Observational Preorders for Alternating Transition Systems

Romain Demangeon[1], Catalin Dima[2], and Daniele Varacca[2]

[1] LIP6, Sorbonne University Paris, France,
romain.demangeon@sorbonne-universite.fr
[2] LACL, Université Paris-Est Créteil, France,
{dima,daniele.varacca}@u-pec.fr

Abstract. We define two notions of observational preorders on Alternating transition systems. The first is based on the notion of being able to enforce a property. The second is based on the idea of viewing strategies as a generalised notion of context. We show that alternating simulation as defined by Alur et al. [3] is a sound proof technique for the enforcing preorder and a complete proof technique for the "contextual" preorder. We conclude by comparing alternating simulation with the classic notion of simulation on labelled transition systems.

1 Introduction

Several process calculi have been defined to model concurrent systems, such as the Calculus of Communicating Systems (CCS) [11], or the π-calculus [12]. In these syntactic frameworks, there is a canonical way to define a preorder between terms. It consists in giving an unlabelled "reduction" semantics of the terms, some notions of basic observation, and then define the preorders *contextually*: one term P is less than a term Q if for every context C, if $C[P]$ produces some basic observation, so does $C[Q]$. This definition is often easy to give, and it's also rather convincing. The usual narrative then says that proving that two terms are in the relation is hard, due to the quantification over all contexts. Labelled semantics comes to the rescue by means of theorems that say that labelled similarity is included in (or coincide with) contextual preorder. This is the case for CCS and the π-calculus for instance.

In this paper we address the following question:

Can we generalise a notion of contextual preorder
to a setting where there is no syntax around?

In particular, how can we generalise the notion of context?

We will consider the model of *Alternating Transition Systems* (ATS) proposed by Alur, Henzinger and Kupferman [2]. In this setting, states can be described by boolean properties and a notion of alternation between an *Agent* and an *Opponent* is present. ATSs come with a notion of *strategy.* The Agent and the Opponent follow strategies according to some rules, and the interaction between the strategies produces a run of the system. We then can make several observations on the run, we can for instance observe the sequence of boolean properties encountered during this run.

The first preorder we define is based on the notion of *enforcing a specification* (which slightly generalizes [14, 15]). The Agent can enforce a specification if it has a winning strategy for it, that is a strategy such that, whatever the Opponent does, the resulting execution satisfies the specification. After formalising a suitable general notion of specification, we propose to define a preorder as follows: an ATS P is less than an ATS Q if for every specification φ, if Agent can enforce φ on P then she can enforce it on Q.

To define a second preorder, we generalise the definition of contextual preorder to ATS using the intuition that *strategies generalise the notion of context.* With this intuition in mind, we say that an ATS P is less than an ATS Q if for every pair of strategies σ_A, σ_O (of Agent and Opponent), if the run produced by these strategies on P exhibits some properties, so does the run produced by the *same* strategies on Q. One problem with this intuition is that the notion of strategy, as defined by Alur et al., is very much bound to the system. We cannot directly apply a strategy for P to a different system Q. We overcome this difficulty by defining a way of "transfering" a strategy from a system to another.

ATS come also equipped with a notion of *alternating simulation* [3], which is used to define another preorder, called *Alternating Similarity.* The natural question to ask is: *what is the relation between these preorders?* We show that alternating similarity and the generalised contextual preorders coincide, and they are both stronger than the enforcing preorder.

In order to carry out our proofs, we propose a simplified presentation of ATS, using a formalism close to labelled transition systems. In the syntactic models, labels are useful when the system interacts with a context, but they are not necessary to the more powerful notion of strategy. We still like to rephrase the notion of ATS in a labelled setting. In this way we are able to stress the connections between the notion of alternating simulation, and Milner and Park's notion of simulation on LTS. We also argue that the labelled presentation may play some role in future extensions of our work.

Plan The main contributions of our work are :

1. the introduction of the enforcing preorder on ATS;
2. the introduction of a contextual preorder which compares ATS by matching their strategies; using a *choice correspondence* operation;
3. the translation of the alternating simulation preorder to our labelled presentation of ATS;
4. the proof that two preorders coincide and they are stronger than the third.

Section 2 presents our new definition for Alternating Transition Systems, considering them as agent/opponent games on LTS, and defines the enforcing preorder. Section 3 introduces our version of *alternating simulation*(AS) as a way to compare ATS taking into account just how labels group together outcoming transitions. Section 4 defines a pre-order relation on ATS which compares how strategies for the two ATS can interact. A pair of mapping on states and labels, called choice correspondence, allows one to compare two systems using actions with different labels. We call this the *Morris preorder*. Section 5 shows that the largest alternating simulation and the Morris preorder coincide. We also show that they are both stronger (we conjecture strictly) than the enforcing preorder. We conclude by discussing the symmetric versions of our relations. We also define a name-aware version of the alternating simulation and compare it with Park and Milner's simulation.

2 An alternating view of transition systems

Alternating Transition Systems (ATS) were introduced by Alur, Henzinger and Kupferman [2] to model open systems. In this model, the execution of a system is produced by the action of different agents. They are a very useful model, that has been used extensively in research related with synthesis and verification [1, 4, 16, 9] (to cite only a few). However it lacks the simplicity of the notion of Labelled Transition Systems (LTS) [13] that is at the basis of the semantics of process algebras. In this section we propose to see traditional LTS as a simplified version of ATS.

2.1 Definition

Alur, Henzinger and Kupferman define ATS by having several players that can form *coalitions*. In this paper, to make things simple, we will only consider two players: the *Agent* and the *Opponent*. As in the definition by Alur, Henzinger and Kupferman, these players make choices that produce an eventual execution of the system. However the choices are made on a standard LTS. The intuition is that at each state of the system, the Agent chooses a label l (among all the labels that are allowed in that state), while the Opponent chooses one of the transitions that are labelled by l. A *Labelled Transition System* (LTS) is a tuple $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ where

1. S is the set of states and L a the set of labels, with $s_0 \in S$ the initial state.
2. $\mathcal{T} \subseteq S \times L \times S$ is the transition relation.
3. P is the set of atomic observations and $(P, \leq)$ forms a discrete partial order.
4. $O : S \to P$ is the observation function.

We write $s \xrightarrow{l}$ if there exists s' such that $s \xrightarrow{l} s'$.

In Figure 1, transitions with the same labels are grouped in *bunch* of transitions, making explicit how the game proceed. From state s_0, Agent chooses one bunch of transitions labelled by either l or k; then Opponent chooses a state reachable by a transition taken from the chosen bunch. For instance, if Agent chooses l, Opponent can choose s_2 but not s_4.

A finite or infinite *run* of an LTS is an alternating sequence of states and labels, starting in the inital state, respecting the transition relation. The set of finite runs of an LTS $\mathcal{L}$ is denoted by $runs(\mathcal{L})$. The set of infinite runs of an LTS

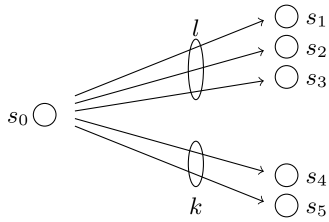


Fig. 1. Bunches of transitions in ATS.

$\mathcal{L}$ is denoted by $runs_\infty(\mathcal{L})$, the set of finite runs ending with a state $runs_\bullet(\mathcal{L})$ and the set of finite runs ending with a label $runs_\rightarrow(\mathcal{L})$. The length of a run ρ is denoted $\ell(\rho)$ (and equals ∞ for infinite runs). Furthermore, for any $k \leq \ell(\rho)$, the $(k+1)$-th item (state or action) in the run ρ is denoted $\rho[k]$, with the first item being denoted $\rho[0]$, while the prefix of length $k+1$ is denoted $\rho[\leq k]$.

The observation function can be extended homomorphycally to a map $O : runs_\infty(\mathcal{L}) \rightarrow P^\omega$ which we call the *infinite observation* of the run. Given two infinite observations $Q_1 = (p_0 \dots p_n \dots)$, $Q_2 = (r_0 \dots r_n \dots)$ where for each $i \geq 0$ we have $p_i, r_i \in P$, we say that $Q_1 \leq Q_2$ if for each $i \geq 0, p_i \leq r_i$. If $(s, l, s') \in \mathcal{T}$ we will write $s \xrightarrow{l} s'$.

As we discuss in details later, the identity of labels is not important. In the presence of the syntax of a process algebras, the identity of a label allows synchronisations of different subsystems. But in the framework we discuss here, labels are just a means of *grouping together* different transitions. We will allow relabelling as long as they produce the same groups (or *bunches*) of transitions.

As the intuition suggests, the successive moves of the Agent and the Opponent produce an infinite execution of the system, as we formalise now.

2.2 Strategies and Observations

Given an LTS $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$, a *strategy for the Agent* is a function $\sigma_A : runs_\bullet(\mathcal{L}) \rightarrow L$ such that if $\sigma_A(s_0 l_0 \dots s_n) = l_n$ then $s_n \xrightarrow{l_n}$. A *strategy for the Opponent* is a function $\sigma_O : runs_\rightarrow(\mathcal{L}) \rightarrow S$ such that if $\sigma_O(s_0 l_0 \dots s_n l_n) = s'$ then $s_n \xrightarrow{l_n} s'$. For simplicity, we can suppose that each state of an LTS has at least one outgoing transition (towards a sink state if necessary). This allows us to define strategies as total functions.

The combination of two strategies produces an infinite run. Given an LTS $\mathcal{L} = (S, s_0, L, \mathcal{T})$, a strategy for the Agent σ_A, a strategy for the Opponent σ_O, we define an infinite run $r = s_0 l_0 \dots s_n l_n \dots \in runs_\infty(\mathcal{L})$, denoted $\rho[\mathcal{L}, \sigma_A, \sigma_O]$, as follows:

- $l_0 = \sigma_A(s_0)$; $l_n = \sigma_A(s_0 \dots s_n)$;
- $s_{n+1} = \sigma_O(s_0 \dots s_n l_n)$;

For the purpose of this paper we will define a *specification* to be an *upward closed* set of infinite observations, so that if an infinite observation satisfies a given specification φ, a larger observation satisfies φ also.

Definition 1. *We say that a run r* satisfies the specification φ *(denoted $r \vDash \varphi$) if $O(r) \in \varphi$.*

2.3 Enforcing preorder

We want to define a preorder between systems based on the above notion of observation. Given two LTS $\mathcal{L}, \mathcal{L}'$, when can we say that $\mathcal{L}'$ is "better" than $\mathcal{L}$? We propose the following intuition: if Agent can enforce some specification on $\mathcal{L}$, then she must be able to enforce it on $\mathcal{L}'$.

Definition 2. *We say that Agent* can enforce *a specification φ on $\mathcal{L}$ if*

$$\exists \sigma_A \forall \sigma_O \rho[\mathcal{L}, \sigma_A, \sigma_O] \vDash \varphi.$$

We can now formalise the notion of enforcing preorder:

Definition 3. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ and $\mathcal{L}' = (S', s'_0, L', \mathcal{T}', P, \leq, O')$ be two LTSs sharing the same observation order $(P, \leq)$. We say that $\mathcal{L} \leq \mathcal{L}'$ if for any specification φ, if Agent can enforce φ on $\mathcal{L}$ then Agent can enforce φ on $\mathcal{L}'$.*

3 Alternating simulations

Alur et al. [3] introduce a notion of bisimulation for ATS, called *alternating bisimulation*. This notion has some resemblance to the notion introduced by Park and Milner [11] for LTS, but there are also major differences, not least because the model they apply to are different.

In this section we propose to define a notion of Alternating (bi)simulation for LTS that follows the intuition explained in the previous section. At first, we will only study the notion of simulation - we will discuss the symmetric relations at the end of the paper.

The notion of simulation by Park and Milner takes the identity of labels very seriously. There are two main reasons for this. First, we argue that this is due to the fact that LTS usually model syntactic process algebras where labels are important for synchronisation. In our setting, however, labels are only needed to group transitions together. Secondly, in the world of process algebras, labels also play the role of observations. In this paper, we use a more general notion of observation.

Therefore our definition of simulation will allow "relabelling". This brings us to the following definition.

Definition 4. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ and $\mathcal{L}' = (S', s'_0, L', \mathcal{T}', P, \leq, O')$ be two LTSs sharing the same observation order $(P, \leq)$. An* Alternating simulation *(AS) between them is, is a binary relation $\mathcal{R} \subseteq S \times S'$ such that whenever $s \; \mathcal{R} \; s'$, then $O(s) \leq O(s')$ and:*

for all labels l s.t. $s \xrightarrow{l}$, there exists h s.t. $s' \xrightarrow{h}$ and for all t' s.t. $s' \xrightarrow{h} t'$, there exists t s.t. $s \xrightarrow{l} t$ and $t \mathrel{\mathcal{R}} t'$.

The largest AS, $\subset_{\mathtt{AS}}$, is called alternating similarity*. If there is a AS $\mathcal{R}$ s.t. $s_0 \mathrel{\mathcal{R}} s_0'$, we say $\mathcal{L} \subset_{\mathtt{AS}} \mathcal{L}'$.*

Remark 1. Note that, contrary to the notion in [3], we do not require $O(s) = O(s')$ in the first item above. The reason will come up later, when defining the "Morris preorder" in Definition 6.

Example 1.

$$\begin{tikzcd} & s_1 \arrow[dl, "l"'] \arrow[dr, "l"] & & t_1 \arrow[d, "k"] \\ s_2 \arrow[d, "h"] & & s_3 & t_2 \arrow[d, "j"] \\ s_4 & & & t_3 \end{tikzcd}$$

Consider the two LTL depicted above, where all the states have the same observation. (We could also imagine that states with no outgoing transitions have one transition towards a sink state with a special observation). In this case the relation $\{(s_1, t_1), (s_2, t_2), (s_4, t_3)\}$ is an AS. However, there is no AS containing (t_1, s_1). Indeed the only choice to match $t_1 \xrightarrow{k}$ is $s_1 \xrightarrow{l}$. But if now we choose $s_1 \xrightarrow{l} s_3$, we need to have (t_2, s_3), in which case there is no transition in the first system to match the action of the second system.

As one of the main results of this paper, we will show that, alternating simulation is a sound proof technique for the enforcing preorder.

Theorem 1. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ and $\mathcal{L}' = (S', s_0', L', \mathcal{T}', P, \leq, O')$ be two LTSs sharing the same observation order. If $\mathcal{L} \subset_{\mathtt{AS}} \mathcal{L}'$ then $\mathcal{L} \leq \mathcal{L}'$.*

This theorem implies Lemma 1 of [3]. Its proof is based on the simulation game, briefly suggested [3], that we formalize here for our variant of the alternating simulation.

3.1 The simulation game

In this section we adapt the classical two-player simulation game between Spoiler and Duplicator to the case of the alternating simulation. We then show that any memoryless winning strategy for Duplicator in this game defines an AS, and vice-versa, any AS gives a memoryless winning strategy for Duplicator.

The simulation game is built from any two LTS $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ and $\mathcal{L}' = (S', s_0', L', \mathcal{T}', P, \leq, O')$. Intuitively, from game positions labelled with pairs of states $(s, s') \in S \times S'$, Spoiler chooses a label $l \in L$ and the game advances to a position labelled (s, s', l) which belongs to Duplicator. Here, Duplicator must reply with a label $l' \in L'$ and then the game proceeds to a position labeled

(s, s', l, l') belonging again to Spoiler. In this new position, Spoiler chooses $t' \in S'$ such that $(s', l', t') \in \mathcal{T}'$, the game advancing further to a position (s, s', l, l', t') belonging to Duplicator. Finally, Duplicator must reply in this position with a state $t \in S$ such that $(s, l, t) \in \mathcal{T}$, after which the game advances to position (t, t'), and the above sequence of moves can be played again. All positions (s, s') with $O(s) \not\leq O(s')$ are winning for Spoiler, hence Duplicator's objective is to avoid these positions – that is, a safety objective.

Formally, the two-player turn-based simulation game is built as follows: $\mathcal{G} = (\mathcal{Q}_D, \mathcal{Q}_S, q_0, \delta)$ where $\mathcal{Q}_D = S \times S' \times L \cup S \times S' \times L \times L' \times S'$, $\mathcal{Q}_S = S \times S' \cup S \times S' \times L \times L'$, $q_0 = (s_0, s'_0) \in \mathcal{Q}_S$ $Act_D = L' \cup S$, $Act_S = L \cup S'$, and the transition function is:

$$
\begin{aligned}
\delta = & \big\{(s, s') \xrightarrow{l} (s, s', l) \mid s \in S, s' \in S', l \in L, s \xrightarrow{l} \big\} \\
& \cup \big\{(s, s', l) \xrightarrow{l'} (s, s', l, l') \mid s \in S, s' \in S', l \in L, l' \in L'\big\} \\
& \cup \big\{(s, s', l, l') \xrightarrow{t'} (s, s', l, l', t') \mid s \in S, s' \in S', l \in L, l' \in L', t' \in S' \\
& \qquad\qquad\qquad \text{with } (s', l', t') \in T'\big\} \\
& \cup \big\{(s, s', l, l', t') \xrightarrow{t} (t, t') \mid s \in S, s' \in S', l \in L, l' \in L', t \in S, t' \in S' \\
& \qquad\qquad\qquad \text{with } (s, l, t) \in T \text{ and } (s', l', t') \in T'\big\}
\end{aligned}
$$

Finally, Duplicator's objective is defined by the set of states $Obj = \{(s, s', \alpha) \in \mathcal{Q}_D \cup \mathcal{Q}_S \mid O(s) \leq O(s')\} \subseteq \mathcal{Q}$.

A strategy for Duplicator is a mapping $\sigma_D : (\mathcal{Q}_S \cdot \mathcal{Q}_D)^* \longrightarrow Act_D$ and a strategy for Spoiler is a mapping $\sigma_S : (\mathcal{Q}_S \cdot \mathcal{Q}_D)^* \times \mathcal{Q}_S \longrightarrow Act_S$. Furthermore, a memoryless strategy for Duplicator is a mapping $\sigma : \mathcal{Q}_D \longrightarrow Act_D$. Due to the particular way in which the states, actions and transitions are constructed, we will identify a memoryless strategy with a pair $(\sigma_{L'}, \sigma_S)$ with $\sigma_{L'} : S \times S' \times L \longrightarrow L'$ and $\sigma_S : S \times S' \times L \times L' \times S' \longrightarrow S$. A run ρ is compatible with a strategy for Duplicator σ if, whenever $\rho[i] \in \mathcal{Q}_D$, then $\rho[i] \xrightarrow{\sigma(\rho[\leq i+1])} \rho[i+2]$.

Note that the AS game $\mathcal{G}$ is a *safety* game, defined by the set of runs $Runs_{Obj} = \big\{\rho \in runs(\mathcal{G}) \mid \forall i \in \mathbb{N}.\rho[i] \in Obj\big\}$.

Theorem 2. *$\mathcal{L} \subset_{\mathtt{AS}} \mathcal{L}'$ if and only if Duplicator has a memoryless winning strategy in the simulation game.*

Proof. The proof proceeds by showing that any memoryless winning strategy for Duplicator gives rise to an alternating simulation, and vice-versa. Technically, this requires restating the notion of AS by skolemizing the existential quantifiers in Definition 4. The skolemized version of AS is given by the following proposition:

Proposition 1. *A relation $\mathcal{R}$ is an AS if and only if, for any $s\mathcal{R}s'$, $O(s) \leq O(s')$ and there exist partial functions $\eta : S \times S' \times L \longrightarrow L'$ and $\theta : S \times S' \times L \times S'$ (called an* AS pair*) such that whenever $s\mathcal{R}s'$:*

1. *For each $l \in L$ with $s \xrightarrow{l}$ we have that $\eta(s, s', l)$ is defined and $s' \xrightarrow{\eta(s,s',l)}$.*

2. For each $t' \in S'$ with $s' \xrightarrow{\eta(s,s',l)} t'$, we have that $\theta(s,s',l,t')$ is defined.
3. $s \xrightarrow{l} \theta(s,s',l,t') \in T$.
4. $\theta(s,s',l,t')\mathcal{R}t'$.

With this preparation, given $\sigma = (\sigma_{L'}, \sigma_S)$ a memoryless winning strategy for Duplicator, we build the following relation $\mathcal{R} \subseteq S \times S'$:

$s\mathcal{R}_\sigma s'$ iff there exists a run $\rho \in Runs(\mathcal{G})$ which is compatible with $(\sigma_{L'}, \sigma_S)$ such that $(s,s') = \rho[i]$ for some $i \in \mathbb{N}$.

We will show that $\mathcal{R}_\sigma$ is an AS between $\mathcal{L}$ and $\mathcal{L}'$.

Note first that, if $s\mathcal{R}_\sigma s'$ then $O(s) \leq O(s')$ since any run ρ which is compatible with $(\sigma_{L'}, \sigma_S)$ must be winning for Duplicator and therefore visit only positions $(s,s') \in Obj$. We then build, using σ, an AS pair $(\eta_\sigma, \theta_\sigma)$ as required by Proposition 1, as follows: for each $s \in S, s', t' \in S', l \in L$,

$$\eta_\sigma(s,s',l) = \sigma_{L'}(s,s',l) \text{ and } \theta_\sigma(s,s',l,t') = \sigma_S(s,s',l,\sigma_{L'}(s,s',l),t')$$

Then the pair $(\eta_\sigma, \theta_\sigma)$ satisfies the hypotheses of Proposition 1 for $\mathcal{R}_\sigma$:

1. For any (s,s',l), $\eta_\sigma(s,s',l) = \sigma_{L'}(s,s',l)$ is defined and $s' \xrightarrow{\sigma_{L'}(s,s',l)}$.
2. For any (s,s',l,t'), $\theta_\sigma(s,s',l,t') = \sigma_S(s,s',l,\sigma_{L'}(s,s',l),t')$ is defined.
3. $(s,s',l,\sigma_{L'}(s,s',l),t') \xrightarrow{\sigma_S(s,s',l,\sigma(s,s',l),t')} (\sigma_S(s,s',l,\sigma(s,s',l),t'),t') \in \delta$, which implies that $s \xrightarrow{l} \sigma_S(s,s',l,\sigma_{L'}(s,s',l),t') \in T$ by definition of $\mathcal{G}$.
4. Any run which reaches (s,s') and is compatible with $(\sigma_{L'}, \sigma_S)$ can be extended to a run which reaches $\sigma_S(s,s',l,\sigma_{L'}(s,s',l),t'),t')$, and therefore $\sigma_S(s,s',l,\sigma_{L'}(s,s',l),t'),t')\mathcal{R}_\sigma t'$.

For the other direction of Theorem 2, the skolemized version of AS will be again of help, by providing us with the Duplicator choices in each state of the simulation game. Namely, given AS $\mathcal{R}$ defined by the AS pair (η, θ) as in Proposition 1, we show that any extension of (η, θ) to a pair of total functions $(\sigma_{L'}, \sigma_S)$ represents a memoryless winning strategy for Duplicator in $\mathcal{G}$. Or, in other words, the tuples where η and θ are undefined cannot be reached by runs which are compatible with these choices.

Formally, take any strategy for Duplicator $(\sigma_{L'}, \sigma_S)$ with $\sigma_{L'} : S \times S' \times L \longrightarrow L'$ and $\sigma_S : S \times S' \times L \times L' \times S' \longrightarrow S$ which is defined as follows:

– For each s, s', l, $\sigma_{L'}(s,s',l) = \eta(s,s',l)$ if $\eta(s,s',l)$ is defined, and is arbitrary otherwise.
– For each s, s', l, l', t', $\sigma_S(s,s',l,\sigma(s,s',l),t') = \theta(s,s',l,t')$ if $\eta(s,s',l)$ and $\theta(s,s',l,t')$ are defined, and is arbitrary otherwise.

Then any finite run ρ which is compatible with $(\sigma_{L'}, \sigma_S)$ visits only states (s,s',α) with $s\mathcal{R}s'$ – and, as a consequence, $(s,s',\alpha) \in Obj$. The proof goes by induction on the length of the run.

The base case is trivial since $s_0\mathcal{R}s'_0$ and the initial position in $\mathcal{G}$ is compatible with any strategy. So assume ρ is a run of length ≥ 1. If the length of the run is

$4k+1$, then $\rho = \rho' \cdot (s, s', l)$ for some ρ' with $\ell(\rho') = 4k$ and $\rho'[4k] = (s, s')$, and therefore $s\mathcal{R}s'$ by the induction hypothesis. Furthermore, for $\ell(\rho) = 4k + 2$ we must have $\rho = \rho' \cdot (s, s', l, l')$ and, by the induction hypothesis, $\rho'[4k + 1] = (s, s', l)$ is such that $s\mathcal{R}s'$. But then, by construction of $\sigma_{L'}$, we must have $l' = \sigma_{L'}(s, s', l) = \eta(s, s', l)$. Going one step further, for $\ell(\rho) = 4k + 3$ we must have $\rho = \rho' \cdot (s, s', l, l', t')$ and again $s\mathcal{R}s'$ by the induction hypothesis and $l' = \sigma_{L'}(s, s', l)$. Finally, for $\ell(\rho) = 4k + 4$ and $\rho = \rho' \cdot (t, t')$ compatible with $(\sigma_{L'}, \sigma_S)$, we must have $\rho[4k + 3] = (s, s', l, l', t')$, $s\mathcal{R}s'$, $l' = \sigma_{L'}(s, s', l)$, and $t = \sigma_S(s, s', l, l', t') = \theta(s, s', l, t')$ and hence $t\mathcal{R}t'$.

4 Strategies as contexts

The previous section does not tell the whole story of alternating simulation, and we explore here the connections with observational preorders from [8].

4.1 Observational preorders

In a syntactic calculus, there is a standard way to define observational preorder on syntactic terms, which we call here the *Morris-style* definition: $t \leq s$ if for every context C, the observations that can be made on $C[t]$ are (in some sense) included in the observations that can be made on $C[s]$ [8]. In the case of the functional language PCF, for instance, the Morris preorder is defined taking termination as the only observation.

While the relation is very easy to define, and very convincing, the quantification over all possible contexts makes it hard to directly prove that two terms are in the relation. Some other, easier to handle, notion is then introduced for this purpose. For instance, in the '70 people tried to capture the observation pre-congruence for PCF using domains, and subsequently using game semantics. The holy Grail of this line of research was "full abstraction", a precise characterisation of the Morris preorder.

In the study of CCS, (bi)-simulation and its large weaponry of up-to techniques, was proven to precise characterise barbed pre-congruence, which can be argued to be a generalisation of the Morris preorder to nondeterministic systems.

In the exemples mentioned above, contexts can be seen as way of testing a term: you submit a term to different experiments, and observe the results. In the setting studied here, there are no terms, only transition systems. The only way to interact with a transition system is by playing on it. Therefore we argue that the right transposition of contexts, here, are the strategies.

Let's try to formulate the Morris-style preorder using this intuition. Given two LTS $\mathcal{L}, \mathcal{L}'$ we say that $\mathcal{L} \leq \mathcal{L}'$ if for any strategies σ_A, σ_O the observations of $\rho[\mathcal{L}, \sigma_A, \sigma_O]$ are included in the observations of $\rho[\mathcal{L}', \sigma_A, \sigma_O]$.

There is a problem with this naive formulation: the definition of strategy does not allow the same strategies to interact with different transition systems. We need to have a way to generalise the notion of "same" strategy. Strategies make choices based on the previous history. We need to put in correspondence different choices, on different histories. We argue then that two strategies are "the same" if they make corresponding choices.

4.2 Choice correspondence and the Morris preorder

We have therefore to propose a suitable definition of "choice correspondence" for states and labels:

Definition 5. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T}, P, \leq, O)$ and $\mathcal{L}' = (S', s_0', L', \mathcal{T}', P, \leq, O')$ be two LTSs sharing the same observation order. A* choice correspondence *is consituted by two mappings:*
$$\begin{cases} f : runs_\bullet(\mathcal{L}) \times runs_\bullet(\mathcal{L}') \to L \to L' \\ g : runs_\to(\mathcal{L}) \times runs_\to(\mathcal{L}') \to S' \to S \end{cases}$$
with the following properties:

- *if $hl \in runs_\to(\mathcal{L})$ and $f(h, h')(l) = l'$ then $h'l' \in runs_\to(\mathcal{L}')$;*
- *if $h's' \in runs_\bullet(\mathcal{L}')$ and $g(h, h')(s') = s$ then $hs \in runs_\bullet(\mathcal{L})$.*

In one direction, the f component builds a correspondence between choices of labels, while in the other direction, the g component builds a correspondence between states. Both mappings take into account the history of the computation.

A choice correspondence allows us to build a run on two LTSs, with just one pair of strategies. Since the corresponding functions act in different directions, we will need one strategy to be defined on each LTS. Then the two other strategies are induced by the correspondence.

Let $\mathcal{L} = (S, s_0, L, \mathcal{T})$ and $\mathcal{L}' = (S', s_0', L', \mathcal{T})$ be two LTSs. Consider (f, g) be a choice correspondence, σ_A a strategy for agent in $\mathcal{L}$ and σ_O' a strategy for opponent in $\mathcal{L}'$.

We define the following:

- a map $\xi' : runs_\to(\mathcal{L}) \to runs_\to(\mathcal{L}')$,
- a strategy σ_O for opponent in $\mathcal{L}$,
- a map $\xi : runs_\bullet(\mathcal{L}') \to runs_\bullet(\mathcal{L})$,
- and a strategy σ_A' for agent in $\mathcal{L}'$.

We do that by induction on the length of the argument. For the base case we put $\xi(s_0') = s_0$ and $\sigma_A'(s_0') = \sigma_A(s_0)$.

For the induction step, let $h'l's' \in runs_\bullet(\mathcal{L}')$, define $h = \xi(h')$, $l = \sigma_A(h)$ and $s = g(hl, h'l')(s')$, then

$$\xi(h'l's') = hls \text{ and } \sigma_A'(h'l's') = f(hls, h'l's')(\sigma_A(hls)).$$

The function ξ' and the strategy σ_O are defined analogously.

Note that the definition of σ_A' does not depend on σ_O' and that the definition σ_O does not depend on σ_A. When f, g is clear from the context, we will denote $\sigma_A' = \xi'(\sigma_A)$ and $\sigma_O = \xi(\sigma_O')$. We obtain thus two runs: $\rho[\mathcal{L}, \sigma_A, \xi(\sigma_O')]$ and $\rho[\mathcal{L}', \xi'(\sigma_A), \sigma_O']$.

We can now say that σ_A and $\xi'(\sigma_A)$ are “the same” strategy up to the choice correspondence (f, g) (and similarly for $\xi(\sigma'_O)$ and σ'_O).

The definitions proposed above allow us to give a generalised definition of contextual preorder. Recall that in the in the original definition, $S \le S'$ if, whatever observations we can make with $C[S]$, it is possible with $C[S']$. Here we don't really care for which term we choose the context as it must be the the same for both. The way we defined the choice correspondence forces us to define the strategies on specific systems, and then “transfer it” to the other one. This informal discussion leads us to this formal definition of Morris preorder up to choice correspondence:

Definition 6. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T})$ and $\mathcal{L}' = (S', s'_0, L', \mathcal{T})$ be two LTSs. Consider a choice correspondence (f, g) for $\mathcal{L}$ and $\mathcal{L}'$. We say that $\mathcal{L} \le_{f,g} \mathcal{L}'$ if for all strategy σ_A for agent in $\mathcal{L}$ and all strategy σ'_O for opponent in $\mathcal{L}'$:*

$$O(\rho[\mathcal{L}, \sigma_A, \xi(\sigma'_O)]) \le O(\rho[\mathcal{L}', \xi'(\sigma_A), \sigma'_O]).$$

5 The adequacy theorems

We can now state the main theorem of the paper.

Theorem 3. *. Let $\mathcal{L} = (S, s_0, L, \mathcal{T})$ and $\mathcal{L}' = (S', s'_0, L', \mathcal{T})$ be two LTSs. The following are equivalent:*

1. *$\mathcal{L} \subset_{\texttt{AS}} \mathcal{L}'$;*
2. *there exists a choice correspondence (f, g) such that $\mathcal{L} \le_{f,g} \mathcal{L}'$.*

To prove this theorem we utilize the AS game defined in Subsection 3.1.

Lemma 1. *Duplicator has a winning strategy in $\mathcal{G}$ if and only if there is a choice correspondence (f, g) for which $\mathcal{L} \le_{f,g} \mathcal{L}'$.*

For the one direction, we build a choice correspondence (f, g) directly from the definition of a strategy σ_D for Duplicator. We then show that if σ_D is winning, $\mathcal{L} \le_{f,g} \mathcal{L}'$. For the other direction, from a choice correspondence (f, g) we build a strategy σ_D for Duplicator, which is winning if $\mathcal{L} \le_{f,g} \mathcal{L}'$. See the appendix for the details.

In general, the strategy we build for Duplicator is aware of all the history, but Theorem 2 requires a memoryless strategy. Therefore, to conclude the Proof of Theorem 3, we need to observe that, $\mathcal{G}$ being a safety game and hence a particular type of parity game, it is memoryless determined [6, 10], that is, if Duplicator has a winning strategy, then she has a memoryless winning strategy.

We are finally able to prove Theorem 1. It is a corollary of the following proposition and of Theorem 3.

Proposition 2. *Let $\mathcal{L} = (S, s_0, L, \mathcal{T})$ and $\mathcal{L}' = (S', s'_0, L', \mathcal{T})$. If there exists a choice correspondence (f, g) for which $\mathcal{L} \le_{f,g} \mathcal{L}'$, then $\mathcal{L} \le \mathcal{L}'$.*

Proof. Consider a choice correspondence (f, g). Given two strategies σ_A for $\mathcal{L}$ and σ'_O for $\mathcal{L}'$, we are able to build two strategies σ'_A for $\mathcal{L}'$ and σ_O for $\mathcal{L}$. If $\mathcal{L} \leq_{f,g} \mathcal{L}'$ then
$$O(\rho[\mathcal{L}, \sigma_A, \sigma'_O]) \leq O(\rho[\mathcal{L}', \sigma_A, \sigma'_O]).$$

Summarizing, the following formula is true:

$$\forall \sigma_A \forall \sigma'_O \exists \sigma'_A \exists \sigma_O . O(\rho[\mathcal{L}, \sigma_A, \sigma_O]) \leq O(\rho[\mathcal{L}', \sigma'_A, \sigma'_O]).$$

Note, however, that the way we constructed σ'_A depends only on the choice correspondence (f, g) and on the definition of σ_A. We can therefore swap the quantifiers:

$$\forall \sigma_A \exists \sigma'_A \forall \sigma'_O \exists \sigma_O . O(\rho[\mathcal{L}, \sigma_A, \sigma_O]) \leq O(\rho[\mathcal{L}', \sigma'_A, \sigma'_O]).$$

In a sense, the choice correspondence acts as a form of *local skolemization* of the existential quantifiers.

Fix now an upward-closed specification φ and assume that $O(\rho[\mathcal{L}, \sigma_A, \sigma_O]) \leq O(\rho[\mathcal{L}', \sigma'_A, \sigma'_O])$. Then $\rho[\mathcal{L}, \sigma_A, \sigma_O] \vDash \varphi \implies \rho[\mathcal{L}', \sigma'_A, \sigma'_O] \vDash \varphi$.

Therefore if there exists a choice correspondence (f, g) for which $\mathcal{L} \leq_{f,g} \mathcal{L}'$, we can conclude that for all upward-closed specification φ:

$$\forall \sigma_A \exists \sigma'_A \forall \sigma'_O \exists \sigma_O . \left(\rho[\mathcal{L}, \sigma_A, \sigma_O] \vDash \varphi \implies \rho[\mathcal{L}', \sigma'_A, \sigma'_O] \vDash \varphi\right).$$

By pushing the quantifiers inward in a suitable way, we obtain that for all upward-closed specification φ:

$$(\exists \sigma_A \forall \sigma_O . \rho[\mathcal{L}, \sigma_A, \sigma_O] \vDash \varphi) \implies (\exists \sigma'_A \forall \sigma'_O . \rho[\mathcal{L}', \sigma'_A, \sigma'_O] \vDash \varphi)$$

which is the definition of enforcing preorder.

Proposition 3. *The inverse of Theorem 1 does not hold.*

Figure 2, inspired from [14], provides the counterexample for the inverse of Theorem 1. The partial order of observations is $P = \{\perp, p, q, r\}$ where $\leq$ is the identity relation. In both LTS, the Agent has two strategies, one enforcing $\varphi_1 = \{\perp p^\omega, \perp q^\omega\}$ and the other enforcing $\varphi_2 = \{\perp p^\omega, \perp r^\omega\}$. Hence $\mathcal{L}_1 \leq \mathcal{L}_2$ and $\mathcal{L}_2 \leq \mathcal{L}_1$. On the other hand, clearly $\mathcal{L}_1 \not\leq_{\mathtt{AS}} \mathcal{L}_2$ and $\mathcal{L}_2 \not\leq_{\mathtt{AS}} \mathcal{L}_1$.

6 Complements

6.1 The quest for symmetric relations

We have studied in detail three preorders. What can we say about the symmetric version of them? The symmetric version of the enforcing preorder, that we call *enforcing equivalence*, is easily defined: We say that $\mathcal{L}$ is enforcing equivalent to $\mathcal{L}'$ if for any specification φ, Agent can enforce φ on $\mathcal{L}$ if and only if Agent can enforce φ on $\mathcal{L}'$.

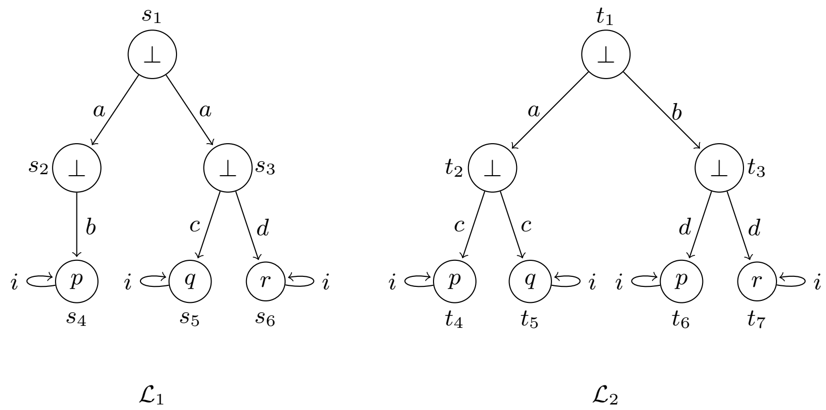
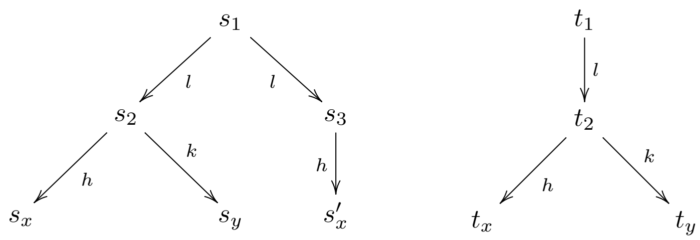


Fig. 2. Two LTS which are enforcing equivalent but for which there exists no alternating simulation in either direction.

The notion of alternating bisimulation is also easily defined as a symmetric alternating simulation. The largest alternating bisimulation is called alternating bisimilarity. As for Park and Milner bisimulation, alternating bisimilarity is stronger than the the equivalence generated by alternating similarity, as we show in the following example.

Example 2.



In this model, the observation in states s_x, s'_x, t_x is x, and the observation in s_y, t_y is y. The reader can verify that $\{(s_1, t_1), (s_2, t_2), (s_x, t_x), (s_y, t_y)\}$ and $\{(t_1, s_1), (t_2, s_2), (t_2, s_3), (t_x, s_x), (t_y, s_y), (t_x, s'_x)\}$ are both alternating simulations, but there is no alternating bisimulation containing (s_1, t_1).

It remains open to find a proper definition of Morris equivalence, as we have not yet pinned down the right symmetric generalisation of the notion of choice correspondence. Asking the functions (f, g) to be bijective seems to us too strong a requirement. However just asking the existence of two unrelated choice correspondences would correspond to having two simulations in both direction, and we have just shown that this is weaker than bisimilarity.

This quest for Morris equivalence should also be guided by the bisimulation game, which is the symmetric version of the simulation game in Section 3.2, and then stating an appropriate adaptation of Theorem 2 and, consequently, Theorem 3. More specifically, in the bisimulation game, in each position $(s, s') \in$

$S \times S'$, Spoiler first chooses the side where she challenges the simulation (that is, challenges Duplicator with either proving that $s \subset_{\mathtt{AS}} s'$ or $s' \subset_{\mathtt{AS}} s$), and then proceeds by proposing Duplicator with a label in the chosen transition system. We call this extra intermediary step a *side-challenging step*. The symmetry in the definition of the alternating bisimulation could be solved in the bisimulation game by requiring that Duplicator have *imperfect information*, in the sense that she "forgets" each Spoiler's side-challenging step. But two-player games with imperfect information are not determined in general, hence more study is needed to properly adapt Theorem 2.

6.2 Taking labels seriously

To get closer to the world of Park and Milner we propose a definition of simulation that takes into account the identity of the labels.

Definition 7. *A* Name-aware alternating simulation *(NAAS) on a labelled transition system, is a binary relation $\mathcal{R}$ such that whenever $s \mathcal{R} t$, $O(s) \leq O(t)$ and:*

> *for all labels l, if $s \xrightarrow{l}$, then $t \xrightarrow{l}$, and for all t' s.t. $t \xrightarrow{l} t'$, there exists s' s.t. $s \xrightarrow{l} s'$ and $s' \mathcal{R} t'$.*

The largest NAAS, $\subset_{\mathtt{NA}}$, is called name-aware similarity*. If there is a NAAS $s \mathcal{R} t$, we say $s \subset_{\mathtt{NA}} t$.*

In the following examples we suppose all the states have the same observation.

Example 3.

$$\begin{tikzcd}
& s_1 \arrow[dl, "l"'] \arrow[dr, "l"] & & & t_1 \arrow[dl, "l"'] \arrow[dr, "l"] & \\
s_2 \arrow[d, "h"] & & s_3 & t_2 \arrow[dr, "h"] & & t_4 \arrow[dl, "h"] \\
s_4 & & & & t_5 &
\end{tikzcd}$$

$\{(s_1, t_1), (s_2, t_4), (s_4, t_5)\}$ is a NAAS

Example 4.

$$\begin{tikzcd}
& s_1 \arrow[dl, "l"'] \arrow[d, "l"] \arrow[dr, "k"] & & & t_1 \arrow[dl, "l"'] \arrow[dr, "l"] & \\
s_2 \arrow[d, "h"] & s_3 & s_5 & t_2 \arrow[dr, "h"] & & t_4 \arrow[dl, "h"] \\
s_4 & & & & t_5 &
\end{tikzcd}$$

There is no NAAS in either direction.

The example above shows that the NAAS is different from the standard notion of similarity by Park and Milner. Indeed there is a standard simulation between t_1 and s_1: the fact that there is a label more from s_1 is irrelevant.

While the two notions of simulation differ, it can be shown that the symmetric notions coincide.

Definition 8. *A* Name-aware alternating bisimulation *(NAAB) on a labelled transition system, is a binary relation $\mathcal{R}$ such that both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are NAAS. The largest NAAB is called* name-aware bisimilarity*.*

A Park and Milner simulation *(PMS) on a labelled transition system, is a binary relation $\mathcal{R}$ such that whenever $s \mathcal{R} t$,*

- $O(s) \leq O(t)$
- *for all labels l, and for all states s' if $s \xrightarrow{l} s'$, then there exists t' such that $t \xrightarrow{l} t'$., there exists s' s.t. $s' \mathcal{R} t'$.*

A Park and Milner bisimulation *(PMB) on a labelled transition system, is a binary relation $\mathcal{R}$ such that both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are PMS.*

Theorem 4. *A binary relation $\mathcal{R} \subseteq S \times T$ is a Name-aware alternating bisimulation if and only if it is a Park and Milner bisimulation.*

Proof. The proof is done in both directions by checking that a NAAB satisfies the conditions for being a PMB, and that a PMB satisfies the conditions for being a NAAB.

7 Conclusions and future work

We have generalised the syntactic notion of observational preorder to a setting without syntax, and we also have presented some notions originally defined on alternating transition systems, using standard labelled transition systems. This leads us to a new definition of a coinductive relation, that happens to characterise the Morris preorder.

Alternating bisimulations were used to prove (manually) bisimulation reductions for multi-agent systems [5], which were specified using the ISPL multi-agent modelling language used in the MCMAS tool for model-checking. We plan to provide LTS semantics to such multi-agent modelling languages together with algorithmic tools for deciding or building alternating bisimulation reductions. This will lead us to an extension of this work to the case of alternating transition systems (or concurrent game structures) with imperfect information, which requires a notion of observation-based strategy.

A notion of choice correspondence that takes into account the identity of the labels can be easily defined. We think that the corresponding notion of Morris preorder coincides with name-aware similarity.

Acknowledgments

We thank the anonymous reviewers for their remarks, suggestions and references, among which the papers [14, 15] provided us with the inspiration for the counterexample in Proposition 3.

References

1. Rajeev Alur and Thomas A. Henzinger. Reactive modules. *Formal Methods Syst. Des.*, 15(1):7–48, 1999.
2. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
3. Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In Davide Sangiorgi and Robert de Simone, editors, *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, volume 1466 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 1998.
4. Katie Atkinson and Trevor J. M. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artif. Intell.*, 171(10-15):855–874, 2007.
5. Francesco Belardinelli, Rodica Condurache, Catalin Dima, Wojciech Jamroga, and Andrew V. Jones. Bisimulations for verifying strategic abilities with an application to threeballot. In Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee, editors, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1286–1295. ACM, 2017.
6. E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 368–377. IEEE Computer Society, 1991.
7. Julian Gutierrez, Paul Harrenstein, Giuseppe Perelli, and Michael J. Wooldridge. Nash equilibrium and bisimulation invariance. *Log. Methods Comput. Sci.*, 15(3), 2019.
8. Jr. James Hiram Morris. *Lambda-Calculus Models of Programming Languages.* PhD thesis, M.I.T., 1968.
9. Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. MCMAS: an open-source model checker for the verification of multi-agent systems. *Int. J. Softw. Tools Technol. Transf.*, 19(1):9–30, 2017.
10. René Mazala. Infinite games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2001.
11. Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science.* Springer, 1980.
12. Robin Milner. *Communicating and mobile systems - the Pi-calculus.* Cambridge University Press, 1999.
13. Mogens Nielsen and Glynn Winskel. *Models for Concurrency*, pages 1–148. Oxford University Press, 1995. Also published in BRICS Research Series as RS-94-12.
14. Johan van Benthem. Extensive games as process models. *J. Log. Lang. Inf.*, 11(3):289–313, 2002.
15. Johan van Benthem, Nick Bezhanishvili, and Sebastian Enqvist. A new game equivalence, its logic and algebra. *J. Philos. Log.*, 48(4):649–684, 2019.
16. Wiebe van der Hoek, Mark Roberts, and Michael J. Wooldridge. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synth.*, 156(1):1–19, 2007.