Property preserving simulations[*]

S. Bensalem A. Bouajjani C. Loiseaux J. Sifakis

IMAG-LGI, BP 53X, F-38041 Grenoble
e-mail: {bensalem,bouajjan,loiseaux,sifakis}@imag.imag.fr

Abstract. We study property preserving transformations for reactive systems. A key idea is the use of $<\varphi,\psi>$-simulations which are simulations parameterized by a Galois connection (φ,ψ), relating the lattices of properties of two systems.

We propose and study a notion of preservation of properties expressed by formulas of a logic, by a function φ mapping sets of states of a system S into sets of states of a system S'. Roughly speaking, φ preserves f if the satisfaction of f at some state of S implies that f is satisfied by any state in the image of this state by φ.

The main results concern the preservation of properties expressed in sublanguages of the branching time μ-calculus when two systems S and S' are related via $<\varphi,\psi>$-simulations. They can be used in particular to verify a property for a system by proving this property on a simpler system which is an abstraction of it.

1 Introduction

A central idea in a rigorous program development methodology is that a designer starting from a formal requirements specification obtains an implementation by performing successive semantics preserving transformations. In this process, semantics can be expressed by a set of properties specified as formulas of an appropriate program logic; the transformations applied should preserve these properties.

The investigation of property preserving implementations or abstractions of reactive systems has been the object of intensive research during the last years. However, the existing theoretical results are too fragmented. They strongly depend on the choice of the specification formalism and the underlying semantics.

Some results [Kur89,AL88,LT88] adopt a linear time semantics framework where specifications are the conjunction of a safety and a liveness property. The safety part is usually expressed as a transition relation (automaton) while the liveness constraint is characterized either by an acceptance condition on the states or by a formula. The notions of implementation proposed are based on the use of structure or language homomorphisms preserving correctness.

For process algebras the problem of the adequacy of a logic used to express process properties, has been studied [HM85,GS86b,GS86a,NV90]. Adequacy means compatibility of the behavioral equivalence of the process algebra with the semantics of the logic. That is, equivalent processes satisfy the same formulas and consequently cannot be distinguished in the logic. Clearly, adequacy corresponds to a particular case of property preservation; in general, the

[*] This work was partially supported by ESPRIT Basic Research Action "SPEC"

property preserving relations need not to be equivalences.

This paper is based on ideas originally presented in [Sif82a,Sif83]. We propose a notion of preservation of a property f by an arbitrary function φ from the powerset of the states of a program S_1 to the powerset of the states of a program S_2. The function φ is said to preserve f if for any state of S_1 which satisfies f, all the states of S_2 in its image satisfy f too. If the converse is also true, then we say that φ strongly preserves f. Some general results are provided allowing to prove that a given function preserves (or strongly preserves) the meaning of the formulas of a language by induction on its structure.

These results are applied to show both preservation and strong preservation of sublanguages of the branching time μ-calculus for functions that represent some general form of homomorphism between the transition structures of the considered programs. Homomorphisms or simulation relations between transition structures are at the base of the definitions of most of the relations used to compare behaviors [Bra78,KM79,Mil71]. Their use allows the application of algebraic techniques and reduces the study of a relation on programs to the study of relations on elements of their structure.

The simulation relations used are parameterized by a pair of functions (φ, ψ) which defines a Galois connection between powersets of the state sets of the programs considered. The results presented concern preservation of properties expressed in fragments of the μ-calculus when programs are related via $<\varphi, \psi>$-simulations. The functions φ and ψ determine precisely state correspondences that preserve the satisfaction of formulas.

These results generalize some results in [CGL92] where this problem is studied in the particular case where the restriction of the property preserving function φ on the states is an abstraction. The idea of using simulations parameterized by Galois connections is quite natural as connections have been proven to be very useful for studying correspondences between partially ordered structures [Ore44], in our case the lattices of state properties of two programs. In the domain of verification and abstract interpretation of programs they have been extensively used by Patrick and Rhadia Cousot (see for example [CC79,CC90]). Our results specialize their approach as far as $<\varphi, \psi>$-simulations induce a particular case of abstract interpretations. These interpretations under some additional conditions preserve the validity of properties on abstractions.

The paper is organized as follows. Section 2 presents the notion of property preservation and general results allowing to prove that a function preserves the validity of formulas of a given language. In section 3, the definition and properties of $<\varphi, \psi>$-simulations are given. Section 4 gives results about the preservation of sublanguages of the μ-calculus. Section 5 shows how the results can be applied to obtain abstractions or implementations of a given program.

2 General results on property preservation

A program is considered to be a transition system defined as follows:

Definition 1. A *transition system* is a tuple $S = (Q, R)$, where Q is a set of states and R is a transition relation on Q ($R \subseteq Q \times Q$).

We adopt the following conventions and notations:

- We identify a unary predicate on Q with its characteristic set as the lattice of unary predicates is isomorphic to 2^Q. Thus, for a unary predicate P and a state $q \in Q$, the notations $P(q) = true$, $P(q)$ and $q \in P$ are equivalent.

- Given two sets Q_1 and Q_2, we represent by $[Q_1 \rightarrow Q_2]$ (resp. $[Q_1 \overset{m}{\rightarrow} Q_2]$) the set of (resp. monotonic) functions from Q_1 to Q_2.
- We denote by Id the *identity* function on 2^Q. For a set $\Pi \subseteq Q$, we denote by Id_Π the restriction of Id to 2^Π.

We suppose that program properties are expressed by formulas of a logical language $\mathcal{F}(\mathcal{P})$ where $\mathcal{P} = \{P_1, P_2, ...\}$ is a set of propositional variables. For a given system $S = (Q, R)$ and an *interpretation function* $\mathcal{I} \in [\mathcal{P} \rightarrow 2^Q]$, the semantics of $\mathcal{F}(\mathcal{P})$ is given by means of a function $|\ |_{S,\mathcal{I}} \in [\mathcal{F}(\mathcal{P}) \rightarrow 2^Q]$. This function is such that $\forall P \in \mathcal{P}$, $|P|_{S,\mathcal{I}} = \mathcal{I}(P)$. Furthermore, it associates with a formula its characteristic set i.e., the set of states satisfying it.
To simplify notations, either one or both of the subscripts S and $\mathcal{I}$ in $|f|_{S,\mathcal{I}}$ will be omitted whenever their values can be determined by the context.

Definition 2. Let $f \in \mathcal{F}(\mathcal{P})$ be a formula, $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems, Π be a subset of Q_1 $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$ an interpretation function and $\varphi \in [2^{Q_1} \rightarrow 2^{Q_2}]$. We say that φ *preserves* (resp. *strongly preserves*) f for $\mathcal{I}$ on Π if and only if for any $q \in \Pi$,

$q \in |f|_{S_1,\mathcal{I}}$ implies (resp. iff) $\varphi(\{q\}) \subseteq |f|_{S_2,\varphi\circ\mathcal{I}}$.

If $\Pi = Q_1$, we omit to precize that the preservation is on Π.

Definition 3. Let $f \in \mathcal{F}(\mathcal{P})$ be a formula, $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems, $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$ be an interpretation function and $\varphi \in [2^{Q_1} \rightarrow 2^{Q_2}]$.

We say that φ *semi-commutes* (resp. *commutes*) with f for $\mathcal{I}$ if and only if $\varphi(|f|_{S_1,\mathcal{I}}) \subseteq |f|_{S_2,\varphi\circ\mathcal{I}}$ (resp. $\varphi(|f|_{S_1,\mathcal{I}}) = |f|_{S_2,\varphi\circ\mathcal{I}}$).

In these definitions, the function φ establishes a correspondence between properties of S_1 and properties of S_2. Preservation means that the function φ is compatible with the satisfaction relation. The notion of semi-commutativity is used in the sequel to prove preservation. The following lemma relates preservation and commutativity.

Lemma 4. *Let f be a formula of $\mathcal{F}(\mathcal{P})$, $\varphi \in [2^{Q_1} \overset{m}{\rightarrow} 2^{Q_2}]$ a* monotonic *function and $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$.*

1. *If φ semi-commutes with f for $\mathcal{I}$ then φ preserves f for $\mathcal{I}$.*
2. *If φ commutes with f for $\mathcal{I}$, φ is one-to-one and φ^{-1} is monotonic then φ strongly preserves f for $\mathcal{I}$.*

Proof. The first point is immediate, from $q \in |f|_\mathcal{I}$, we obtain by monotonicity of φ and its semi-commutativity with f, $\varphi(\{q\}) \subseteq \varphi(|f|_\mathcal{I}) \subseteq |f|_{\varphi\circ\mathcal{I}}$.
For the second point it remains to prove that for any $q \in Q_1$, $\varphi(\{q\}) \subseteq |f|_{\varphi\circ\mathcal{I}}$ implies $q \in |f|_\mathcal{I}$. $\varphi(\{q\}) \subseteq |f|_{\varphi\circ\mathcal{I}} = \varphi(|f|_\mathcal{I})$ by commutativity of φ with f. From $\varphi(\{q\}) \subseteq \varphi(|f|_\mathcal{I})$ we deduce that $q \in |f|_\mathcal{I}$ as φ^{-1} is monotonic and φ is one-to-one.

Theorem 5. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. For any set $\Pi \subseteq Q_1$ and for any monotonic functions $\varphi \in [2^{Q_1} \overset{m}{\rightarrow} 2^{Q_2}]$ and $\psi \in [2^{Q_2} \overset{m}{\rightarrow} 2^{Q_1}]$ such that $\psi \circ \varphi \circ \psi = \psi$ and $Id_\Pi \subseteq \psi \circ \varphi$, if φ semi-commutes with f for $\mathcal{I} \in [\mathcal{P} \rightarrow Im(\psi)]$ and ψ semi-commutes with f for $\varphi\circ\mathcal{I}$ then φ strongly preserves f for $\mathcal{I}$ on Π.*

Proof. The preservation is obtained by lemma 4 since φ semi-commutes with f for $\mathcal{I}$.
In order to show strong preservation, suppose that, for $q \in \Pi$, $\varphi(\{q\}) \subseteq |f|_{\varphi\circ\mathcal{I}}$. We have,

$\psi\circ\varphi(\{q\}) \subseteq \psi(|f|_{\varphi\circ\mathcal{I}})$ (monotonicity of ψ),
$q \in \psi(|f|_{\varphi\circ\mathcal{I}})$ $(Id_\Pi \subseteq \psi\circ\varphi)$,
$q \in |f|_{\psi\circ\varphi\circ\mathcal{I}}$ (semi-commutativity of ψ for $\varphi\circ\mathcal{I}$).

Since $\mathcal{I} \in [\mathcal{P}\rightarrow Im(\psi)]$, there exists an interpretation function $\mathcal{I}' \in [\mathcal{P}\rightarrow 2^{Q_2}]$ such that $\mathcal{I} = \psi\circ\mathcal{I}'$. Thus $\psi\circ\varphi\circ\mathcal{I} = \psi\circ\varphi\circ\psi\circ\mathcal{I}' = \psi\circ\mathcal{I}' = \mathcal{I}$ which implies $q \in |f|_{\mathcal{I}}$.

3 Simulations based on connections ($<\varphi,\psi>$-simulations)

The notion of $<\varphi,\psi>$-simulation plays a central role for the preservation of properties of two transition systems $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$. To introduce it, the following definitions and well-known results are needed.

Definition 6. Given a relation R from a set Q_1 to a set Q_2 $(R \subseteq Q_1\times Q_2)$ we define two predicate transformers $pre[R] \in [2^{Q_2}\rightarrow 2^{Q_1}]$ and $post[R] \in [2^{Q_1}\rightarrow 2^{Q_2}]$:

$$pre[R] \overset{def}{=} \lambda X.\ \{q_1 \in Q_1\ :\ \exists q_2 \in X \wedge\ q_1\ R\ q_2\}$$
$$post[R] \overset{def}{=} \lambda X.\ \{q_2 \in Q_2\ :\ \exists q_1 \in X \wedge\ q_1\ R\ q_2\}$$

In the sequel, we denote by $\widetilde{\varphi}$ the dual of a function $\varphi \in [Q_1\rightarrow Q_2]$ that is
$\widetilde{\varphi} \overset{def}{=} \lambda X.\neg\varphi(\neg X)$.
Notice that for ${Q_2}' \subseteq Q_2$, $pre[R]({Q_2}')$ represents the set of "predecessors" of the states of ${Q_2}'$ via the relation R and for ${Q_1}' \subseteq Q_1$, $post[R]({Q_1}')$ represents the set of "successors" of the states of ${Q_1}'$ via R. The following proposition states that the operator *pre* is strict and distributive over union.

Proposition 7. *For any relation R from a set Q_1 to a set Q_2 $(R \subseteq Q_1 \times Q_2)$, we have:*

1. $pre[R](\emptyset) = \emptyset$,
2. *For any X_1, X_2 subsets of Q_2,* $pre[R](X_1 \cup X_2) = pre[R](X_1) \cup pre[R](X_2)$,

Definition 8. A connection from 2^{Q_1} and 2^{Q_2} is a pair of monotonic functions (φ,ψ), $\varphi \in [2^{Q_1} \overset{m}{\rightarrow} 2^{Q_2}]$ and $\psi \in [2^{Q_2} \overset{m}{\rightarrow} 2^{Q_1}]$, such that $Id_{Q_1} \subseteq \psi\circ\varphi$ and $\varphi\circ\psi \subseteq Id_{Q_2}$.

Proposition 9. *[San77] Let φ and ψ be two monotonic functions $\varphi \in [2^{Q_1} \overset{m}{\rightarrow} 2^{Q_2}]$, $\psi \in [2^{Q_2} \overset{m}{\rightarrow} 2^{Q_1}]$. If (φ,ψ) is a connection between 2^{Q_1} and 2^{Q_2} then φ is distributive w.r.t. union and ψ is distributive w.r.t. intersection.*

Proposition 10. *Let ρ be a relation from a set Q_1 to a set Q_2 ($R \subseteq Q_1\times Q_2$), the pairs $(post[\rho], \widetilde{pre}[\rho])$ and $(pre[\rho], \widetilde{post}[\rho])$ are connections.*

Now, we define a notion of simulation and bisimulation parameterized by connections relating the lattices of properties of two transition systems.

Definition 11. Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$, two transition systems and (φ, ψ) a connection from 2^{Q_1} to 2^{Q_2}.

- $S_1 < \varphi, \psi >$-simulates S_2 if $\varphi \circ pre[R_1] \circ \psi \subseteq pre[R_2]$
- $S_1 < \varphi, \psi >$-bisimulates S_2 if $S_1 < \varphi, \psi >$-simulates S_2 and $S_2 < \psi, \widetilde{\varphi} >$-simulates S_1.

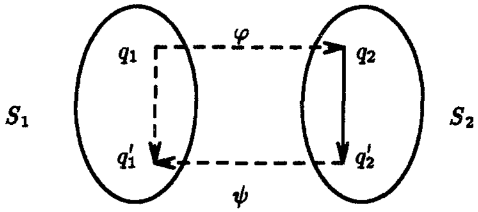

Fig. 1. $< \varphi, \psi >$-simulation

Notice that the definition above implies that for any $< \varphi, \psi >$-simulation, we have necessarily $\varphi(\emptyset) = \emptyset$ since $pre[R_2](\emptyset) = \emptyset$ and φ is monotonic. In that case, as φ is distributive for $\cup$, it can be shown that there exists a relation $\rho \subseteq Q_1 \times Q_2$ such that $\varphi = post[\rho]$ [Sif82b]. Indeed, we have for any $q_1 \in Q_1$ and $q_2 \in Q_2$, $q_1 \rho q_2$ if and only if $q_2 \in \varphi(q_1)$.

We show below that the notion of $< \varphi, \psi >$-simulation is equivalent to the standard notion of simulation [Mil71]: S_1 simulates S_2 if there exists some relation $\rho \subseteq Q_1 \times Q_2$ such that $R_1{}^{-1}\rho \subseteq \rho R_2{}^{-1}$. That is, if $q_1 \ \rho \ q_2$ then for any q_1' such that $q_1 R_1 q_1'$ there exists q_2' such that $q_2 R_2 q_2'$ and $q_1' \rho q_2'$.

Suppose that for some relation $\rho \subseteq Q_1 \times Q_2$, $S_1 < post[\rho], \widetilde{pre}[\rho] >$-simulates S_2, i.e.

$$post[\rho] \circ pre[R_1] \circ \widetilde{pre}[\rho] \subseteq pre[R_2].$$

Then, as $post[\rho]$ is monotonic and $Id_{Q_1} \subseteq \widetilde{pre}[\rho] \circ post[\rho]$, we obtain,

$post[\rho] \circ pre[R_1] \circ \widetilde{pre}[\rho] \circ post[\rho] \subseteq pre[R_2] \circ post[\rho]$ which implies
$post[\rho] \circ pre[R_1] \subseteq pre[R_2] \circ post[\rho]$ which is equivalent to $R_1{}^{-1}\rho \subseteq \rho R_2{}^{-1}$.

Thus, if $S_1 < post[\rho], \widetilde{pre}[\rho] >$-simulates S_2, then S_1 simulates S_2 in the sense of [Mil71]. It can be shown in a similar way that the converse also holds. A direct consequence of this is that $S_1 < post[\rho], \widetilde{pre}[\rho] >$-bisimulates S_2 for some ρ iff S_1 bisimulates S_2.

Notice also that $< \varphi, \psi >$-simulations define abstract interpretations in the sense of [CC79,CC90]. So, they should allow to establish relationships between on one hand, theories based on behavioral equivalences expressed in term of simulations and on the other hand, existing powerful results on program analysis using abstract interpretation.

The following proposition gives a useful dual condition for the definition of $< \varphi, \psi >$-simulation:

Proposition 12. *For $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ two transition systems and (φ, ψ) a connection from 2^{Q_1} to 2^{Q_2},*
$\varphi \circ pre[R_1] \circ \psi \subseteq pre[R_2]$ *iff* $pre[R_1] \subseteq \psi \circ pre[R_2] \circ \varphi$

Proof. From $\varphi \circ pre[R_1] \circ \psi \subseteq pre[R_2]$ and monotonicity of φ and ψ one can deduce that $\psi \circ \varphi \circ pre[R_1] \circ \psi \circ \varphi \subseteq \psi \circ pre[R_2] \circ \varphi$. As $Id_{Q_1} \subseteq \psi \circ \varphi$ one obtains $pre[R_1] \subseteq \psi \circ pre[R_2] \circ \varphi$. It is easy to show that the latter implies $\varphi \circ pre[R_1] \circ \psi \subseteq pre[R_2]$. Thus, the two conditions are equivalent.

4 Preservation of the μ-calculus

We consider the problem of preservation of the properties expressible in the branching-time propositional μ-calculus L_μ [Koz83]. We recall that this logic subsumes in expressiveness all the commonly used specification logics as the branching-time temporal logics CTL [CES83] and CTL^* [EH83] and also the linear-time temporal logics as PTL [Pnu77] and ETL [Wol83]. We define two fragments of the μ-calculus called $\Box\mu$-calculus ($\Box L_\mu$) and $\Diamond\mu$-calculus ($\Diamond L_\mu$) and we show that when a system S_1 $<\varphi, \psi>$-simulates another system S_2, the function φ (resp. $\widetilde{\psi}$) preserves $\Diamond L_\mu$ (resp. $\Box L_\mu$). We obtain strong preservation of these fragments in case of a simulation equivalence, i.e., existence of simulations in both directions. Furthermore, we show that in the case of $<\varphi, \psi>$-bisimulation, the two functions mentioned above preserve the whole L_μ and that under some conditions they strongly preserve it.

4.1 Fragments of the propositional μ-calculus

We recall the syntax and the semantics of the propositional μ-calculus L_μ [Koz83]. Let $\mathcal{P}$ be a set of atomic propositions, $\mathcal{X}$ a set of variables. The set of the formulas of the μ-calculus is defined by the following grammar:

> $f ::= \top \mid P \in \mathcal{P} \mid X \in \mathcal{X} \mid \Diamond f \mid f \vee f \mid \neg f \mid \mu X.f$ where f is syntactically monotonic on X, i.e. any occurrence of X in f is under an even number of negations.

The notion of free occurrences of variables in a formula is defined as in the first-order predicate calculus by considering the operator μ as a quantifier. As usually, a formula is *closed* if there are no variables occurring free in it.

The semantics of the formulas is defined for a given transition system $S = (Q, R)$ and an interpretation function for the atomic propositions $\mathcal{I} \in [\mathcal{P} \rightarrow 2^Q]$. A formula with n free variables is interpreted as a function of $[(2^Q)^n \rightarrow 2^Q]$. In particular, a closed formula is interpreted as a set of states. The interpretation function is inductively defined as follows, for a *valuation* $V = (V_1, ..., V_n) \in (2^Q)^n$ of the variables occurring free.

$$\begin{aligned}
|\top|_{\mathcal{I}} &= Q, \\
|P|_{\mathcal{I}} &= \mathcal{I}(P), \\
|X_j|_{\mathcal{I}}(V) &= V_j, \\
|f_1 \vee f_2|_{\mathcal{I}}(V) &= |f_1|_{\mathcal{I}}(V) \cup |f_2|_{\mathcal{I}}(V), \\
|\neg f|_{\mathcal{I}}(V) &= Q - |f|_{\mathcal{I}}(V), \\
|\Diamond f|_{\mathcal{I}}(V) &= \{q \in Q \ : \ \exists q' \in Q, qRq' \text{ and } q' \in |f|_{\mathcal{I}}(V)\} = pre[R](|f|_{\mathcal{I}}(V)), \\
|\mu X.f|_{\mathcal{I}}(V) &= \bigcap\{Q' \subseteq Q \ : \ |f|_{\mathcal{I}}(Q', V) \subseteq Q'\}.
\end{aligned}$$

We extend the language of L_μ by adding the formulas $\bot$, $f \wedge g$, $f \Rightarrow g$, $\nu X.f(X)$, $\Box f$ which are respectively abbreviations for $\neg\top$, $\neg(\neg f \vee \neg g)$, $\neg f \vee g$, $\neg\mu X.\neg f(\neg X)$, $\neg\Diamond\neg f$.

A formula of this extended language is in *positive normal form* if and only if all the negations occurring in it are applied on atomic propositions. It can be shown that any formula of L_μ has an equivalent formula which is in positive normal form.

We define two fragments of L_μ called $\Box L_\mu$ and $\Diamond L_\mu$. Their sets of formulas are given respectively by the two following grammars.

$$g ::= \top \mid \bot \mid P \mid \neg P \mid X \mid \Box g \mid g \vee g \mid g \wedge g \mid \mu X.g \mid \nu X.g$$

$$h ::= \top \mid \bot \mid P \mid \neg P \mid X \mid \Diamond h \mid h \vee h \mid h \wedge h \mid \mu X.h \mid \nu X.h$$

Notice that properties expressible in the $\Box L_\mu$ involve only universal quantification on computation sequences (due to the use of the $\Box$ operator) whereas those expressible in $\Diamond L_\mu$ involve only existential quantification.

We consider the *positive* fragments $\Box L_\mu^+$ and $\Diamond L_\mu^+$ obtained from the above languages by forbidding the use of the negation. We consider also the logic L_μ^+ corresponding to the subset of L_μ formulas in positive normal form without negations. We can translate any formula of L_μ which is in positive normal form into an equivalent formula in L_μ^+ by replacing negations of atomic propositions, i.e, formulas in the form $\neg P$, by new atomic propositions. Thus, since any formula of L_μ has an equivalent formula in positive normal form, we can express in L_μ^+ any property expressible in L_μ, modulo this encoding of the formulas $\neg P$. Obviously, the same translation can be done from $*L_\mu$ to $*L_\mu^+$ for $* \in \{\Box, \Diamond\}$.

We can express in $\Box L_\mu$ branching-time properties as for instance, the *safety* properties w.r.t. the simulation preorder [BFG*91]. The class of these properties corresponds to the fragment of $\Box L_\mu$ without the least fixpoint operator μ.

Furthermore, it can be shown that any ω-regular linear-time property, i.e., expressible by a nondeterministic Büchi automaton [Buc62], can be expressed in $\Box L_\mu$ [Bou89]. For example, the *safety* property [Lam77,LPZ85,MP90] *"always P"* can be expressed by the formula $\nu X.(P \wedge \Box X)$. Moreover, the *guarantee* property (according to [MP90]) *"eventually P in any infinite computation sequence"* can be expressed by the formula $\mu X.(P \vee \Box X)$. Properties in the other classes in the hierarchy given in [MP90] are obtained by using alternations of the μ and the ν operators. The formulas of $\Diamond L_\mu$ are equivalent to negations of the $\Box L_\mu$ formulas and conversely. However, the formulas of $\Diamond L_\mu^+$ are equivalent to the duals of $\Box L_\mu^+$ formulas and conversely.

4.2 Preservation results

In the sequel we consider only finite branching transition systems, i.e., transition systems where any state has a finite number of successors. This condition guaranties that the formulas can be interpreted as continuous functions on sets of states.

When we consider any interpretation function of the atomic propositions the preservation and strong preservation results concerns the positive fragments of the μ-calculus. To deal with the fragments where negations of atomic propositions are allowed, we need the following *consistency* condition.

Definition 13. Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems, $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$ be an interpretation and a function $\varphi \in [2^{Q_1} \rightarrow 2^{Q_2}]$. We say that φ is *consistent* with $\mathcal{I}$ if and only if $\forall P \in \mathcal{P}.\ \varphi(|\neg P|_{S_1,\mathcal{I}}) \cap \varphi(|P|_{S_1,\mathcal{I}}) = \emptyset$.

We give hereafter the theorems concerning the preservation and strong preservation of $\Box L_\mu$, $\Diamond L_\mu$ and L_μ in presence of $<\varphi, \psi>$-simulations or bisimulations. We consider in the sequel that the functions $\varphi \in [2^{Q_1} \rightarrow 2^{Q_2}]$ and $\psi \in [2^{Q_2} \rightarrow 2^{Q_1}]$ forming the $<\varphi, \psi>$ simulations and bisimulations satisfy the following conditions: $\varphi(\emptyset) = \emptyset$, $\varphi(Q_1) = Q_2$, $\widetilde{\psi}(\emptyset) = \emptyset$ and $\widetilde{\psi}(Q_2) = Q_1$.

We start with two theorems concerning the preservation of $\Box L_\mu$ and $\Diamond L_\mu$ respectively by $\widetilde{\psi}$ and φ when $<\varphi, \psi>$ is a simulation. Their proofs are given in the appendix.

Theorem 14. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. If S_1 $<\varphi, \psi>$-simulates S_2 then $\widetilde{\psi}$ preserves the formulas of $\Box L_\mu^+$ for any interpretation function $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_2}]$. Furthermore, if $\widetilde{\psi}$ is consistent with $\mathcal{I}$ then $\widetilde{\psi}$ preserves $\Box L_\mu$ for $\mathcal{I}$.*

Theorem 15. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. If S_1 $<\varphi, \psi>$-simulates S_2 then φ preserves the formulas of $\Diamond L_\mu^+$ for any interpretation function $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$. Furthermore, if φ is consistent with $\mathcal{I}$, then φ preserves $\Diamond L_\mu$ for $\mathcal{I}$.*

Now, we give a theorem about strong preservation of $\Box L_\mu$ and $\Diamond L_\mu$ in case of a simulation equivalence.

Theorem 16. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. If S_1 $<\varphi, \psi>$-simulates S_2 and S_2 $<\varphi', \psi'>$-simulates S_1 then*

1. *For any set $\Pi \subseteq Q_1$, if $\varphi' \circ \varphi \circ \varphi' = \varphi'$ and $Id_\Pi \subseteq \varphi' \circ \varphi$, then φ strongly preserves $\Diamond L_\mu^+$ on Π for any interpretation $\mathcal{I} \in [\mathcal{P} \rightarrow Im(\varphi')]$. Furthermore, if φ (resp. φ') is consistent with $\mathcal{I}$ (resp. $\varphi \circ \mathcal{I}$) then φ strongly preserves $\Diamond L_\mu$ for $\mathcal{I}$ on Π.*
2. *For any set $\Pi \subseteq Q_2$, if $\widetilde{\psi}' \circ \widetilde{\psi} \circ \widetilde{\psi}' = \widetilde{\psi}'$ and $Id_\Pi \subseteq \widetilde{\psi}' \circ \widetilde{\psi}$, then $\widetilde{\psi}$ strongly preserves $\Box L_\mu^+$ on Π for any interpretation $\mathcal{I} \in [\mathcal{P} \rightarrow Im(\widetilde{\psi}')]$. Furthermore, if $\widetilde{\psi}$ (resp. $\widetilde{\psi}'$) is consistent with $\mathcal{I}$ (resp. $\widetilde{\psi} \circ \mathcal{I}$) then $\widetilde{\psi}$ strongly preserves $\Box L_\mu$ for $\mathcal{I}$ on Π.*

Proof. Direct application of the theorem 5 using the theorems 14 and 15.

We consider now the case of bisimulation connections. The following theorems concerns the preservation and the strong preservation of the whole μ-calculus in presence of such connections.

Theorem 17. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. If S_1 $<\varphi, \psi>$-bisimulates S_2 then φ (resp. $\widetilde{\psi}$) preserves L_μ^+ for any interpretation function $\mathcal{I}_1 \in [\mathcal{P} \rightarrow 2^{Q_1}]$ (resp. $\mathcal{I}_2 \in [\mathcal{P} \rightarrow 2^{Q_2}]$).*

Furthermore, if φ (resp. $\widetilde{\psi}$) is consistent with $\mathcal{I}_1$ (resp. $\mathcal{I}_2$) then φ (resp. $\widetilde{\psi}$) preserves L_μ for $\mathcal{I}_1$ (resp. $\mathcal{I}_2$).

Proof. The proof is a combination of the proofs of the theorems 14, 15 and the definition of the bisimulation (see definition 8).

Theorem 18. *Let $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$ be two transition systems. If S_1 $<\varphi, \psi>$-bisimulates S_2 then*

1. *If $Id_{Q_1} \subseteq \widetilde{\psi} \circ \varphi$ and $\widetilde{\psi} \circ \varphi \circ \widetilde{\psi} = \widetilde{\psi}$ then φ strongly preserves L_μ^+ for any interpretation $\mathcal{I}_1 \in [\mathcal{P} \rightarrow Im(\widetilde{\psi})]$. Furthermore, if φ and $\widetilde{\psi}$ are respectively consistent with $\mathcal{I}_1$ and $\varphi \circ \mathcal{I}_1$, then φ strongly preserves L_μ for $\mathcal{I}_1$.*
2. *If $Id_{Q_1} \subseteq \widetilde{\psi} \circ \varphi$ and $Id_{Q_2} \subseteq \varphi \circ \widetilde{\psi}$ then for any interpretation $\mathcal{I}_1 \in [\mathcal{P} \rightarrow Im(\widetilde{\psi})]$, φ (resp. $\widetilde{\psi}$) strongly preserve L_μ^+ for $\mathcal{I}_1$ (resp. $\varphi \circ \mathcal{I}_1$).*

   Furthermore, if φ (resp. $\widetilde{\psi}$) is consistent with $\mathcal{I}_1$ (resp. $\varphi \circ \mathcal{I}_1$) then φ (resp. $\widetilde{\psi}$) strongly preserves L_μ for $\mathcal{I}_1$ (resp. $\varphi \circ \mathcal{I}_1$).

Proof. The first point is obtained by direct application of theorem 5 and using the proof of theorem 17 showing actually semi-commutativity which is stronger than preservation. The second point is also obtained by direct application of theorem 5, given that $Id_{Q_1} \subseteq \widetilde{\psi} \circ \varphi$ and $Id_{Q_2} \subseteq \varphi \circ \widetilde{\psi}$ implies that $\widetilde{\psi} \circ \varphi \circ \widetilde{\psi} = \widetilde{\psi}$ and $\varphi \circ \widetilde{\psi} \circ \varphi = \varphi$.

5 Applications

In this section we consider that the functions φ and ψ are defined in terms of a relation $\rho \subseteq Q_1 \times Q_2$ relating states of two transition systems $S_1 = (Q_1, R_1)$ and $S_2 = (Q_2, R_2)$.

As the pair $(post[\rho], \widetilde{pre}[\rho])$ is a connection from 2^{Q_1} to 2^{Q_2}, it is natural to consider $<post[\rho], \widetilde{pre}[\rho]>$-simulations from S_1 to S_2. In this context the results presented can be applied to tackle two problems: the implementation problem and the abstraction problem.

5.1 Implementation

Problem: given a transition system $S_2 = (Q_2, R_2)$, a set of states Q_1 and a relation $\rho \subseteq Q_1 \times Q_2$, find a transition system $S_1 = (Q_1, R_1)$ ***implementing*** S_2 via ρ that is, a system S_1 such that S_1 $<post[\rho], \widetilde{pre}[\rho]>$-simulates S_2.

From the relation $pre[R_1] \subseteq \widetilde{pre}[\rho] \circ pre[R_2] \circ post[\rho]$, one gets that in general there may be many solutions (relations R_1). However, there exists a largest relation. It is obtained by computing the largest function F such that $F \subseteq \widetilde{pre}[\rho] \circ pre[R_2] \circ post[\rho]$ which is distributive with respect to union and strict ($F(\emptyset) = \emptyset$). As it is shown in [Sif82b], for any such function there exists a unique relation $pre[R_1]$ such that $pre[R_1] = F$. This function is defined by taking $F(\emptyset) = \emptyset$, $F(P) = \bigvee_{q \in P} H(q)$, where $H = \widetilde{pre}[\rho] \circ pre[R_2] \circ post[\rho]$.

5.2 Abstraction

Problem: given a transition system $S_1 = (Q_1, R_1)$, a set of state Q_2 and a relation $\rho \subseteq Q_1 \times Q_2$, find a transition system $S_2 = (Q_2, R_2)$ which is an ***abstraction*** of S_1 via ρ. That is, a transition system S_2 such that S_1 $<post[\rho], \widetilde{pre}[\rho]>$-simulates S_2.

Obviously, the relation $post[\rho] \circ pre[R_1] \circ \widetilde{pre}[\rho] \subseteq pre[R_2]$ may have several solutions (i.e., relations R_2). We are interested in solutions which are sufficiently faithful to S_1. In general, a least solution does not exists ; however, if ρ is total on Q_2, we have $\widetilde{pre}[\rho] \subseteq pre[\rho]$ and taking $post[\rho] \circ pre[R_1] \circ pre[\rho] = pre[R_2]$ defines an acceptable abstraction of S_1. The least abstraction exists if ρ is taken to be a total function. Then, $\widetilde{pre}[\rho] = pre[\rho]$ and $post[\rho] \circ pre[R_1] \circ pre[\rho] = pre[R_2]$.

We compute the abstraction of a program w.r.t. to a relation ρ using symbolic representations for both the program and its abstraction.

Consider a program S_1 defined on a tuple of variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ ranging on a domain $\mathbf{D} = D_1 \times D_2 \times \cdots \times D_n$, represented by guarded commands:

$$do\ \{C_1 \rightarrow \alpha_1 [] ... C_i \rightarrow \alpha_i [] ... C_k \rightarrow \alpha_k\}\ od$$

where the C_i's are predicates on $\mathbf{X}$ and the α_i's are assignements defining functions of $[\mathbf{D} \rightarrow \mathbf{D}]$.

Symbolically, sets of states of S_1 are represented by predicates on $\mathbf{X}$ and the transitions between these sets are given by the function $pre[R_1]$ which, for any predicate $P(\mathbf{X})$ (representing a set of states), is defined by:

$$pre[R_1](P(\mathbf{X})) = \bigvee_{i=1}^{k} C_i \wedge P[\alpha_i(\mathbf{X})/\mathbf{X}].$$

The abstraction S_2 is a program on a tuple of variables $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$ ranging on an *abstract* domain $\mathbf{D}' = D'_1 \times D'_2 \times \cdots \times D'_m$.

Given a relation ρ between the domains $\mathbf{D}$ ans $\mathbf{D}'$ expressed by a predicate on $\mathbf{X}$ and $\mathbf{Y}$, the symbolic representation of S_2 is defined by :

$$pre[R_2] = \bigvee_{i=1}^{k} post[\rho](C_i \wedge pre[\rho](P[\alpha_i(\mathbf{X})/\mathbf{X}])).$$

In [CGL92] a construction of a program abstraction is described. Given a system S_1 and a function h from the domain of the program variables to an abstract domain, a system S_2 is constructed in such a manner that h induces a *homomorphism* from S_1 to S_2. The notion of homomorphism corresponds in our approach to a $<\varphi, \psi>$-simulation where $\varphi = post[\rho]$ and $\psi = \widetilde{pre}[\rho]$ for ρ a total function and φ and $\tilde{\psi}$ are respectively consistent with the interpretation functions of the atomic propositions $\mathcal{I}$ and $\varphi \circ \mathcal{I}$. In that case, it is shown that the logic ∀CTL* is preserved from S_2 to S_1. This result is generalized by the theorem 14 since $\Box L_\mu$ is more expressive than ∀CTL*.

Furthermore, the notion of *exact homomorphism* considered in this paper corresponds to a $<\varphi, \psi>$-bisimulation where $\varphi = post[\rho]$ and $\psi = \widetilde{pre}[\rho]$ for ρ a total function, φ and $\tilde{\psi}$ are consistent respectively with the interpretation functions $\mathcal{I}$ and $\varphi \circ \mathcal{I}$. If S_1 and S_2 are related by an exact homomorphism, the logic CTL* is strongly preserved. This result is generalized by the theorem 18 since L_μ is more expressive than CTL* (notice that this theorem can be applied as ρ is a total relation and this implies $Id_{Q_1} \subseteq pre[\rho] \circ post[\rho]$ and $Id_{Q_2} \subseteq post[\rho] \circ pre[\rho]$).

5.3 Example

The motion of a mobile on a grid is controlled so as to visit cyclically the points $CDACDA$.... Initially the mobile is within the rectangle defined by the points (A, B, C, D).

The following program describes the mobile's motion where $Ctrl$ is a control variable with domain $\{A, C, D\}$ recording the most recently visited point and X, Y are its discrete coordinates with respective domains $[0, .., X_0]$ and $[0, .., Y_0]$.

```
do
    $(Ctrl = A) \wedge (0 \leq X < X_0)$                  $\rightarrow (X, Y, Ctrl) := (X + 1, Y, Ctrl)$
 [] $(Ctrl = A) \wedge (0 \leq Y < Y_0)$                  $\rightarrow (X, Y, Ctrl) := (X, Y + 1, Ctrl)$
 [] $(Ctrl = A) \wedge (X = X_0) \wedge (Y = Y_0)$        $\rightarrow (X, Y, Ctrl) := (X - 1, Y, C)$
 [] $(Ctrl = C) \wedge (X > 0)$                           $\rightarrow (X, Y, Ctrl) := (X - 1, Y, Ctrl)$
 [] $(Ctrl = C) \wedge (X = 0)$                           $\rightarrow (X, Y, Ctrl) := (X, Y - 1, D)$
 [] $(Ctrl = D) \wedge (Y > 0)$                           $\rightarrow (X, Y, Ctrl) := (X, Y - 1, Ctrl)$
 [] $(Ctrl = D) \wedge (Y = 0)$                           $\rightarrow (X, Y, Ctrl) := (X, Y, A)$
od
```

An abstraction of this system with two three-valued variables h and v of domains respectively $\{h_0, h_1, h_2\}$ and $\{v_0, v_1, v_2\}$, is defined via the relation ρ:

$(X, Y, Ctrl)\ \rho\ (h, v, Ctrl)$ iff

$[(X = 0 \wedge h = h_0) \vee (0 < X < X_0 \wedge h = h_1) \vee (X = X_0 \wedge h = h_2)] \wedge$

$[(Y = 0 \wedge v = v_0) \vee (0 < Y < Y_0 \wedge v = v_1) \vee (Y = Y_0 \wedge v = v_2)]$

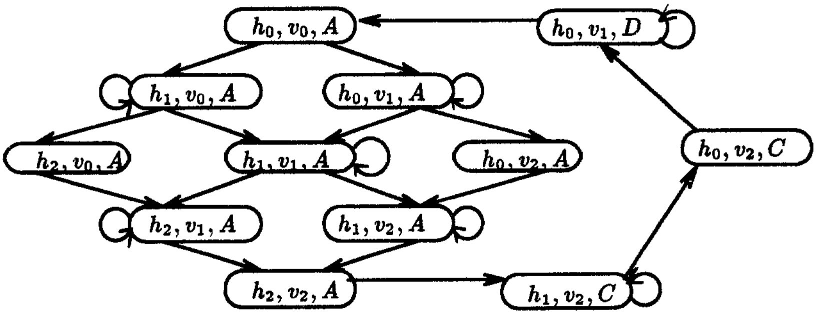

Fig. 2. Abstract mobile system

This abstraction is represented by the finite transition system given in figure 2. Consider the formula,

$$\begin{aligned} f = {} & \textit{not } (Ctrl = A) \textit{ to } (Ctrl = D) \textit{ unless } (Ctrl = C) \wedge \\ & \textit{not } (Ctrl = C) \textit{ to } (Ctrl = A) \textit{ unless } (Ctrl = D) \wedge \\ & \textit{not } (Ctrl = D) \textit{ to } (Ctrl = C) \textit{ unless } (Ctrl = A) \end{aligned}$$

where *not* P_1 *to* P_2 *unless* P_3 is an abbreviation of the $\Box L_\mu$ formula $P_1 \Rightarrow \nu X.(\neg P_2 \wedge (P_3 \vee \Box X))$.

It can be seen that the function $pre[\rho]$ is consistent with the interpretation of the atomic propositions used in f on the abstract mobile system. We verify that f is true on the abstract system and by theorem 14, we deduce that it is true for the initial mobile system.

6 Conclusion

The paper studies property preserving transformations for reactive systems. A key idea is the use of $<\varphi, \psi>$-simulations which are compatible with the standard notion of simulation (structure homomorphism) often used to define implementations. Furthermore, $<\varphi, \psi>$-simulations induce abstract interpretations à la Cousot and this allows to apply an existing powerful theory for program analysis.

The theory is developed on transition systems but it can be trivially extended to labeled transition systems by requiring $<\varphi,\psi>$-simulation of the corresponding labeled relations. Also this theory can be adapted so as to be applied to preorders and equivalences that can be expressed in terms of simulations or bisimulations by adopting some abstraction criterion. For instance, one can define a $<\varphi,\psi>$-observational equivalence by considering as models, labeled transition systems with silent actions and using the well-known fact that observational equivalence is strong bisimulation equivalence on a modified transition relation.

As a continuation of this work, we intend to focus on the applicability of the results for the verification of properties of reactive systems described as the composition of simple programs with guarded commands.

Acknowledgements

We thank Susanne Graf for many helpful discussions and judicious remarks.

References

[AL88] M. Abadi and L. Lamport. *The existence of Refinement Mappings.* SRC 29, Digital Equipment Corporation, Systems Research Center, August 1988.

[BFG*91] A. Bouajjani, J.C. Fernandez, S. Graf, C. Rodriguez, and J. Sifakis. Safety for branching time semantics. In J.L. Albert, B. Monein, and M.R. Artalejo, editors, *18th ICALP*, pages 76–92, LNCS 510, Springer-Verlag, October 1991.

[Bou89] A. Bouajjani. *From Linear-Time Propositional Temporal Logics to a Branching-Time μ-calculus.* RTC 15, LGI-IMAG, Grenoble, 1989.

[Bra78] D. Brand. *Algebraic simulation between parallel programs.* RC 7206 30923, IBM, Yorktown Heights, 1978.

[Buc62] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Intern. Cong. Logic, Method and Philos. Sci.*, Stantford Univ. Press, 1962.

[CC79] P. Cousot and R. Cousot. Systematic design of program analysis framework. In *Proc. 6th ACM Symp. on Principle of Programming Languages*, 1979.

[CC90] P. Cousot and R. Cousot. *Comparing the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation.* Technical Report, LIX, Ecole Polytechnique, May 1990.

[CES83] E. M. Clarke, E. A. Emerson, and E. Sistla. Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications: A Practical Approach. In *10th Symposium on Principles of Programming Languages (POPL 83)*, ACM, 1983. Complete version published in ACM TOPLAS, 8(2):244–263, April 1986.

[CGL92] E.M. Clarke, O. Grumberg, and D.E. Long. Model checking and abstraction. In *Symposium on Principles of Programming Languages (POPL 92)*, ACM, October 1992.

[EH83] E.A. Emerson and J. Y. Halpern. 'sometimes' and 'not never' revisited: on branching versus linear time logic. In *10th. Annual Symp. on Principles of Programming Languages*, 1983.

[GS86a] S. Graf and J. Sifakis. A logic for the specification and proof of regular controllable processes of CCS. *Acta Informatica*, 23, 1986.

[GS86b] S. Graf and J. Sifakis. A modal characterization of observational congruence on finite terms of CCS. *Information and Control*, 68, 1986.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery*, 32:137–161, 1985.

[KM79] T. Kasai and R.E. Miller. *Homomorphisms between models of parallel computation.* RC 7796 33742, IBM, Yorktown Heights, 1979.

[Koz83] D. Kozen. Results on the propositional μ-calculus. In *Theoretical Computer Science*, North-Holland, 1983.

[Kur89] R.P. Kurshan. *Analysis of Discrete Event Coordination. LNCS 430*, Springer-Verlag, May 1989.

[Lam77] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, 1977.

[LPZ85] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Conference on Logics of Programs, LNCS 194*, Springer Verlag, 1985.

[LT88] N.A. Lynch and M.R. Tuttle. *An introduction to Input/Ouput Automata.* MIT/LCS/TM 373, MIT, Cambridge, Massachussetts, November 1988.

[Mil71] R. Milner. An algebraic definition of simulation between programs. In *Proc. Second Int. Joint Conf. on Artificial Intelligence*, pages 481–489, BCS, 1971.

[MP90] Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proc. 9th ACM Symp. on Princ. of Dist. Comp.*, 1990.

[NV90] R. De Nicola and F. Vaandrager. Three logics for branching bisimulation. In *Proc. of Fifth Symp. on Logic in Computer Science*, Computer Society Press, 1990.

[Ore44] O. Ore. Galois connexions. *Trans. Amer. Math. Soc*, 55:493–513, February 1944.

[Pnu77] A. Pnueli. The Temporal Logic of Programs. In *18th Symposium on Foundations of Computer Science (FOCS 77)*, IEEE, 1977. Revised version published in Theoretical Computer Science, 13:45–60, 1981.

[San77] Luis E. Sanchis. Data types as lattices: retractions, projection and projection. In *RAIRO Theorical computer science, vol 11, nomber 4*, pages 339–344, 1977.

[Sif82a] J. Sifakis. *Property preserving homomorphisms and a notion of simulation of transition systems.* RR IMAG 332, IMAG, November 1982.

[Sif82b] J. Sifakis. A unified approach for studying the properties of transition systems. *Theorical Computer Science*, 18, 1982.

[Sif83] J. Sifakis. Property preserving homomorphisms of transition systems. In E. Clarke and D. Kozen, editors, *Workshop on logics of programs*, LNCS 164, Springer-Verlag, 1983.

[Wol83] P. Wolper. Temporal logic can be more expreessive. *Inform. Contr.*, 56, 1983.

Appendix

We give hereafter the proofs of the theorems 14 and 15. These proofs are based on an induction argument using a well-founded ordering on L_μ formulas defined below. We suppose that the formulas are in positive normal form. Consider the binary relation defined for any formulas f and g by : $f \rhd g$ if and only if either

- f is a subformula of g or
- $g = \mu X.h$ (resp. $g = \nu X.h$) and $\exists k \geq 0.\ f = h^k[\bot/X]$ (resp. $f = h^k[\top/X]$).

Now, we consider the order $\preceq$ defined as the transitive closure of the relation $\rhd$.

Proof of Theorem 14 Let us prove the first part of the theorem. Consider an interpretation function $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_2}]$. By lemma 4, it is sufficient to prove that for any formula f in $\Box L_\mu^+$ and for any valuation V, we have $\widetilde{\psi}(|f|_{s_2,\mathcal{I}}(V)) \subseteq |f|_{s_1,\widetilde{\psi}\circ\mathcal{I}}(\widetilde{\psi}(V))$. The proof is by induction on the order $\preceq$ defined above. To simplify the notations, we omit the valuation V whenever it is not relevant in a proof.

- $\widetilde{\psi}(|\bot|_{s_2,\mathcal{I}}) = |\bot|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$ and $\widetilde{\psi}(|\top|_{s_2,\mathcal{I}}) = |\top|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$ as $\widetilde{\psi}(\emptyset) = \emptyset$ and $\widetilde{\psi}(Q_2) = Q_1$.
- $\widetilde{\psi}(|P|_{s_2,\mathcal{I}}) = |P|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$ by definition of the interpretation function.
- $\widetilde{\psi}(|X_j|_{s_2,\mathcal{I}}(V)) = \widetilde{\psi}(V_j) = |X_j|_{s_1,\widetilde{\psi}\circ\mathcal{I}}(\widetilde{\psi}(V))$
- $\widetilde{\psi}(|\Box f|_{s_2,\mathcal{I}}) = \widetilde{\psi} \circ \widetilde{pre}[R_2](|f|_{s_2,\mathcal{I}})$ since $\widetilde{\psi}$ is monotonic and by definition of the interpretation function. The dual of the $<\varphi,\psi>$-simulation condition is $\widetilde{pre}[R_2] \subseteq \widetilde{\varphi} \circ \widetilde{pre}[R_1] \circ \widetilde{\psi}$. We get, $\widetilde{\psi}(|\Box f|_{s_2,\mathcal{I}}) \subseteq \widetilde{\psi} \circ \widetilde{\varphi} \circ \widetilde{pre}[R_1] \circ \widetilde{\psi}(|f|_{s_2,\mathcal{I}})$. As $\widetilde{\psi} \circ \widetilde{\varphi} \subseteq Id_{Q_1}$, we obtain $\widetilde{\psi}(|\Box f|_{s_2,\mathcal{I}}) \subseteq \widetilde{pre}[R_1] \circ \widetilde{\psi}(|f|_{s_2,\mathcal{I}})$. By induction hypothesis, we have $\widetilde{\psi}(|f|_{s_2,\mathcal{I}}) \subseteq |f|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$. Thus, we have
  $\widetilde{\psi}(|\Box f|_{s_2,\mathcal{I}}) \subseteq \widetilde{pre}[R_1](|f|_{s_1,\widetilde{\psi}\circ\mathcal{I}})$ equivalent to $\widetilde{\psi}(|\Box f|_{s_2,\mathcal{I}}) \subseteq |\Box f|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$.
- $\widetilde{\psi}(|f_1 \vee f_2|_{s_2,\mathcal{I}}) = \widetilde{\psi}(|f_1|_{s_2,\mathcal{I}} \cup |f_2|_{s_2,\mathcal{I}})$ by definition of the interpretation function. As $\widetilde{\psi}$ is distributive with respect to $\cup$, we have $\widetilde{\psi}(|f_1 \vee f_2|_{s_2,\mathcal{I}}) = \widetilde{\psi}(|f_1|_{s_2,\mathcal{I}}) \cup \widetilde{\psi}(|f_2|_{s_2,\mathcal{I}})$. By induction hypothesis, we obtain
  $\widetilde{\psi}(|f_1 \vee f_2|_{s_2,\mathcal{I}}) \subseteq |f_1|_{s_1,\widetilde{\psi}\circ\mathcal{I}} \cup |f_2|_{s_1,\widetilde{\psi}\circ\mathcal{I}} = |f_1 \vee f_2|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$.
- $\widetilde{\psi}(|f_1 \wedge f_2|_{s_2,\mathcal{I}}) \subseteq \widetilde{\psi}(|f_1|_{s_2,\mathcal{I}}) \cap \widetilde{\psi}(|f_2|_{s_2,\mathcal{I}})$ and
  $\widetilde{\psi}(|f_1|_{s_2,\mathcal{I}}) \cap \widetilde{\psi}(|f_2|_{s_2,\mathcal{I}}) = |f_1|_{s_1,\widetilde{\psi}\circ\mathcal{I}} \cap |f_2|_{s_1,\widetilde{\psi}\circ\mathcal{I}} = |f_1 \wedge f_2|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$.
- The proof for $\mu X.f$ and $\nu X.f$ is based on the fact that, since the formulas can be interpreted as continuous functions on sets of states we have $|\mu X.f|_{s_2,\mathcal{I}} = \bigcup_{k\geq 0} |f^k|_{s_2,\mathcal{I}}(\emptyset)$ and $|\nu X.f|_{s_2,\mathcal{I}} = \bigcap_{k\geq 0} |f^k|_{s_2,\mathcal{I}}(Q_2)$, $\widetilde{\psi}(\emptyset) = \emptyset$, $\widetilde{\psi}(Q_2) = Q_1$, $\widetilde{\psi}$ is monotonic and that the formulas $f^k[\bot/X]$ and $f^k[\top/X]$ are strictly inferior w.r.t. $\preceq$ to $\mu X.f$ and $\nu X.f$ respectively.

Now, if $\widetilde{\psi}$ is consistent with $\mathcal{I}$, it is straightforward to deduce that $\widetilde{\psi}(|\neg P|_{s_2,\mathcal{I}}) \subseteq |\neg P|_{s_1,\widetilde{\psi}\circ\mathcal{I}}$.

Proof of Theorem 15 Similar to the proof of theorem 14.
The semi-commutativity with the $\Diamond$ operator is proved in the following manner. Let $\mathcal{I} \in [\mathcal{P} \rightarrow 2^{Q_1}]$. By definition of the interpretation function of the formulas, we have $\varphi(|\Diamond f|_{s_1,\mathcal{I}}) = \varphi \circ pre[R_1](|f|_{s_1,\mathcal{I}})$. As $Id_{Q_1} \subseteq \psi \circ \varphi$, we get $\varphi(|\Diamond f|_{s_1,\mathcal{I}}) \subseteq \varphi \circ pre[R_1] \circ \psi \circ \varphi(|f|_{s_1,\mathcal{I}})$. Since $<\varphi,\psi>$ is a simulation, $\varphi \circ pre[R_1] \circ \psi \subseteq pre[R_2]$. Thus, we get $\varphi(|\Diamond f|_{s_1,\mathcal{I}}) \subseteq pre[R_2] \circ \varphi(|f|_{s_1,\mathcal{I}})$. Furthermore, by induction hypothesis, $\varphi(|f|_{s_1,\mathcal{I}}) \subseteq |f|_{s_2,\varphi\circ\mathcal{I}}$. Thus, we have
$\varphi(|\Diamond f|_{s_1,\mathcal{I}}) \subseteq pre[R_2](|f|_{s_2,\varphi\circ\mathcal{I}}) = |\Diamond f|_{s_2,\varphi\circ\mathcal{I}}$.