forall For | \; For \beta For \end{aligned}$$

where $\alpha \in \{\mathbf{X}, \mathbf{G}\}$ and $\beta \in \{\wedge, \Rightarrow\}$.
Before expressing this preservation result, note $\_^\bullet$ the mapping that transforms every action over two signatures $\mathscr{L}_1 = (\Sigma, V_1, \mathcal{C}_1)$ and $\mathscr{L}_2 = (\Sigma, V_2, \mathcal{C}_2)$ into an action over $\mathscr{L} = (\Sigma, V_1 \cup V_2, \mathcal{C}_1 \cup \mathcal{C}_2)$ as follows:

$$\begin{array}{ll} \tau \mapsto \tau & \\ c\#u \mapsto \tau & \text{if } c \in \mathcal{C}_1 \cap \mathcal{C}_2 \\ c\#u \mapsto c\#u & \text{otherwise} \end{array}$$

where $\# \in \{?, !\}$ and $u \in T_\Sigma(V_i)$ $i = 1, 2$. Note also $\_^\bullet$ its canonical extension to formulae defined as follows:

$$\begin{array}{l} \varphi \in Sen(\Sigma) \mapsto \varphi \\ \mathbf{after}[a]\varphi \mapsto \mathbf{after}[a^\bullet]\varphi^\bullet \\ @\varphi \mapsto @\varphi^\bullet \\ \varphi \; \mathbf{U} \; \psi \mapsto \varphi^\bullet \; \mathbf{U} \; \psi^\bullet \end{array}$$

where $@ \in \{\mathbf{X}, \mathbf{G}\}$

Theorem 4.1 *Let* $\mathbb{G}_i$ *be an IOSTS over* $\mathscr{L}_i = (\Sigma, V_i, \mathcal{C}_i)$ *for* $i = 1, 2$ *such that* $V_1 \cap V_2 = \emptyset$. *Let* φ *be a formula over* $\mathscr{L} = (\Sigma, V_1 \cup V_2, \mathcal{C}_1 \cup \mathcal{C}_2)$ *that satisfies production rules of* For'. *Then, we have:*

$$\mathbb{G}_1 \models \varphi \wedge \mathbb{G}_2 \models \varphi \Rightarrow \mathbb{G}_1 \otimes \mathbb{G}_2 \models \varphi^\bullet$$

4.3.2 Adequacy

In a modal logic $\mathbb{L}$ interpreted over symbolic transition systems $(\mathbb{Q}, q, \mathbb{T})$, $\mathbb{L}$ is said *adequate* w.r.t. a binary relation $\mathcal{R}$ on $\mathbb{Q}$ (which is usually the strong bisimilarity relation) if and only if

$$\forall \mathbb{G}_1, \mathbb{G}_2, (\forall \varphi, \mathbb{G}_1 \models \varphi \Leftrightarrow \mathbb{G}_2 \models \varphi) \Longleftrightarrow \mathbb{G}_1 \sim \mathbb{G}_2$$

Theorem 4.2 $\mathcal{F}$ *is adequate w.r.t.* $\sim$.

4.3.3 Refinement

Refinement correctness as defined in Definition 3.7 expresses that the refining IOSTS meets all properties of the refined IOSTS. Indeed, we can show the following result:

Theorem 4.3 *Let* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *be two IOSTS built respectively over* $\mathscr{L}_1$ *and* $\mathscr{L}_2$. *Assuming that* $\mathbb{G}_2$ *is a correct refinement of* $\mathbb{G}_1$. *Then, for every formula* φ *built over* $\mathscr{L}_1$ *we have:*

$$\mathbb{G}_1 \models \varphi \Longleftrightarrow \mathbb{G}_2 \models \varphi$$

5 Conclusion

In this paper, we have defined a logic dedicated to express properties on IOSTS. This logic has been defined as an extension of CTL^* to take into account communications and data. Moreover, we establish appropriate properties on it such adequacy w.r.t. strong bissimulation, and preservation of properties along refinement.
We are currently investigating how to automatically genetate test cases from test purposes given by properties in $\mathcal{F}$. We are also investigating how to test conformance between a more concrete IOSTS w.r.t. an abstract one. This will be based on the refinement relation as presented in this paper.

References

[1] L. du Bousquet, F. Ouabdesselam, J.-L. Richier, and N. Zuanon. Feature interaction detection using synchronous approach and testing. *Computer Networks and ISDN Systems*, 11(4):419–446, 2000.

[2] L. Frantzen, J. Tretmans, and T. A. Willemse. Test generation based on symbolic specifi cations. In J. Grabowski and B. Nielsen, editors, *FATES 2004*, number 3395 in LNCS, pages 1–15. Springer-Verlag, 2005.

[3] B. Jeannet, T. J´eron, V. Rusu, and E. Zinovieva. Symbolic test selection based on approximate analysis. In *11th Int. Conference on Tools and Algorithms for tthe Construction and Analysis of Systems (TACAS)*, Edinburgh, Scottland, April 2005.

[4] V. Rusu, L. du Bousquet, and T. J´eron. An approach to symbolic test generation. In *IFM '00: Proceedings of the Second International Conference on Integrated Formal Methods*, pages 338–357, London, UK, 2000. Springer-Verlag.

[5] J. Tretmans. Conformance Testing with Labelled Transition Systems: Implementation Relations and Test Generation. *Computer Networks and ISDN Systems*, 29:49–79, 1996.

[6] M. Yannakakis and D. Lee. Testing fi nite state machines. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 476–485. ACM Press, 1991.