Strong Preservation as Completeness in Abstract Interpretation

Francesco Ranzato and Francesco Tapparo

Dipartimento di Matematica Pura ed Applicata, Università di Padova
Via Belzoni 7, 35131 Padova, Italy
{franz,tapparo}@math.unipd.it

Abstract. Many algorithms have been proposed to minimally refine abstract transition systems in order to get strong preservation relatively to a given temporal specification language. These algorithms compute a state equivalence, namely they work on abstractions which are partitions of system states. This is restrictive because, in a generic abstract interpretation-based view, state partitions are just one particular type of abstraction, and therefore it could well happen that the refined partition constructed by the algorithm is not the optimal generic abstraction. On the other hand, it has been already noted that the well-known concept of complete abstract interpretation is related to strong preservation of abstract model checking. This paper establishes a precise correspondence between complete abstract interpretation and strongly preserving abstract model checking, by showing that the problem of minimally refining an abstract model checking in order to get strong preservation can be formulated as a complete domain refinement in abstract interpretation, which always admits a fixpoint solution. As a consequence of these results, we show that some well-known behavioural equivalences used in process algebra like simulation and bisimulation can be elegantly characterized in pure abstract interpretation as completeness properties.

1 Introduction

The design of any abstract model checking framework and/or tool comes always together with a preservation result, roughly stating that for a formula φ specified in some temporal language $\mathcal{L}$, the validity of φ on the abstract model implies the validity of φ on the concrete model. On the other hand, *strong preservation* means that any formula of $\mathcal{L}$ is valid in the abstract model if and only if it is valid in the concrete model. Strong preservation is highly desirable since it allows to draw consequences from negative answers on the abstract side.

This paper follows a standard abstract interpretation approach to abstract model checking, as applied for instance in temporal abstract interpretation [9]. The concrete state semantics of a temporal specification language $\mathcal{L}$ is given by a function $[\![\cdot]\!]$ mapping a formula $\varphi \in \mathcal{L}$ to the set of states $s \in \mathit{State}$ satisfying φ, that is $[\![\varphi]\!] = \{s \in \mathit{State} \mid s \models \varphi\}$. This concrete state semantics is approximated by the abstract semantics induced by any abstract interpretation of $\wp(\mathit{State})$,

D.A. Schmidt (Ed.): ESOP 2004, LNCS 2986, pp. 18–32, 2004.
© Springer-Verlag Berlin Heidelberg 2004

namely a Galois connection (or, equivalently, a closure operator). This approach is more general than classical abstract model checking [5,6] where the abstract model is, analogously to the concrete model, a transition system or a Kripke structure. In our framework, this classical approach corresponds to a particular case of abstraction, namely an abstract domain encoding a partition of the system states. In a general abstract interpretation setting, an abstract model checking is associated to *any abstraction* of the powerset of system states, and this obviously enables a finer-grain taxonomy of abstract model checkers. The concept of *complete* abstract interpretation is well known [8,15]: this encodes an ideal situation where the abstract semantics coincides with the abstraction of the concrete semantics. It should be quite clear that completeness of an abstract interpretation with respect to some semantic functions and strong preservation of an abstract model checker with respect to a temporal language are, somehow, related concepts: this was first formalized by Giacobazzi and Quintarelli [13], who put forward a method for systematically refining abstract model checking in order to eliminate Clarke et al.'s [4] spurious counterexamples. The relationship between completeness and spurious counterexamples was further studied in [10], where it is also shown that stability *à la* Paige and Tarjan [20] for a state partition can be formulated through complete abstract interpretations.

We first generalize the notion of classical strong preservation to our abstract interpretation framework. Namely, the classical concept of strong preservation for an abstract model specified as an abstract transition system, viz. a state partition, is here generalized to an abstract model specified by any generic abstract domain. It turns out that any generic abstract model induces a classical partition-based abstract model checking, but this could give rise to a loss of information. Our results rely on the notion of *forward complete* abstract domain. An abstract domain μ, viewed as a closure operator (closure operators are particularly useful here since they allow us to be independent from the representation of abstract objects), is forward complete for a concrete semantic function $f : \mathit{Concrete} \to \mathit{Concrete}$ when $\mu \circ f \circ \mu = f \circ \mu$. Forward completeness is dual to the aforementioned standard completeness, i.e. $\mu \circ f \circ \mu = \mu \circ f$ — called backward completeness — and intuitively states that when the concrete function f is restricted to abstract objects then it coincides with the abstract function $\mu \circ f$, i.e., the best correct approximation of f in the abstract domain μ. Giacobazzi et al. [15] showed how to systematically and constructively derive backward complete abstract domains from non-complete ones by minimal refinements. This can be done for forward completeness as well: Given any domain μ, the most abstract domain which refines μ and is forward complete for f does exist and it can be characterized as a greatest fixpoint. We call such a domain the (forward) *complete shell* of μ for f.

Let us turn to strong preservation. We consider generic inductively defined languages $\mathcal{L}$, generated from atomic propositions and a set of logical operators $op_1, ..., op_k$ whose interpretations are $\mathbf{op_i} : \wp(\mathit{State})^n \to \wp(\mathit{State})$, where n is the arity of the operator. In our framework any abstraction μ of $\wp(\mathit{States})$ induces an abstract semantics $[\![\cdot]\!]^\mu : \mathcal{L} \to \mu$ for the language $\mathcal{L}$. Given any abstraction μ,

we show that the most abstract semantics which refines μ and is strongly preserving for $\mathcal{L}$ is precisely the complete shell of μ for all the language operators $\mathbf{op_1}, ..., \mathbf{op_n}$. This result can be also read as follows. A number of algorithms have been proposed to minimally refine classical abstract models, i.e. state partitions, in order to get strong preservation relatively to some relevant temporal specification languages. Typically, these are coarsest partition refinement algorithms which compute the state equivalence induced by some *behavioural state equivalence*, e.g., bisimulation, stuttering equivalence (or branching bisimulation) or simulation equivalence, since they exploit the fact that this behavioural equivalence coincides with the state equivalence $\equiv_{\mathcal{L}}$ induced by a temporal language $\mathcal{L}$, namely, $s \equiv_{\mathcal{L}} s'$ iff s and s' agree on each formula of $\mathcal{L}$. This is the case of Paige and Tarjan algorithm [20] for strong preservation of CTL* [2] and of Groote and Vaandrager algorithm [16] for strong preservation of CTL*-X. Our results allow us to provide a clean and elegant generalization of these coarsest partition refinement algorithms in our abstract interpretation framework. Due to lack of space we do not consider stuttering equivalence here (we refer to the full version of the paper).

2 Basic Notions

Notation. Let X be any set. When writing a set $S \in \wp(\wp(X))$, we often write the sets in S in a compact form like in $\{1, 12, 123\} \in \wp(\wp(\{1, 2, 3\}))$. $\complement$ denotes the complement operator. $\mathrm{Fun}(X)$ denotes the set of all the functions $f : X^n \to X$, for some $n \geq 0$. When $n = 0$, f is just a specific object of X. We will denote by $\mathrm{Part}(X)$ the set of partitions on X. The sets in a partition are called blocks. If $\equiv \subseteq X \times X$ is an equivalence relation then we will denote by $P_{\equiv} \in \mathrm{Part}(X)$ the corresponding partition of X. Vice versa, if $P \in \mathrm{Part}(X)$ then we will denote by $\equiv_P \subseteq X \times X$ the corresponding equivalence relation on X. $\mathrm{Part}(X)$ is endowed with the following standard partial order $\preccurlyeq$: given $P_1, P_2 \in \mathrm{Part}(X)$, $P_1 \preccurlyeq P_2$, i.e. P_2 is coarser than P_1 (or P_1 refines P_2) iff $\forall B \in P_1.\exists B' \in P_2.B \subseteq B'$.

We consider transition systems (Q, R) where the relation $R \subseteq Q \times Q$ (also denoted by $\xrightarrow{R}$) is total, i.e., for any $s \in Q$ there exists some $t \in Q$ such that sRt. A Kripke structure (Q, R, AP, ℓ) consists of a transition system (Q, R) together with a (typically finite) set AP of atomic propositions and a labelling function $\ell : Q \to \wp(AP)$. A transition relation $R \subseteq Q \times Q$ defines the usual pre/post transformers on $\wp(Q)$: $\mathrm{pre}[R]$, $\widetilde{\mathrm{pre}}[R]$, $\mathrm{post}[R]$, $\widetilde{\mathrm{post}}[R]$.

Abstract interpretation and completeness. In standard Cousot and Cousot's abstract interpretation theory, abstract domains can be equivalently specified either by Galois connections or by (upper) closure operators (uco's) [8]. These two approaches are equivalent, modulo isomorphic representations of domain's objects. The closure operator approach has the advantage of being independent from the representation of domain's objects and is therefore appropriate for reasoning on abstract domains independently from their representation. Given a complete lattice C, it is well known that the set $\mathrm{uco}(C)$ of all uco's

on C, endowed with the pointwise ordering $\sqsubseteq$, gives rise to the complete lattice $\langle \mathrm{uco}(C), \sqsubseteq, \sqcup, \sqcap, \lambda x.\top_C, id\rangle$. Let us recall that each $\mu \in \mathrm{uco}(C)$ is uniquely determined by the set of its fixpoints, which is its image. Moreover, a subset $X \subseteq C$ is the set of fixpoints of a uco on C iff X is meet-closed, i.e. $X = \mathcal{M}(X) \stackrel{\text{def}}{=} \{\wedge Y \mid Y \subseteq X\}$ (where $\top_C = \wedge_C \varnothing \in \mathcal{M}(X)$). $\mathcal{M}(X)$ is called the Moore-closure of X. Also, $\mu \sqsubseteq \rho$ iff $\rho(C) \subseteq \mu(C)$; in this case, we say that μ is a refinement of ρ. Often, we will identify closures with their sets of fixpoints since this does not give rise to ambiguity. In view of the equivalence above, throughout the paper, $\langle \mathrm{uco}(C), \sqsubseteq\rangle$ will play the role of the lattice of abstract interpretations of C [7,8], i.e. the complete lattice of all the abstract domains of the concrete domain C. The ordering on $\mathrm{uco}(C)$ corresponds to the standard order used to compare abstract domains with regard to their precision: A_1 is more precise than A_2 (or A_2 is more abstract than A_1) iff $A_1 \sqsubseteq A_2$ in $\mathrm{uco}(C)$.

Let $f : C \to C$ be a concrete semantic function and let $f^\sharp : A \to A$ be a corresponding abstract function, where $A = \mu(C)$ for some closure $\mu \in \mathrm{uco}(C)$. Then, $\langle A, f^\sharp\rangle$ is a sound abstract interpretation when $\mu \circ f \sqsubseteq f^\sharp \circ \mu$. The abstract function $\mu \circ f : A \to A$ is called the best correct approximation of f in A. Completeness in abstract interpretation corresponds to require that, in addition to soundness, no loss of precision is introduced by the approximated function $f^\sharp \circ \mu$ on a concrete object $c \in C$ with respect to approximating by μ the concrete computation $f(c)$, namely the equation $\mu \circ f = f^\sharp \circ \mu$ holds. The following dual form of completeness may be considered. The soundness equation $\mu \circ f \sqsubseteq f^\sharp \circ \mu$ is also equivalent to the equation $f \circ \mu \sqsubseteq f^\sharp \circ \mu$. Then, *forward completeness* for $f^\sharp$ corresponds to the equation $f \circ \mu = f^\sharp \circ \mu$, and therefore means that no loss of precision occurs by approximating a concrete computation of f on an abstract object $a \in A$ with the abstract computation of $f^\sharp$ on the same a.

Giacobazzi et al. [15] observed that completeness uniquely depends upon the abstraction map, namely, one may define a complete abstract semantic operation $f^\sharp : A \to A$ over A if and only if $\mu \circ f : A \to A$ is complete. Hence, an abstract domain $\mu \in \mathrm{uco}(C)$ is defined to be complete for f iff $\mu \circ f = \mu \circ f \circ \mu$ holds. This simple observation makes completeness an abstract domain property. The same observations are also true for forward completeness, which is therefore a domain property as well: μ is forward complete iff $f \circ \mu = \mu \circ f \circ \mu$. This justifies the terminology forward completeness and, dually, backward completeness for the first standard form of completeness. A constructive characterization of backward complete abstract domains is given in [15], under the assumption of dealing with Scott-continuous concrete functions. This result allows to systematically and constructively derive complete abstract domains from non-complete ones by minimal refinements: this complete minimal refinement of a domain A for a function f is called *backward complete shell* of A for f.

3 Partitions as Abstractions

Let Q be any (possibly infinite) set of states. We associate to a closure $\mu \in \mathrm{uco}(\wp(Q)_\subseteq)$ a state equivalence $\equiv_\mu$ on Q, i.e. a partition of Q, by identifying

those states that cannot be distinguished by the closure μ, namely those states belonging to the same set of fixpoints of the closure μ:

$$s \equiv_\mu s' \;\Leftrightarrow\; \forall S \in \mu.\, (s \in S \Leftrightarrow s' \in S).$$

It is easy to show that $s \equiv_\mu s'$ iff $\mu(\{s\}) = \mu(\{s'\})$. Hence, this allows us to view partitions as particular abstractions of $\wp(Q)$. We will denote by $\mathrm{par}(\mu) \in \mathrm{Part}(Q)$ the partition associated to the abstract domain $\mu \in \mathrm{uco}(\wp(Q))$. For example, for $Q = \{1,2,3\}$, the closures $\mu_1 = \{1, 12, 13, 123\}$, $\mu_2 = \{\varnothing, 1, 2, 3, 123\}$ and $\mu_3 = \wp(\{1,2,3\})$ all induce the same partition $\mathrm{par}(\mu) = \{\{1\}, \{2\}, \{3\}\}$. However, these closures carry additional information other than the underlying state partition, and this additional information allows us to distinguish them. It is then natural to say that a closure μ represents exactly a state partition when μ carries all this possible additional information, or, otherwise stated, when μ is the most precise among the closures inducing a given partition.

Definition 3.1. μ is *partitioning* if $\mu = \mathbb{P}(\mu) \stackrel{\text{def}}{=} \sqcap \{\eta \in \mathrm{uco}(\wp(Q)) \mid \equiv_\eta = \equiv_\mu\}$. $\mathrm{uco}^{\mathrm{p}}(\wp(Q))$ will denote the set of partitioning closures. □

The operator $\mathbb{P}$ is a refinement of abstract domains in the sense of [14], i.e., it can be proved that $\mathbb{P}$ is a lower closure operator on $\mathrm{uco}(\wp(Q))$. Accordingly, $\mathbb{P}$ will be called the *partitioning shell* operator. It turns out that partitioning closures can be characterized as follows.

Lemma 3.2. *Let $\mu \in \mathrm{uco}(\wp(Q))$. Then, $\mu \in \mathrm{uco}^{\mathrm{p}}(\wp(Q))$ iff μ is additive and $\{\mu(\{q\})\}_{q \in Q}$ is a partition of Q. Moreover, in this case, $\mathrm{par}(\mu) = \{\mu(\{q\})\}_{q \in Q}$.*

For instance, for all the above closures we have that $\mathbb{P}(\mu_i) = \wp(\{1,2,3\})$, and hence μ_1 and μ_2 are not partitioning. Also, the closure $\{\varnothing, 3, 12, 123\}$ is partitioning and represents the partition $\{12, 3\}$.

Lemma 3.2 allows us to see classical partition-based abstractions — i.e., partitions induced by a surjective abstraction map "h" in the style of Clarke et al. [5] — as a particular case of generic abstract domain through the following isomorphism between partitions and partitioning closures:

– $\mathrm{par} : \mathrm{uco}^{\mathrm{p}}(\wp(Q)) \to \mathrm{Part}(Q)$ is the restriction of the above operator par to partitioning closures, i.e. $\mathrm{par}(\mu) = \{\mu(\{q\}) \mid q \in Q\}$;

– $\mathrm{pcl} : \mathrm{Part}(Q) \to \mathrm{uco}^{\mathrm{p}}(\wp(Q))$ is defined as follows: $\mathrm{pcl}(P) \stackrel{\text{def}}{=} \mathbb{P}(\mathcal{M}(P)) = \lambda X \in \wp(Q).\, \cup \{B \in P \mid X \cap B \neq \varnothing\}$.

Theorem 3.3. *The mappings* par *and* pcl *are well defined and give rise to an order isomorphism between $\langle \mathrm{Part}(Q), \preccurlyeq \rangle$ and $\langle \mathrm{uco}^{\mathrm{p}}(\wp(Q)), \sqsubseteq \rangle$.*

4 Strongly Preserving Abstract Model Checking

We deal with specification languages $\mathcal{L}$ whose syntactic (state) formulae φ are inductively defined by a grammar:

$$\varphi ::= p \mid f(\varphi_1, ..., \varphi_n)$$

where $p \in AP$ ranges over a finite set of atomic propositions and $f \in Op$ ranges over a finite set of operators. Each operator $f \in Op$ will have an arity[1] $\sharp(f) > 0$.

The concrete semantic domain for interpreting formulae is the boolean algebra $\langle \wp(Q), \subseteq \rangle$, where Q is any (possibly infinite) set of states. The state semantics of formulae in $\mathcal{L}$ is determined by an interpretation function I such that for any $p \in AP$, $I(p) \in \wp(Q)$ and for any $f \in Op$, $I(f) : \wp(Q)^{\sharp(f)} \to \wp(Q)$. We will also use the notations $\mathbf{p}$ and $\mathbf{f}$ to denote, respectively, $I(p)$ and $I(f)$. As usual, the interpretation I on AP can be equivalently specified by a labelling function $\ell : Q \to \wp(AP)$ provided that $\ell(s) = \{p \in AP \mid s \in I(p)\}$ holds for any s. The *concrete state semantic function* $[\![\cdot]\!]_I : \mathcal{L} \to \wp(Q)$ is inductively defined as follows:

$$[\![p]\!]_I = \mathbf{p} \quad \text{and} \quad [\![f(\varphi_1, ..., \varphi_n)]\!]_I = \mathbf{f}([\![\varphi_1]\!]_I, ..., [\![\varphi_n]\!]_I).$$

We will freely use standard logical and temporal operators together with their usual interpretations: for example, $\wedge/\cap$, $\vee/\cup$, $\neg/\complement$, $\mathrm{EX}/\mathrm{pre}[R]$, etc.

If g is any synctatic operator with arity $\sharp(g) = n$ whose semantics is given by $\mathbf{g} : \wp(Q)^n \to \wp(Q)$ then we say that the language $\mathcal{L}$ is *closed under* g when for any $\varphi_1, ..., \varphi_n \in \mathcal{L}$, there exists some $\psi \in \mathcal{L}$ such that $\mathbf{g}([\![\varphi_1]\!]_I, ..., [\![\varphi_n]\!]_I) = [\![\psi]\!]_I$.

Let us now apply the standard abstract interpretation approach for defining abstract semantics. Consider any abstract domain $\mu \in \mathrm{uco}(\wp(Q))$. The *abstract semantic function* $[\![\cdot]\!]^\mu_I : \mathcal{L} \to \mu$ induced by the abstract domain μ evaluates any formula $\varphi \in \mathcal{L}$ to an abstract value $[\![\varphi]\!]^\mu_I$ belonging to μ. $[\![\cdot]\!]^\mu_I$ is induced by the abstraction μ (and the interpretation I) and is inductively defined as best correct approximation of the concrete semantics as follows:

$$[\![p]\!]^\mu_I = \mu(\mathbf{p}) \quad \text{and} \quad [\![f(\varphi_1, ..., \varphi_n)]\!]^\mu_I = \mu(\mathbf{f}([\![\varphi_1]\!]^\mu_I, ..., [\![\varphi_n]\!]^\mu_I)).$$

Generalizing strong preservation. Classical abstract model checking [5,6] is state-based, namely it relies on an abstract model which, like the concrete model, is a transition system. If $\mathcal{T} = (Q, R)$ is the concrete transition system then this classical approach is based on a surjective abstraction $h : Q \to A$ mapping concrete states into abstract states, namely a state partition $P_h \in \mathrm{Part}(Q)$ is required. This gives rise to an abstract model $\mathcal{A} = (A, R^\sharp)$ which is a transition system, where, typically, the abstract transition relation $R^\sharp$ is existentially defined as the following $R^{\exists\exists}$:

$$h(s_1)\, R^{\exists\exists}\, h(s_2) \;\Leftrightarrow\; \exists s'_1, s'_2. h(s'_1) = h(s_1) \;\&\; h(s'_2) = h(s_2) \;\&\; s'_1 R s'_2.$$

Abstract model checking then consists in checking a temporal formula φ specified in some language $\mathcal{L}$ in the abstract model $\mathcal{A}$: a *preservation theorem* ensures that if $a \models^{\mathcal{A}} \varphi$ and $h(s) = a$ then $s \models^{\mathcal{T}} \varphi$. In this classical state-based framework, strong preservation (s.p. for short) for $\mathcal{L}$ means that for any $\varphi \in \mathcal{L}$, if $h(s) = a$ then $a \models^{\mathcal{A}} \varphi \;\Leftrightarrow\; s \models^{\mathcal{T}} \varphi$. Loiseaux et al. [19] generalized this approach to more general abstract models where an abstraction relation $\sigma \subseteq Q \times A$ is used instead of the surjection $h : Q \to A$. However, the strong preservation results given there (cf. [19, Theorems 3 and 4]) require the hypothesis that the relation

[1] It is possible to consider generic operators whose arity is any possibly infinite ordinal, thus allowing, for example, infinite conjunctions or disjunctions.

σ is difunctional: as a consequence, the class of abstract models allowed by this framework is not really larger than the class of classical partition-based abstract models (see the detailed discussion by Dams et al. [12, Sect. 8.1]).

Following Dams [11, Sect. 6.1], the above classical state-based notion of strong preservation can be equivalently given through state equivalences as follows. The language $\mathcal{L}$ and the semantic function $[\![\cdot]\!]_I$ induce the following state logical equivalence $\equiv_{\mathcal{L}} \subseteq Q \times Q$: $s \equiv_{\mathcal{L}} s'$ iff $\forall \varphi \in \mathcal{L}. s \in [\![\varphi]\!]_I \Leftrightarrow s' \in [\![\varphi]\!]_I$. Let $P_{\mathcal{L}} \in \mathrm{Part}(Q)$ be the corresponding partition of Q. Then, a partition $P \in \mathrm{Part}(Q)$ is strongly preserving for $\mathcal{L}$ (and interpretation I) if $P \preccurlyeq P_{\mathcal{L}}$, while P is fully abstract[2] for $\mathcal{L}$ if $P = P_{\mathcal{L}}$. For most known temporal languages (e.g., CTL*, CTL-X, $\forall$CTL*, see [11]), if the partition P is s.p. for $\mathcal{L}$ then it turns out that the quotient transition system $(Q/_{\equiv_P}, R^{\exists\exists})$ is s.p. for $\mathcal{L}$. Moreover, if P is fully abstract then the quotient $(Q/_{\equiv_P}, R^{\exists\exists})$ is the smallest transition system (smallest in the number of states) that strongly preserves $\mathcal{L}$.

We consider now an equivalent formulation of strong preservation for partitions that will allow us to generalize the notion of strong preservation to generic abstract domains. We showed above how any partition $P \in \mathrm{Part}(Q)$ can be viewed as a partitioning closure $\mathrm{pcl}(P) \in \mathrm{uco}^{\mathrm{p}}(\wp(Q))$. Thus, any partition P induces the abstract semantics $[\![\cdot]\!]_I^P = [\![\cdot]\!]_I^{\mathrm{pcl}(P)} : \mathcal{L} \to \mathrm{pcl}(P)$. The following result characterizes strong preservation for P in terms of the associated closure $\mathrm{pcl}(P)$.

Lemma 4.1. *$P \in \mathrm{Part}(Q)$ is s.p. for $\mathcal{L}$ and I iff $\forall \varphi \in \mathcal{L}$ and $s \in Q$, $s \in [\![\varphi]\!]_I \Leftrightarrow \mathrm{pcl}(P)(\{s\}) \subseteq [\![\varphi]\!]_I^P$.*

Let us stress the paradigm shift stated by Lemma 4.1. This tells us that a partition $P \in \mathrm{Part}(Q)$ is s.p. for $\mathcal{L}$ and I if and only if to check that some system state $s \in Q$ satisfies some formula $\varphi \in \mathcal{L}$, i.e. $s \in [\![\varphi]\!]_I$, is equivalent to checking whether the abstract state associated to s, i.e. the block $\mathrm{pcl}(P)(\{s\})$ of P containing s, is less than or equal to, namely is contained in, the abstract semantics $[\![\varphi]\!]_I^P$ of φ, which is an element of the abstract domain $\mathrm{pcl}(P)$. Here, the key observation is that in our abstract interpretation-based framework partitions are just particular abstract domains. Thus, the above characterization leads to generalize the notion of strong preservation from partitions to generic abstract semantic functions as follows.

Definition 4.2. Let I be an interpretation for a language $\mathcal{L}$ (inducing the concrete semantic function $[\![\cdot]\!]_I$). Let $\mu \in \mathrm{uco}(\wp(Q))$ and let $[\![\cdot]\!]^\sharp : \mathcal{L} \to \mu$ be an abstract semantic function for $\mathcal{L}$. We say that $[\![\cdot]\!]^\sharp$ is *strongly preserving* for $\mathcal{L}$ and I if for any $S \subseteq Q$ and $\varphi \in \mathcal{L}$, $\mu(S) \subseteq [\![\varphi]\!]^\sharp \Leftrightarrow S \subseteq [\![\varphi]\!]_I$. □

Definition 4.2 generalizes classical state-based strong preservation, as characterized by Lemma 4.1, both to an arbitrary abstract domain $\mu \in \mathrm{uco}(\wp(Q))$ and to an arbitrary semantic function $[\![\cdot]\!]^\sharp : \mathcal{L} \to \mu$ evaluating formulae on μ. Hence, $[\![\cdot]\!]^\sharp$ may be different from the abstract semantics $[\![\cdot]\!]_I^\mu$ induced by the abstract domain μ. It turns out that indeed this notion of *strong preservation*

[2] Dams [11] uses the term "adequate".

is an abstract domain property, namely if a s.p. abstract semantics $[\![\cdot]\!]^\sharp$ may be defined on some abstract domain μ then the induced abstract semantics $[\![\cdot]\!]_I^\mu$ is s.p. as well, as stated by the following result.

Lemma 4.3. *If $[\![\cdot]\!]^\sharp : \mathcal{L} \to \mu$ is strongly preserving for $\mathcal{L}$ and I then $[\![\cdot]\!]_I^\mu$ is strongly preserving for $\mathcal{L}$ and I.*

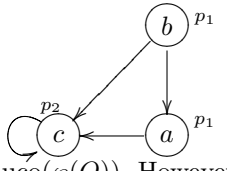
This result allows us to define without loss of generality strong preservation for abstract domains as follows: a closure $\mu \in \mathrm{uco}(\wp(Q))$ is strongly preserving for $\mathcal{L}$ and I when $[\![\cdot]\!]_I^\mu$ is s.p. for $\mathcal{L}$ and I.

As recalled above, the concrete semantic function $[\![\cdot]\!]_I$ induces a state partition $P_\mathcal{L}$, which is the fully abstract partition according to our definition, since $P_\mathcal{L}$ encodes the "best" strongly preserving partition, where "best" means coarsest (i.e. the greatest w.r.t. the ordering $\preceq$). In other terms, $P_\mathcal{L}$ is the coarsest partition such that the states in any block cannot be distinguished by the language $\mathcal{L}$. We aim at defining an analogous of $P_\mathcal{L}$ for closure-based abstractions, namely the "best" strongly preserving closure. Given a partition $P \in \mathrm{Part}(Q)$, the associated partitioning closure $\mathrm{pcl}(P) \in \mathrm{uco}^\mathrm{p}(\wp(Q))$ is completely determined by its behaviour on states $q \in Q$, namely on singletons $\{q\} \in \wp(Q)$, since $\mathrm{pcl}(P)$ is additive (cf. Lemma 3.2). Of course, this does not happen for a generic closure $\mu \in \mathrm{uco}(\wp(Q))$. This means that a generalization of $P_\mathcal{L}$ to closures must take into account the behaviour of the closure on all the subsets of Q. Thus, if we define, for $S \subseteq Q$, $S \models \varphi$ iff $\forall s \in S. s \models \varphi$, one might try the following definition: the "best" s.p. closure $\mu_\mathcal{L}$ for $\mathcal{L}$ is given by

$$\mu_\mathcal{L}(S) = \cup\{T \in \wp(Q) \mid \forall \varphi \in \mathcal{L}.\, S \models \varphi \Leftrightarrow T \models \varphi\}.$$

However, this does not work, as shown by the following example.

Example 4.4. Let us consider the simple transition system (Q, R) depicted in the figure and the language $\mathcal{L}$ generated by the grammar $\varphi ::= p \mid \mathrm{EX}\varphi$, where the set of the atomic propositions is $AP = \{p_1, p_2\}$ with interpretation $I(p_1) = \{a, b\}$ and $I(p_2) = \{c\}$ (and $I(\mathrm{EX}) = \mathrm{pre}[R]$). Note that $b \models \mathrm{EX}p_1$ while $a \not\models \mathrm{EX}p_1$. In this case, we have that $P_\mathcal{L} = \{a, b, c\} \in \mathrm{Part}(Q)$. By using the above definition of $\mu_\mathcal{L}$, it turns out that $\mu_\mathcal{L} = \lambda x.x \in \mathrm{uco}(\wp(Q))$. However, the closure $\mu = \{\varnothing, b, c, ab, abc\}$ is more abstract than $\mu_\mathcal{L}$ and still strongly preserving. In fact, it is not difficult to check that for any $\varphi \in \mathcal{L}$, $[\![\varphi]\!]_I^\mu = [\![\varphi]\!]_I$, and therefore, according to Definition 4.2, μ is strongly preserving for $\mathcal{L}$. □



Instead, we may note that if $\mu \in \mathrm{uco}(\wp(Q))$ is s.p. for $\mathcal{L}$ then the following property holds: for any $S \in \wp(Q)$ and $\varphi \in \mathcal{L}$, $S \models \varphi \Rightarrow \mu(S) \models \varphi$. In fact, if $S \models \varphi$ and $x \in \mu(S)$ then $\mu(\{x\}) \subseteq \mu(S)$ and therefore, since $\mu(S) \subseteq [\![\varphi]\!]_I^\mu \Leftrightarrow S \models \varphi$, we have that $\mu(\{x\}) \subseteq [\![\varphi]\!]_I^\mu$, and thus, again by strong preservation, $x \models \varphi$. Actually, it turns out that this weaker property characterizes the best s.p. closure for $\mathcal{L}$.

Definition 4.5. Let $[\![\cdot]\!]_I : \mathcal{L} \to \wp(Q)$ be the concrete semantic function. Then, we define $\mu_{\mathcal{L}} : \wp(Q) \to \wp(Q)$ as follows: for any $S \in \wp(Q)$,

$$\mu_{\mathcal{L}}(S) \stackrel{\text{def}}{=} \bigcup \{T \in \wp(Q) \mid \forall \varphi \in \mathcal{L} \,.\, S \models \varphi \Rightarrow T \models \varphi\}. \qquad \square$$

It is not hard to check that $\mu_{\mathcal{L}}$ is indeed a closure. It turns out that $\mu_{\mathcal{L}}$ actually is the right candidate for the best s.p. abstract domain.

Theorem 4.6. *Let $\mu \in \mathrm{uco}(\wp(Q))$. Then, μ is s.p. for $\mathcal{L}$ and I iff $\mu \sqsubseteq \mu_{\mathcal{L}}$.*

Thus, $\mu_{\mathcal{L}}$ actually is the "best" s.p. abstract domain, i.e., it is the most abstract s.p. closure: $\mu_{\mathcal{L}} = \sqcup\{\mu \in \mathrm{uco}(\wp(Q)) \mid \mu \text{ is s.p. for } \mathcal{L} \text{ and } I\}$. Moreover, it turns out that $\mu_{\mathcal{L}}$ is exactly the closure meet-generated by the set of concrete semantics of all the formulae in $\mathcal{L}$.

Proposition 4.7. $\mu_{\mathcal{L}} = \mathcal{M}(\{[\![\varphi]\!]_I \mid \varphi \in \mathcal{L}\})$.

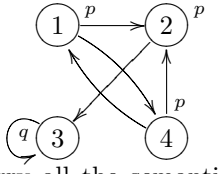
As a consequence, we have that μ is s.p. for $\mathcal{L}$ and I iff $\forall \varphi \in \mathcal{L}.\, [\![\varphi]\!]_I \in \mu$ iff $\forall \varphi \in \mathcal{L}. [\![\varphi]\!]_I = [\![\varphi]\!]_I^{\mu}$. Let us remark that as strong preservation for a partition P w.r.t. $\mathcal{L}$ means that P is a refinement of the state partition $P_{\mathcal{L}}$ induced by $\mathcal{L}$ likewise strong preservation for a closure μ means that μ is a refinement of the closure $\mu_{\mathcal{L}}$ of Definition 4.5 induced by $\mathcal{L}$. Strong preservation and full abstraction for partitions become particular instances, through the isomorphism of Theorem 3.3, of the corresponding notions for closures, as stated by the following result.

Proposition 4.8. *Let $P \in \mathrm{Part}(Q)$.*
(1) *$P_{\mathcal{L}} = \mathrm{par}(\mu_{\mathcal{L}})$ and $\mathrm{pcl}(P_{\mathcal{L}}) = \mathbb{P}(\mu_{\mathcal{L}})$.*
(2) *P is s.p. for $\mathcal{L}$ and I iff $P \preccurlyeq \mathrm{par}(\mu_{\mathcal{L}})$ iff $\mathrm{pcl}(P) \sqsubseteq \mathbb{P}(\mu_{\mathcal{L}})$.*

Finally, it is natural to ask when the closure $\mu_{\mathcal{L}}$ induced by a language $\mathcal{L}$ is partitioning.

Proposition 4.9. *Let $\mathcal{L}$ be closed under possibly infinite logical conjunction. Then, $\mu_{\mathcal{L}}$ is partitioning iff $\mathcal{L}$ is closed under logical negation.*

Example 4.10. Consider the transition system (Q, R) depicted in the figure and the temporal language CTL with atomic propositions p and q where $I(p) = \{1, 2, 4\}$ and $I(q) = \{3\}$. Consider the partition $P = \{124, 3\} \in \mathrm{Part}(Q)$ induced by the interpretation I. It is well known [2] that the state partition $P_{\mathrm{CTL}} \in \mathrm{Part}(Q)$ induced by CTL can be obtained by refining P using the Paige-Tarjan [20] partition refinement algorithm. It is easy to check that $P_{\mathrm{CTL}} = \{14, 2, 3\}$. However, the partition P_{CTL} does not carry all the semantic information. By Proposition 4.8 (1), P_{CTL} is the state equivalence induced by μ_{CTL}. Also, by Proposition 4.7, μ_{CTL} is the Moore-closure of $\{[\![\varphi]\!]_I \mid \varphi \in \mathrm{CTL}\}$. In this very simple case, it is easy to check that $\mu_{\mathrm{CTL}} = \{\varnothing, 2, 3, 14, 23, 124, 134, 1234\}$. Thus, as expected from Proposition 4.8 (1), $P_{\mathrm{CTL}} = \mathrm{par}(\mu_{\mathrm{CTL}})$. Since CTL is closed under logical negation, by



Proposition 4.9, it turns out that μ_{CTL} is partitioning and $\mu_{\mathrm{CTL}} = \mathrm{pcl}(P_{\mathrm{CTL}})$. Of course, this is not always the case. As an example, consider the following sublanguage $\mathcal{L}$ of CTL: $\varphi ::= p \mid q \mid \varphi_1 \wedge \varphi_2 \mid \mathrm{EX}\varphi$. Then, $\{1,3,4\} \notin \mu_{\mathcal{L}}$: in fact, $\{1,3,4\}$ can be obtained as the semantics of the CTL formula $q \vee \mathrm{EX}p$, i.e. $[\![q \vee \mathrm{EX}p]\!]_I = \{1,3,4\}$, while it is easy to observe that it cannot be obtained from a formula in $\mathcal{L}$. In this case, $\mu_{\mathcal{L}} = \{\varnothing, 2, 3, 14, 23, 124, 1234\}$ and $P_{\mathcal{L}} = \{14, 2, 3\}$. Here, it turns out that $\mu_{\mathcal{L}}$ is not a partitioning closure, namely a loss of precision occurs in abstracting $\mu_{\mathcal{L}}$, through the mapping par, to the partition $P_{\mathcal{L}}$. □

5 Completeness and Strong Preservation

We need to consider forward completeness of abstract domains for generic n-ary semantic functions. Let C be any complete lattice, $f : C^n \to C$, with $n \geq 0$, and $\mu \in \mathrm{uco}(\wp(Q))$. Thus, μ is forward f-complete, or simply f-complete, when $f \circ \langle \mu, ..., \mu \rangle = \mu \circ f \circ \langle \mu, ..., \mu \rangle$, i.e., for any $\vec{x} \in \mu^n$, $f(\vec{x}) \in \mu$. If $F \subseteq \mathrm{Fun}(C)$, μ is F-complete when μ is f-complete for each $f \in F$. Note that when $f : C^0 \to C$, i.e. $f \in C$, μ is forward f-complete iff $f \in \mu$. Moreover, note that any $\mu \in \mathrm{uco}(C)$ is always forward meet-complete, because any closure operator is Moore-closed.

We first note that the *forward F-complete shell* refinement always exists.

Lemma 5.1. *Let $F \subseteq \mathrm{Fun}(\wp(Q))$ and $\mu \in \mathrm{uco}(C)$. Then, $\mathcal{S}_F(\mu) \stackrel{\mathrm{def}}{=} \sqcup \{\rho \in \mathrm{uco}(C) \mid \rho \sqsubseteq \mu, \rho \text{ is } F\text{-complete}\}$ is F-complete.*

We call $\mathcal{S}_F : \mathrm{uco}(C) \to \mathrm{uco}(C)$ the *F-complete shell* (or complete shell, when F is clear from the context) refinement, since $\mathcal{S}_F(\mu)$ is the most abstract F-complete domain which refines μ. As a straight consequence, the complete shell of a closure admits the following constructive fixpoint characterization.

Lemma 5.2. *Let $R_F : \mathrm{uco}(C) \to \wp(C)$ be defined as follows: $R_F(\mu) \stackrel{\mathrm{def}}{=} \{f(\vec{x}) \in C \mid f \in F, \vec{x} \in \mu^{\sharp(f)}\}$. Then, $\mathcal{S}_F(\mu) = \mathrm{gfp}(\lambda\rho.\mu \sqcap \mathcal{M}(R_F(\rho)))$.*

For finite state systems, for any $\mu \in \mathrm{uco}(C)$, the operator $\lambda\rho.\mu \sqcap \mathcal{M}(R_F(\rho)) : \mathrm{uco}(C) \to \mathrm{uco}(C)$ is trivially co-continuous and therefore its greatest fixpoint can be computed through the Kleene's iteration sequence. Moreover, for unary operators, the iterative computation of the fixpoint $\mathcal{S}_F(\mu)$ can be simplified by applying R_F just to the new sets added in the previous iteration step, as follows.

Lemma 5.3. *Let C be finite, $F \subseteq C \to C$ and $\mu \in \mathrm{uco}(C)$. Define inductively $\mu_0 \stackrel{\mathrm{def}}{=} \mu$, $\mu_1 \stackrel{\mathrm{def}}{=} \mathcal{M}(\mu_0 \cup R_F(\mu_0))$, and, for $i \in \mathbb{N}$, $\mu_{i+2} \stackrel{\mathrm{def}}{=} \mathcal{M}(\mu_{i+1} \cup R_F(\mu_{i+1} \smallsetminus \mu_i))$. Then, there exists $n \in \mathbb{N}$ such that $\mathcal{S}_F(\mu) = \mu_n$.*

Let us now turn to strong preservation. Given a language $\mathcal{L}$, our goal is to single out a set of operators F such that refining a closure η for F-completeness is equivalent to refining η in order to get strong preservation for $\mathcal{L}$. The semantics of $\mathcal{L}$ is determined by the interpretations **AP** and **Op** of, respectively, the set of atomic propositions AP and the set of operators Op. Thus, **Op** is the obvious candidate for F. Moreover, we know (cf. Theorem 4.6) that an atomic

proposition p is strongly preserved by a domain η if and only if $\mathbf{p} \in \eta$. Also, recall that in partition refinement algorithms like Paige-Tarjan [20] for CTL and Groote-Vaandrager [16] for CTL-X, the interpretation of the atomic propositions determine the blocks of the initial partition, or, otherwise stated, the blocks of the partition to refine give the atomic propositions of the language. Likewise, here the fixpoints of the initial closure η provide the interpretation of the atomic propositions of $\mathcal{L}$. This is more general, since η need not to be a partition of system states. This can be formalized by associating to any set of sets $S \subseteq \wp(Q)$ a set of atomic propositions $AP_S = \{p_T \mid T \in S\}$ that are interpreted by the interpretation function I_S defined by $I_S(p_T) = T$. Also, given a closure $\eta \in \mathrm{uco}(\wp(Q))$ and a language $\mathcal{L}$ with operators ranging in Op, we consider the language $\mathcal{L}_\eta$ where AP_η is the set of atomic propositions while the operators still are in Op. The interpretation for $\mathcal{L}_\eta$ therefore is $I_\eta = \eta \cup \mathbf{Op}$. Then, the following key result shows the announced relationship between forward complete shells and strong preservation.

Theorem 5.4. *Let $\eta \in \mathrm{uco}(\wp(Q))$ and $\mathcal{L}$ be a language with operators in Op closed under logical conjunction. Then, $\mathcal{S}_{\mathbf{Op}}(\eta)$ is the most abstract closure which refines η and is s.p. for $\mathcal{L}_\eta$. In particular, $\mathcal{S}_{\mathbf{Op}}(\eta) = \mu_{\mathcal{L}_\eta}$.*

The opposite direction is also interesting: given a language $\mathcal{L}$, the following result characterizes the "best" s.p. closure $\mu_{\mathcal{L}}$ for $\mathcal{L}$ as a forward complete shell of a closure associated to $\mathcal{L}$. This comes as a straight consequence of Theorem 5.4.

Corollary 5.5. *Let $\mathcal{L}$ be given by AP and Op, let I be the interpretation and let $\mathcal{L}$ be closed under logical conjunction. Let $\mu_{AP} \stackrel{\text{def}}{=} \mathcal{M}(\{I(p) \mid p \in AP\})$. Then, $\mu_{\mathcal{L}} = \mathcal{S}_{\mathbf{Op}}(\mu_{AP})$.*

It is also worth remarking that, as a consequence of Proposition 4.8 (1), the state equivalence induced by the language $\mathcal{L}_\mu$ can be retrieved from the closure $\mathcal{S}_{\mathbf{Op}}(\mu)$: $\equiv_{\mathcal{S}_{\mathbf{Op}}(\mu)} = \equiv_{\mathcal{L}_\mu}$.

Theorem 5.4 provides a clean and precise generalization of the many existing coarsest partition refinement algorithms from an abstract interpretation perspective. Indeed, the coarsest refinement of a given partition which is strongly preserving for a given language can be characterized using our abstract domain-based approach as follows.

Corollary 5.6. *Let $\mathcal{L}$ be closed under logical conjunction and $P \in \mathrm{Part}(Q)$.*
(1) *Let P^r be the coarsest partition refinement of P which is strongly preserving for $\mathcal{L}$ and I_P. Then, $P^r = \mathrm{par}(\mathcal{S}_{\mathbf{Op}}(\mathcal{M}(P)))$.*
(2) *Let $\mathcal{L}$ be closed under logical negation and let P^r be the coarsest partition refinement of P which is strongly preserving for $\mathcal{L}$ and I_P. Then, $\mathrm{pcl}(P^r) = \mathcal{S}_{\mathbf{Op}}(\mathrm{pcl}(P))$.*

Note that, by the corollary above, in general the closure-based complete refinement of a partitioning closure $\mathrm{pcl}(P)$ associated to a partition P does not provide the closure associated to the corresponding partition-based refinement,

but a more abstract closure. The following result shows that a closure is partitioning iff it is forward complete for the complement $\complement$. As a consequence, when the language is closed under logical negation the two refinement techniques agree.

Lemma 5.7. *If $\mu \in \mathrm{uco}(\wp(Q))$ then, μ is partitioning iff μ is forward $\complement$-complete.*

We can draw the following consequence. Let $\mathcal{L}$ be closed under logical conjunction. Then, by Theorem 5.4 and Lemma 5.7, $\mathcal{L}$ is closed under logical negation iff for any $\mu \in \mathrm{uco}(\wp(Q))$, $\mathcal{S}_{\mathbf{Op}}(\mu)$ is partitioning. Hence, for languages which are not closed under logical negation, the output partition of any partition refinement algorithm for achieving strong preservation for $\mathcal{L}$ is not the optimal, i.e. "best", abstraction refinement.

6 Application to Behavioural State Equivalences

We apply the above results to a number of standard temporal languages. It is well known that some of these languages like CTL and CTL-X induce state equivalences which coincide with standard behavioural equivalences used in process algebra like bisimulation and stuttering equivalence. We obtain as consequences a new characterization of these behavioural equivalences in terms of forward complete abstract interpretations which shows the following remarkable fact: these behavioural properties for a state equivalence $\sim$ can be interpreted as suitable completeness properties of $\sim$ viewed as an abstract interpretation. Due to lack of space we do not consider here the case of CTL-X and stuttering equivalence.

Bisimulation. Let $\mathcal{T} = (Q, R, AP, \ell)$ be a Kripke structure (R is assumed to be total). Let us recall the well known notion of bisimulation. A symmetric relation $\sim \subseteq Q \times Q$ is a bisimulation on the Kripke structure $\mathcal{T}$ if for any $s, s' \in Q$ such that $s \sim s'$: (1) $\ell(s) = \ell(s')$; (2) for any $t \in Q$ such that $s \xrightarrow{R} t$, there exists $t' \in Q$ such that $s' \xrightarrow{R} t'$ and $t \sim t'$. In particular, a partition $P \in \mathrm{Part}(Q)$ is called a bisimulation on $\mathcal{T}$ when the relation $\equiv_P$ is a bisimulation on $\mathcal{T}$. The (set-theoretically) largest bisimulation relation exists and will be denoted by $\sim_{\mathrm{bis}}$. It is well known [2] that $\sim_{\mathrm{bis}}$ is an equivalence relation, called *bisimulation equivalence*, which coincides with the state logical equivalence induced by the language CTL, i.e., $\sim_{\mathrm{bis}} = \equiv_{\mathrm{CTL}}$ (the same holds for CTL*). On the other hand, it is also known that it is enough to consider Hennessy-Milner logic [17], i.e. a language $\mathcal{L}_1$ allowing full propositional logic, i.e. conjunction plus negation, and the temporal connective EX, in order to have a state equivalence $\equiv_{\mathcal{L}_1}$ which coincides with $\equiv_{\mathrm{CTL}}$. The language $\mathcal{L}_1$ is then defined by the following grammar:

$$\varphi ::= p \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \mathrm{EX}\varphi$$

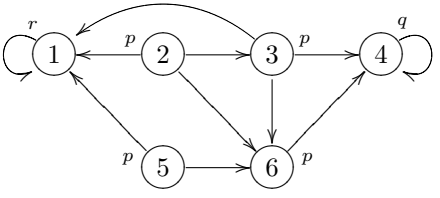
where $p \in AP$ and the interpretation I for the connectives is standard.

It is also well known that the bisimulation equivalence $\sim_{\mathrm{bis}}$ can be obtained through the Paige-Tarjan [20] partition refinement algorithm on the input partition determined by the interpretation of atomic propositions, i.e., the partition $\mathrm{par}(\mu_{AP})$ where $\mu_{AP} = \mathcal{M}(\{\mathbf{p} \mid p \in AP\})$. Here, exploiting Theorem 5.4, we get the characterization of bisimulation equivalence in terms of forward completeness (points (1) and (3) below) and of the Paige-Tarjan partition refinement algorithm as a complete shell refinement (points (2) and (4) below).

Corollary 6.1. *Let $\mathcal{T}$ be finite and $P \in \mathrm{Part}(Q)$.*
(1) *P is a bisimulation on $\mathcal{T}$ iff $pcl(P)$ is forward complete for $\{\mathbf{p} \mid p \in AP\} \cup \{\mathrm{pre}[R]\}$.*
(2) *Let P^r be the output partition of the Paige-Tarjan algorithm on input P. Then, $\mathrm{pcl}(P^r) = \mathcal{S}_{\{\mathtt{C},\mathrm{pre}[R]\}}(\mathcal{M}(P))$.*
(3) $\sim_{\mathrm{bis}} = \equiv_{\mathcal{S}_{\{\mathtt{C},\mathrm{pre}[R]\}}(\mu_{AP})}$.
(4) *For any $\mu \in \mathrm{uco}(\wp(Q))$, $\mathcal{S}_{\{\mathtt{C},\mathrm{pre}[R]\}}(\mu) = \mathrm{pcl}(P^r_\mu)$, where P^r_μ is the output partition of the Paige-Tarjan algorithm on input $\mathrm{par}(\mu)$.*

In our abstract interpretation-based terminology, given a generic closure $\mu \in \mathrm{uco}(\wp(Q))$, $\mathcal{S}_{\{\mathtt{C},\mathrm{pre}[R]\}}(\mu)$ is the most abstract refinement of μ which is s.p. for $\mathcal{L}_1$ (where the atomic propositions are determined by μ). Since the operators of $\mathcal{L}_1$ include both logical conjunction and negation, by Lemma 5.7, this complete shell is always partitioning, i.e. it is a representation of a partition by a closure.

Example 6.2. Let us consider the transition system in the figure, taken from [11, Sect. 6.6]. Also, consider the partition $P = \{1, 4, 2356\}$ which induces the atomic propositions $AP = \{p, q, r\}$, where $\mathbf{p} = \{2356\}$, $\mathbf{q} = \{4\}$ and $\mathbf{r} = \{1\}$.



It turns out that P is not a bisimulation. This can be checked on the abstract interpretation side by Corollary 6.1 (1): in fact, $\mu = \mathrm{pcl}(P) = \{\varnothing, 1, 4, 2356, 14, 12356, 23456, 123456\}$ is not pre$[R]$-complete, because, for instance, $\mathrm{pre}[R](\{1\}) = \{1235\} \notin \mathrm{pcl}(P)$. Obviously, this logically corresponds to the fact that for $\mathrm{EX}\, r \in \mathcal{L}_1$, $[\![\mathrm{EX}r]\!] = \{1, 2, 3, 5\}$ while $[\![\mathrm{EX}r]\!]^\mu = \mu(\{1235\}) = \{12356\}$. Using Definition 4.2 of strongly preserving closure, this corresponds to the fact that $\mu(\{6\}) = \{2356\} \subseteq [\![\mathrm{EX}r]\!]^\mu$ while $6 \notin [\![\mathrm{EX}r]\!]$. It is easy to check that the Paige-Tarjan algorithm on the input partition P yields the partition $P^r = \{1, 2, 3, 4, 5, 6\}$. Thus, by Corollary 6.1 (4), we have that $\mathcal{S}_{\{\mathtt{C},\mathrm{pre}[R]\}}(\mu) = \mathrm{pcl}(P^r) = \wp(Q)$. Thus, for any $S \subseteq Q$ there exists a formula $\varphi \in \mathcal{L}_1$ such that $[\![\varphi]\!] = S$. □

Simulation equivalence. Consider the language $\mathcal{L}_2$ obtained from $\mathcal{L}_1$ by dropping logical negation, namely $\mathcal{L}_2$ is defined by the following grammar:

$$\varphi ::= p \mid \varphi_1 \wedge \varphi_2 \mid \mathrm{EX}\varphi$$

It is known — see for example the handbook chapter by van Glabbeek [22, Sect. 8] — that the state equivalence $\equiv_{\mathcal{L}_2}$ induced by $\mathcal{L}_2$ coincides with simulation equivalence. Let us briefly recall the notion of simulation relation. Let $\mathcal{T} = (Q, R, AP, \ell)$ be a Kripke structure. A relation $\sigma \subseteq Q \times Q$ is a simulation if for any $s, s' \in Q$ such that $s\sigma s'$: (1) $\ell(s) = \ell(s')$; (2) if, for some $t \in Q$, $s \xrightarrow{R} t$ then there exists some $t' \in Q$ such that $s' \xrightarrow{R} t'$. Then, *simulation equivalence* $\sim_{\mathrm{sim}} \subseteq Q \times Q$ is defined as follows: $s \sim_{\mathrm{sim}} s'$ iff there exist two simulation relations $\sigma, \tau \subseteq Q \times Q$ such that $s\sigma s'$ and $s'\tau s$. A number of algorithms for computing the partition $\sim_{\mathrm{sim}}$, which coincides with $P_{\mathcal{L}_2}$, have been designed (e.g. [1,3,18]). Here, as a consequence of Theorem 5.4, we get the following characterization of simulation equivalence in terms of forward completeness (recall that μ_{AP} is the closure determined by the interpretation of atomic propositions AP):

Corollary 6.3. $\sim_{\mathrm{sim}} = \equiv_{\mathcal{S}_{\mathrm{pre}[R]}(\mu_{AP})}$.

Moreover, as argued after Lemma 5.7, since $\mathcal{L}_2$ is not closed under logical negation, we have that the output partition P^r of simulation equivalence computed by the aforementioned algorithms is not optimal for the strong preservation of $\mathcal{L}_2$, in the sense that the partitioning closure $\mathrm{pcl}(P^r)$ does not coincide with the set of formula semantics $\{[\![\varphi]\!]_\ell \mid \varphi \in \mathcal{L}_2\}$.

Example 6.4. Consider the transition system of Example 6.2. Let us compute the simulation equivalence $\sim_{\mathrm{sim}}$ by resorting to Corollary 6.3 and to the iterative method of Lemma 5.3. Let $\mu_{AP} = \mathcal{M}(\{1, 4, 2356\}) = \{\varnothing, 1, 4, 2356, 123456\}$.

$$\mu_0 = \mu_{AP}$$

$$\begin{aligned}\mu_1 &= \mathcal{M}(\mu_0 \cup R_{\mathrm{pre}[R]}(\mu_0)) \\ &= \mathcal{M}(\mu_0 \cup \{1235 = \mathrm{pre}[R](\{1\}), 346 = \mathrm{pre}[R](\{4\}), 235 = \mathrm{pre}[R](\{2356\})\}) \\ &= \{\varnothing, 1, 1235, 235, 2356, 3, 346, 36, 4, 123456\}\end{aligned}$$

$$\begin{aligned}\mu_2 &= \mathcal{M}(\mu_1 \cup R_{\mathrm{pre}[R]}(\mu_1 \smallsetminus \mu_0)) = \mathcal{M}(\mu_1 \cup \{23456 = \mathrm{pre}[R](\{346\})\} \\ &= \{\varnothing, 1, 1235, 2, 235, 2356, 23456, 3, 346, 36, 4, 123456\} \qquad \text{(fixpoint)}\end{aligned}$$

Thus, $\mu_{\mathcal{L}_2} = \mu_2 = \mathcal{S}_{\mathrm{pre}[R]}(\mu_{AP})$ and $\equiv_{\mu_2} = \sim_{\mathrm{sim}}$, i.e., the partition associated to the simulation equivalence is $P_{\mathcal{L}_2} = \{1, 2, 3, 4, 5, 6\}$. As expected, note that $\mathrm{pcl}(P_{\mathcal{L}_2}) \sqsubset \mathcal{S}_{\mathrm{pre}[R]}(\mu_{AP})$. Also note that $P_{\mathcal{L}_2}$ is the same partition obtained for bisimulation in Example 6.2, although the closures $\mu_{\mathcal{L}_1}$ and $\mu_{\mathcal{L}_2}$ are well different. □

7 Conclusion

We designed an abstract interpretation-based framework to study the properties of strong preservation of abstract model checking, where classical abstract model checking systems based on state partitions are embedded as particular abstract interpretations. Our main result showed that the minimal refinement of an abstract domain for achieving the completeness property w.r.t. the semantic operators of some language $\mathcal{L}$ is exactly equivalent to the minimal refinement of the corresponding abstract model checking in order to get strong preservation for $\mathcal{L}$. It is worth mentioning that we exploited the results in this paper to devise a generalized abstract interpretation-based Paige-Tarjan partition refinement algorithm which is able to compute the minimal refinements of abstract models which are strongly preserving for a generic inductively defined language [21].

Acknowledgements. We wish to thank Mila Dalla Preda and Roberto Giacobazzi who contributed to the early stage of this work. This work was partially supported by the FIRB Project "Abstract interpretation and model checking for the verification of embedded systems" and by the COFIN2002 Project "CoVer".

References

1. B. Bloom and R. Paige. Transformational design and implementation of a new efficient solution to the ready simulation problem. *Sci. Comp. Program.*, 24(3):189–20, 1995.
2. M.C. Browne, E.M. Clarke and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *TCS*, 59:115–131, 1988.
3. D. Bustan and O. Grumberg. Simulation-based minimization. *ACM TOCL*, 4(2):181–204, 2003.
4. E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In *Proc. CAV'00*. LNCS 1855:154–169, 2000.
5. E.M. Clarke, O. Grumberg and D. Long. Model checking and abstraction. *ACM TOPLAS*, 16(5):1512–1542, 1994.
6. E.M. Clarke, O. Grumberg and D.A. Peled. *Model checking*. The MIT Press, 1999.
7. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. 4th ACM POPL*, pp. 238–252, 1977.
8. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. 6th ACM POPL*, pp. 269–282, 1979.
9. P. Cousot and R. Cousot. Temporal abstract interpretation. In *Proc. 27th ACM POPL*, pp. 12–25, 2000.
10. M. Dalla Preda. *Completeness and stability in abstract model checking*. Master Thesis, Univ. of Verona, 2003.
11. D. Dams. *Abstract interpretation and partition refinement for model checking*. Ph.D. Thesis, Eindhoven Univ., 1996.
12. D. Dams, O. Grumberg, and R. Gerth. Abstract interpretation of reactive systems. *ACM TOPLAS*, 16(5):1512–1542, 1997.
13. R. Giacobazzi and E. Quintarelli. Incompleteness, counterexamples and refinements in abstract model checking. In *Proc. SAS '01*, LNCS 2126:356–373, 2001.
14. R. Giacobazzi and F. Ranzato. Refining and compressing abstract domains. In *Proc. 24th ICALP*, LNCS 1256:771–781, 1997.
15. R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *J. ACM*, 47(2):361–416, 2000.
16. J.F. Groote and F. Vaandrager. An efficient algorithm for branching bisimulation and stuttering equivalence. In *Proc. ICALP '90*, LNCS 443:626–638, 1990.
17. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.
18. M.R. Henzinger, T.A. Henzinger and P.W. Kopke. Computing simulations on finite and infinite graphs. In *Proc. 36th FOCS*, pp. 453–462, 1995.
19. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6:1–36, 1995.
20. R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987
21. F. Ranzato and F. Tapparo. Generalizing the Paige-Tarjan partition refinement algorithm through abstract interpretation. Manuscript, Univ. of Padova, 2004.
22. R.J. van Glabbeek. The linear time - branching time spectrum. In *Handbook of Process Algebra*, pp. 3–99, 2001.