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, LaMI CNRS UMR 8042,
Tour Évry2, 523 place des terrasses, F-91000 Évry
{aiguier,legall,dlonguet,atouil}@lami.univ-evry.fr
fax number: (+33) 1 60 87 37 89

[2]CEA/LIST Saclay
F-91191 Gif sur Yvette Cedex
Christophe.Gaston@cea.fr

Abstract— **In this paper, we present a temporal logic called $\mathcal{F}$ whose interpretation is over Input Output Symbolic Transition Systems (IOSTS). IOSTS extend transition systems to communications and data in order to tackle communications with system environment. $\mathcal{F}$ is then defined as an extension of temporal logic CTL^* (a temporal logic which mixes together the features of Linear Temporal Logic (LTL) and Computational Temporal Logic (CTL)). Three basic properties are established on $\mathcal{F}$: adequacy and preservation of properties along synchronized product and IOSTS refinement.**

Keywords: input output symbolic transition systems, temporal logic, strong bisimulation, refinement, adequacy

I. Introduction

Many works have been done to mathematically model reactive systems and verify their correctness. Reactive systems are open and dynamic systems whose behaviours are formally represented by (labelled) transition systems. Two kinds of techniques are mainly used to verify correctness of such systems: model-checking or testing [1], [2]. Most of these works simply deal with system behaviours, independently of other aspects such as data. Thus, properties under verification are expressed in propositional modal logic. Recently, in testing context, transition systems have been extended to communications and data in order to tackle communications with system environment: this gave rise to Input Output Symbolic Transition Systems (IOSTS) [3]–[5]. As far as we know, no logic whose interpretation is over IOSTS has been defined. However, verification techniques 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 expressed in a simple logic in order to characterize a class of scenarios such as behavioural patterns [6]. When dealing with conformance testing for IOSTS[1], some works succeeded considering symbolic test purposes [3]–[5]. However, no work has been done to propose a logic that can abstractly express properties to test.

This paper is then devoted to define a logic powerful enough to express properties of reactive systems represented by IOSTS, mixing both data and communication actions with dynamic aspects[2]. To specify the behaviour of IOSTS, we may choose to extend any possible modal logic to communications and data (e.g. Hennessy-Milner logic [7], modal fix-point logic [8], Linear Temporal Logic (LTL) [9], Computational Tree Logic (CTL) [10]. . .). In this paper, we choose CTL* [11] which mixes together the features of both LTL and CTL, to express properties on states and paths respectively. 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 [7], that is when two bisimilar IOSTS are elementary equivalent. In this paper, we will go beyond that, showing that this logic, in addition to be adequate, preserves properties along synchronized product

[1] Conformance testing consists in showing that an implementation meets all the requirements of its specification when both are formally specified by transition systems.

[2] This work is performed within a national French project STACS (*Spécification et Test, Abstraits et Compositionnels, de Systèmes*) in collaboration with the Nuclear Research Center (CEA). This project is devoted to automatically generate test data sets for Input Output Symbolic Transition Systems (IOSTS).

and refinement of IOSTS.

The paper is organized as follows. In Section II, we recall basic definitions and notations about many-sorted first-order logic. In Section III, we introduce IOSTS and define the three operations on IOSTS: synchronized product, strong bisimulation and refinement. In Section IV, we present a temporal logic whose interpretation is over IOSTS. Moreover, we give three results that express respectively that this logic is adequate, and preserves properties along synchronized product and refinement.

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

The mathematical interpretation of any signature $\Sigma = (S, F, R)$ is given by a S-set $M = (M_s)_{s \in S}$ provided with a total function $f^{\mathcal{M}} : M_{s_1} \times \cdots \times M_{s_n} \rightarrow M_s$ for each function name $f : s_1 \dots s_n \rightarrow s \in F$ 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 $\sigma^\natural : T_\Sigma(V) \rightarrow M$ defined as the canonical extension of any interpretation of variables $\sigma : V \rightarrow M$. 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 $\sigma : V \rightarrow M$, $\mathcal{M} \models_\sigma \varphi$.

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

III. Input Output Symbolic Transition Systems

A. Syntax

Input Output Symbolic Transition Systems (IOSTS) are used to model 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 an IOSTS which can also be decomposed into several communicating IOSTS, each one representing one of its subsystems. Communications consist in sending or receiving messages represented 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 III.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: guard, assignment and actions. A *guard* will be a first-order formula built over Σ. An *assignment* will be defined by a mapping $\delta : V \rightarrow T_\Sigma(V)$ 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 represents an internal action while $c?x$ and $c!t$ represent, respectively, a receipt on the variable x and sending of the value t through the channel c.

An IOSTS is then defined as follows:

Definition III.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*[3] *from* q_0.

Example III.1 *All through this paper, we are going to take the example of a cash dispenser. Its informal specification is the following. A user inserts his card and keys his code. If it is wrong, the user has to key his code again, except if it is the third time that the code is wrong. In this case, the user does not get his card back, and the dispenser is reinitialized. If the code is valid, the user keys the amount he wants to withdraw. Then the dispenser gives an authorization depending on the card number and the asked amount. According to this authorization, the dispenser will give or not his card back to the user, and will give or not his money. In all these cases, when the operation is finished, the dispenser is reinitialized.*

The $isvalid$ *function checks the validity of the code. The* $authorize$ *function gives an authorization (0, 1 or 2) according to the card number and the asked amount.*

An IOSTS modelling such a system is shown on figure 1.

[3] 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 $q \in \mathbb{Q} \setminus \{q_0\}$, $(q_0, q) \in \mathbb{T}_\mathbb{Q}^+$.

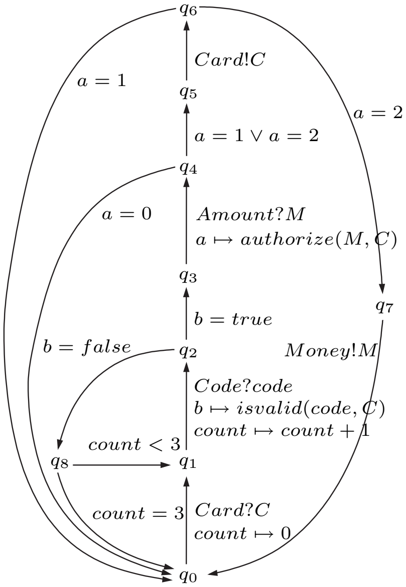

Fig. 1. A cash dispenser

Notation III.1 *Note $source : \mathbb{T} \rightarrow \mathbb{Q}$ and $target : \mathbb{T} \rightarrow \mathbb{Q}$ 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 $\{pa \in Path(\mathbb{G}) \mid source^\natural(pa) = q\}$.

B. 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.

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

Definition III.3 (Semantics of IOSTS) *Let $\mathscr{L}$ be a signature. Let $\mathbb{G} = (\mathbb{Q}, q_0, \mathbb{T})$ be an IOSTS over $\mathscr{L}$ whose first-order structure is $\mathcal{M}$.*

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:

- *$\mathcal{M} \models_{\nu^i} \varphi$ and $\nu^f = {\nu_a^i}^\natural \circ \delta$ if $act = c?x$ and for all $y \neq x$ in V, $\nu_a^i(y) = \nu^i$*
- *$\mathcal{M} \models_{\nu^i} \varphi$ 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[4]*.*

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

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

C. Classical operations on transition systems

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 models communications 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 III.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 III.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 $\delta_{|_{V_2}} = \delta_2 \circ x \mapsto t$*

[4] . is defined as follows : $(a, b).(b, c) = (a, c)$

- *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 III.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:*

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

2) Bisimulation: Various equivalences have been studied in the literature that identify transition systems on the basis of their behaviour. 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 a relation between $\mathbb{Q}_1$ and $\mathbb{Q}_2$, it 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 relations $R \subseteq \mathbb{Q}_1 \times \mathbb{Q}_2$ ordered by set inclusion and is defined by: $q \; F_\sim(R) \; q'$ iff both following conditions are satisfied:

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

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

D. Refinement

1) Syntax: IOSTS 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 behaviour of systems, that is transitions and paths. We suppose that data are preserved from an abstract level to a more concrete one[5]. 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 $\mathbb{G}_{tr}$.
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:

Definition III.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* $\mathcal{C}_{tr} \setminus \mathcal{C}$.

Example III.2 *We are going to refine one of the transitions of the IOSTS presented in example III.1. We refine it making more explicit what the* $authorize$ *function does. The dispenser calls the bank to first check the date of validity of the card. If it is over, then the dispenser gives authorization 0. If the card is valid, then the dispenser asks the bank the total amount available on the user's account. If the amount the user wants to withdraw is available, he is given authorization 2, and if it is not, he is given authorization 1. An IOSTS refining this transition is shown on figure 2*

Remark. A transition $tr = (q, act, \varphi, \delta, q')$ can also be considered as an IOSTS $\mathbb{G}^{Id}_{tr} = (\mathbb{Q}_{tr}, q_{0_{tr}}, \mathbb{T}_{tr})$ where $\mathbb{Q}_{tr} = \{q, q'\}$, $q_{0_{tr}} = q$ and $\mathbb{T}_{tr} = \{tr\}$. By Definition III.5, $\mathbb{G}^{Id}_{tr}$ is a syntactical refinement of tr.

Syntactical refinement of an IOSTS is then defined as follows:

Definition III.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*

[5] There are many works that have been done on data refinement by using algebraic techniques. A very good survey on this subject can be found in [12]. Here, we do not consider such a refinement to lighten the paper. However, such a refinement combining together data and dynamic behaviour refinement can be found in [13], [14].

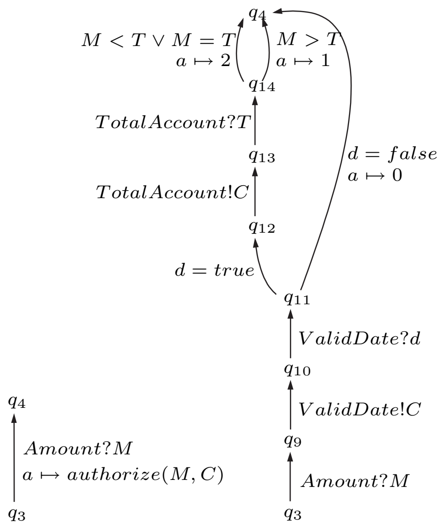

Fig. 2. Syntactical refinement of a transition

$(\mathbb{G}_{tr})_{tr \in \mathbb{T}_1}$ where[6] $\mathbb{G}_{tr}$ 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 III.5 and Definition III.6 that $\mathbb{Q}_1 \subseteq \mathbb{Q}_2$ and $\mathbb{T}_1 \subseteq \mathbb{T}_2$.

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

Definition III.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 structuring 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, we have shown that correctness is preserved. For lack of space, we do not present these results. However, they can be found in [13], [14].

IV. A TEMPORAL LOGIC FOR IOSTS

We present in this section a first-order temporal logic $\mathcal{F}$ whose interpretation will be over IOSTS. $\mathcal{F}$ extends CTL^* [11] to first-order in order to take into account communication actions 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 that $\mathbf{after}[a]_{-}$ is the extension to paths of the modality $[a]_{-}$ of the standard Hennessy-Milner logic [7]. Hence, $\mathcal{F}$ is a branching-time temporal logic where the structure representing all possible executions is *tree-like* rather than linear.

A. Syntax

As interpretation of $\mathcal{F}$ is over IOSTS, signatures are those of Definition III.1. Actions are extended in order to consider finite sequences of actions.

Hence, actions are defined as $Act_{\mathscr{L}}$ for $\mathscr{L}$ a signature, to 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 IV.1 (Formulas) *Let $\mathscr{L} = (\Sigma, V, \mathcal{C})$ be a signature. Formulas are defined as follows:*

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

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

Example IV.1 *We give here some formulas we can express on the IOSTS of example III.1.*
After the user's card is inserted, the counter is initialized:

$$\mathbf{after}[Card?C](count = 0)$$

If the code is valid, the authorization given after the receipt of the amount will be 0, 1 and 2:

$$\mathbf{F}(b = true) \Rightarrow \mathbf{after}[Amount?M](a = 0 \vee a = 1 \vee a = 2)$$

If the given authorization is 1, then the variable a will keep this value at least until the counter is reinitialized:

$$\mathbf{F}(a = 1) \Rightarrow (a = 1)\mathbf{U}(count = 0)$$

The counter is always greater than or equal to 0 and less than or equal to 3:

$$\mathbf{G}(\neg(count < 0 \vee count > 3))$$

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

If the given authorization is 2, then the user will be given his card back and his money:

$$\mathbf{F}(a = 2) \Rightarrow \mathbf{after}[Card!C; Money!M]\top$$

where $\top$ stands for the always true formula.

B. Semantics

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

Definition IV.2 (Embedding of a term in a path) *Let $a = a_1; \dots; a_n$ be a term. Let $pa = tr_1 \dots tr_m \in Path(\mathbb{G})$ 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 $(i_1, \dots, i_n)$ such that for every $1 \leq j \leq n$, $i_j \in \{1, \dots, m\}$, $i_j < i_{j+1}$, $i_n = m$, and $a_j = act_{i_j}$.*

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

Definition IV.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 $\sigma = (tr_0, \dots, tr_n, \dots)$ 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 $i \in \mathbb{N}$, note $\sigma^i = (tr_i, \dots, tr_n, \dots)$ 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 $pa = (tr_0, \dots, tr_{i-1})$ and for every $(\nu, \nu') \in \llbracket pa \rrbracket$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,*
- *if φ is of the form $\mathbf{X}\psi$, then $\mathbb{G} \models_{\sigma,\nu} \varphi$ iff for every $(\nu, \nu') \in \llbracket tr_0 \rrbracket$, $\mathbb{G} \models_{\sigma^1,\nu'} \psi$,*
- *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 $(\nu, \nu') \in \llbracket tr_0 \dots tr_{i-1} \rrbracket$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,*
- *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 $(\nu, \nu') \in \llbracket tr_0 \dots tr_{i-1} \rrbracket$, $\mathbb{G} \models_{\sigma^i,\nu'} \psi$,*
- *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 $(\nu, \nu') \in \llbracket tr_0 \dots tr_{i-1} \rrbracket$ $\mathbb{G} \models_{\sigma^i,\nu'} \chi$ and for every $1 \leq k < i$ and every $(\nu, \nu') \in \llbracket tr_0 \dots tr_{k-1} \rrbracket$, $\mathbb{G} \models_{\sigma^k,\nu'} \psi$,*
- *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.*

$\mathbb{G}$ satisfies *φ, noted $\mathbb{G} \models \varphi$ if and only if for every run σ starting at q_0 and every interpretation ν, $\mathbb{G} \models_{\sigma,\nu} \varphi$.*

C. 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 here. For interested readers, they can be found in [13], [14].

1) Synchronized product: Synchronized product restricts IOSTS behaviour. Therefore, preservation cannot hold for all formulas. It can only hold for a subset of them. Actually, all formulas implicitly dealing with existness quantifiers such as the modalities $\mathbf{F}$, $\mathbf{U}$, and $\exists$ do not preserve properties along synchronized product. This subset of formulas is defined as follows:

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

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 the 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:

$$\begin{array}{ll} \tau \mapsto \tau & \\ c\#u \mapsto \tau & \text{if } c \in \mathcal{C}_1 \cap \mathcal{C}_2 \\ c\#u \mapsto c\#u & \text{if } c \notin \mathcal{C}_1 \cap \mathcal{C}_2 \\ a_1; a_2 \mapsto a_1^\bullet; a_2^\bullet & \end{array}$$

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

$$\begin{array}{l} \varphi \in Sen(\Sigma) \mapsto \varphi \\ \mathbf{after}[a]\varphi \mapsto \mathbf{after}[a^\bullet]\varphi^\bullet \\ \alpha\varphi \mapsto \alpha\varphi^\bullet \\ \forall\varphi \mapsto \forall\varphi^\bullet \\ \varphi\ \beta\ \psi \mapsto \varphi^\bullet\ \beta\ \psi^\bullet \end{array}$$

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

Theorem IV.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:*

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

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 IV.2 *$\mathcal{F}$ is adequate w.r.t. $\sim$.*

3) Refinement: Refinement correctness as defined in Definition III.7 expresses the fact that the refining IOSTS meets all properties of the refined IOSTS, except those dealing with too specific states, like the modality $\mathbf{X}$ which unables to express a property about the next state. Indeed, if the transition leading to this next state is refined by an IOSTS, then the property will be verified in *a* next state but not in *the* next state. Then we need to restrict the initial set of formulas 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 $\alpha \in \{\mathbf{F}, \mathbf{G}\}$ and $\beta \in \{\vee, \wedge, \Rightarrow\}$.

We can now show the following result for this set of formulas:

Theorem IV.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 $\mathscr{L}_1$ satisfying production rules of For'' we have:*

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

V. 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 have established appropriate properties on it such as adequacy w.r.t. strong bisimulation, and preservation of properties along refinement.

We are currently investigating how to automatically generate 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] M. Yannakakis and D. Lee, “Testing finite state machines,” in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. ACM Press, 1991, pp. 476–485.

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

[3] V. Rusu, L. du Bousquet, and T. Jéron, “An approach to symbolic test generation,” in *IFM ’00: Proceedings of the Second International Conference on Integrated Formal Methods*. London, UK: Springer-Verlag, 2000, pp. 338–357.

[4] L. Frantzen, J. Tretmans, and T. A. Willemse, “Test generation based on symbolic specifications,” in *FATES 2004*, ser. LNCS, J. Grabowski and B. Nielsen, Eds., no. 3395. Springer-Verlag, 2005, pp. 1–15.

[5] B. Jeannet, T. Jéron, 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.

[6] L. du Bousquet, F. Ouabdesselam, J.-L. Richier, and N. Zuanon, “Feature interaction detection using synchronous approach and testing,” *Computer Networks and ISDN Systems*, vol. 11, no. 4, pp. 419–446, 2000.

[7] M. Hennessy and R. Milner, “Algebraic laws for nondeterminism and concurrency,” *Journal of the ACM*, vol. 32, no. 1, pp. 177–161, 1985.

[8] D. Kozen, “Results on the propositional mu-calculus,” *Theoretical Computer Science*, vol. 27, pp. 333–354, 1983.

[9] A. Pnueli, “The temporal logic of programs,” in *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*. ACM, 1977, pp. 46–77.

[10] E. M. Clarke and E.-A. Emerson, *Logics of Programs*. Springer, 1981, ch. Design and synthesis of synchronisation skeletons using branching time temporal logics, pp. 52–71.

[11] E.-A. Emerson, *Handbook of Theoretical Computer Science*. Elsevier, 1990, ch. Temporal and Modal Logic, pp. 995–1073.

[12] H. Ehrig and H. Kreowski, *Algebraic Foundations of Systems Specification*, ser. IFIP State-of-the-Art Reports. Springer, 1999, ch. Refinement and implementation, pp. 201–243.

[13] D. Longuet, “Une théorie du raffinement orientée propriétés pour les automates communicants,” Master’s thesis, University of Evry, 2004, available at http://www.lami.univ-evry.fr/∼dlonguet/.

[14] M. Aiguier, C. Gaston, P. L. Gall, D. Longuet, and A. Touil, “A temporal logic for input output symbolic transistion systems,” University of Evry, Tech. Rep., 2005, available at http://www.lami.univ-evry.fr/∼dlonguet/.