Refinement for open automata

Quentin Corradi

March 26, 2021

1 Introduction

Open automata are used to give an interpretation for open pNets.

2 Notations

Throughout this paper, tuples might be noted differently depending on what they represent. Quantification will be used loosely, any bound name in a quantification means that the object is constrained to have that previous value, and any new name is bound by the quantifier. For instance $\forall(x, y), \exists(x', y), P$ means $\forall(x, y), \exists(x', y'), y = y' \wedge P$.

Family of values, or equivalently maps will be noted $\{i \mapsto x_i \mid i \in I\}$, $\{i \leftarrow x_i \mid i \in I\}$ or $x_i^{i \in I}$. The latter will only be used when unambiguous; for instance $(ax)^{x \in \mathbb{R}}$ represents a scaling function, $c^{i \in I}$ is a constant function over I, but $\{\alpha \mapsto 1, \beta \mapsto 2, \gamma \mapsto 3\}$ must be represented with one of the two first notation. The disjoint union of two maps $\varphi : I \to X$ and $\psi : J \to Y$ with $I \cap J = \emptyset$ is $\varphi \uplus \psi : (I \uplus J) \to (X \cup Y)$.

3 Open Automaton

To define an open automaton we need some preliminary definitions.

Definition 1 (Expression algebra, Formulae, Terms)**.** An expression algebra E is a disjoint union $E \coloneqq \mathcal{T} \uplus \mathcal{F}$ where $\mathcal{T}$ is a term algebra and $\mathcal{F}$ the formulae over $\mathcal{T}$.

The terms algebra is arbitrary. The formulae form a first order logic over $\mathcal{T}$ and some relations. An example of term algebra can be Peano integers (zero, variable and successor function), the formulae associated can use equality relation, $\mathit{sum}(a, b, c) \coloneqq a = b + c$ relation and $\mathit{prod}(a, b, c) \coloneqq a = b \times c$ relation.

Definition 2 (Free variables, Closed terms, Closed formulae)**.** $\mathit{vars}(e)$ is the set of unbound variables in $e \in E$. The terms restricted to variables in V are $\mathcal{T}_V ::= \{t \in \mathcal{T} \mid \mathit{vars}(t) = \emptyset\}$; $\mathcal{T}_V \subset \mathcal{T}$. The formulae restricted to variables in V are $\mathcal{F}_V ::= \{f \in \mathcal{F} \mid \mathit{vars}(f) = \emptyset\}$; $\mathcal{F}_V \subset \mathcal{T}$. The closed terms are terms restricted to variables in $\emptyset$, $\mathcal{T}_\emptyset$. The closed formulae are formulae restricted to variables in $\emptyset$, $\mathcal{F}_\emptyset$.

By extension, expressions restricted to variables and closed expressions are also defined. A closed expression is an expression without unbound variable.

Definition 3 (Interpretation, Values, Satisfiability, (Parallel) substitution)**.** We assume that the following functions are given:

- The interpretations of closed terms, $[\![\_]\!] : \mathcal{T}_\emptyset \to \mathcal{P}$, with $\mathcal{P}$ a set of interpreted value of terms;
- The satisfiability relation on closed formulae, $\vdash \subseteq \mathcal{F}_\emptyset$ and unsatisfiability relation $\nvdash \subseteq \mathcal{F}_\emptyset$;
- The substitution in $e \in E$ of $x \in \mathit{vars}(e)$ by $t \in \mathcal{T}$, $e[t/x]$;
- The parallel substitution in $e \in E$ of variables in V by $\psi : V \to \mathcal{T}$, $e\{\!\{\psi\}\!\}$.

For the parallel substitution, the set V is not required to be a subset of $\mathit{vars}(e)$. In the case where it is not, the variables in $V \setminus \mathit{vars}(e)$ are not substituted. The substitutions might give a non-sensical expression; for instance terms can be integers and pairs with addition, $(a+b)\{\!\{a \mapsto 7, b \mapsto (4,5)\}\!\}$ is not a valid term. This can be guarded with the atom $e[t/x] \in E$ and $e\{\!\{\psi\}\!\} \in E$ and it will implicitely be the case for instance when there is quantification on t and ψ to simplify notations.

The interpretation of values is supposed to be decidable. The (un)satisfiability of formulae might not be decidable nor complete nor consistent, however we will pretend like they are because these are really hard problems for logicians that we don't want to deal with.[1]

A value v may be used for keeping a variable state, and then injected in terms for substitution. While this is correct when $\mathcal{P} \subseteq \mathcal{T}_\emptyset$, in the other cases this will be used as a shorthand for substitution with any $t \in [\![v]\!]^{-1}$.

We are not interested in how $\mathcal{P}$ is defined. Whether it is predefined or caracterised by $\mathcal{P} ::= [\![\mathcal{T}_\emptyset]\!]$ is not the point of this definition.

The satisfiability relations will be noted using the classical relational notation $\vdash f ::= f \in \vdash$ and $\nvdash f ::= f \in \nvdash$. The usual satisfiability under a partial valuation is noted $\sigma \vdash f ::= \exists \varphi : (\mathit{vars}(f) \setminus V) \to \mathcal{P}, \vdash f\{\!\{\sigma \uplus \varphi\}\!\}$ with $f \in \mathcal{F}, \sigma : V \to \mathcal{P}$.

From there the expression algebra is fixed. With these common definitions and notations settled, the objects of interest can now be defined.

Definition 4 (Open automaton)**.** An open automaton is a tuple $\langle S, i, V, \varphi, J, T \rangle$ with S the set of states, $i \in S$ the initial state, V the set of variable names unique to this automaton, $\varphi : V \to \mathcal{P}$ the initial valuation of variables, J the set of hole names and T a set of open-transitions.

S, V, J are arbitrary sets, only J is required to be finite.

The variable names may clash when considering two automata, in that case we suppose that we can still distinguish the variables in the formulas. All the possible open transitions, from which transitions in T are selected, are defined below.

Definition 5 (Open transition)**.** An open transition is a tuple $\dfrac{(\beta_j(v_{\beta j}))^{j \in J'}, g, \psi}{s \xrightarrow{\alpha(v_\alpha)} s'}$ with $s, s' \in S$ the source and target states, $\alpha \in A$ an action label, $J' \subseteq J$ the holes involved, $\beta_j \in A$ the action labels of the holes, $v_\alpha, v_{\beta j}^{j \in J'}$ fresh distinct variables, $g \in \mathcal{F}$ the guard and $\psi : V \to \mathcal{T}$ the new value of the variables.

[1]In practise a term algera with only equality is used, and it is has all these properties.

The set of action labels A is arbitrary and implicit. To simplify formulae, the notations $\alpha ::= \alpha(v_\alpha)$ and $\alpha = \beta ::= \alpha = \beta \wedge v_\alpha = v_\beta$ will be used. The guard and new values of the variables can use any defined variable: $\bigcup_{v \in V} \mathit{vars}(\psi(v)) \cup \mathit{vars}(g) \subseteq V \cup \{v_\alpha\} \cup \{v_{\beta j} \,|\, j \in J'\}$.

Now we can define some utilitary functions:

Definition 6 (Guard, Variables, Out-transition, In-transition)**.** $\mathrm{OT}(s)$ are called the out-transitions of s and $\mathrm{IT}(s)$ and its in-transitions.

$$\mathit{guard}\left(\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'}\right) ::= g \qquad \mathit{vars}\left(\frac{\left(\beta_j(v_{\beta j})\right)^{j\in J'}, g, \psi}{s \xrightarrow{\alpha(v_\alpha)} s'}\right) ::= \{v_\alpha\} \cup \{v_{\beta j} \,|\, j \in J'\}$$

$$\mathrm{OT}(s) ::= \left\{\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'} \in T\right\} \qquad \mathrm{IT}(s') ::= \left\{\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'} \in T\right\}$$

The intuition of an open automaton is a partially defined LTS with variables, guards on transitions and parametrised actions.

4 Semantic and composition of Open Automata

When the auomaton is in a state $s \in S$ with a valuation of its variables $\sigma \in V \to \mathcal{P}$, it cannot perform transitions $t := \frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'} \in T$ such that $\nvDash g\{\!\{\sigma\}\!\}$. After performing the transition and emitting action α, the automaton is in the state s' with valuation $\psi\{\!\{\sigma \uplus \varphi\}\!\}$ for some $\varphi : \mathit{vars}(t) \to \mathcal{P}$. The holes are synchronised using a handshake like mecanism when the transition happens.

Example 1.

An open automaton can be partially specified, the partial specification comes from the holes. A hole can be filled with an open automaton, this operation is called composition.

Definition 7 (Composition of open automata)**.** The composition of $\mathrm{OA}_c := \langle S_c, i_c, V_c, \varphi_c, J_c, T_c\rangle$ in the hole $k \in J_p$ of $\mathrm{OA}_p := \langle S_p, i_p, V_p, \varphi_p, J_p, T_p\rangle$ is:

$$\mathrm{OA}_p[\mathrm{OA}_c/k] ::= \langle S_p \times S_c, (i_p, i_c), V_p \uplus V_c, \varphi_p \uplus \varphi_c, J_c \uplus J_p \setminus \{k\}, T\rangle$$

$$\text{With } T := \left\{\frac{\beta_j^{j\in J'_c \uplus J'_p \setminus\{k\}}, g_p \wedge g_c \wedge \alpha_c = \beta_k, \psi_p \uplus \psi_c}{(s_p, s_c) \xrightarrow{\alpha_p} (s'_p, s'_c)} \;\middle|\; \frac{\beta_j^{j\in J'_p}, g_p, \psi_p}{s_p \xrightarrow{\alpha_p} s'_p} \in T_p, \frac{\beta_j^{j\in J'_c}, g_c, \psi_c}{s_c \xrightarrow{\alpha_c} s'_c} \in T_c\right\}$$

$$\cup \left\{\frac{\beta_j^{j\in J'_p}, g_p, \psi_p}{(s_p, s_c) \xrightarrow{\alpha_p} (s'_p, s_c)} \;\middle|\; \frac{\beta_j^{j\in J'_p}, g_p, \psi_p}{s_p \xrightarrow{\alpha_p} s'_p}, k \notin J', s_c \in S_c\right\}$$

Similarly to expression substitution, the parallel composition is noted $\mathrm{OA}\left\{\!\!\left\{\mathrm{OA}_j^{j\in J}\right\}\!\!\right\}$.

The actions emitted when OA_c makes a transition is sychronised with the action of the hole k in transitions of OA_p which have it as a hole actions (first transition set, $\alpha_c = \beta_k$). No transition of OA_c is performed when a transition that do not refer to the hole k is performed in OA_p (second transition set, $k \notin J'$). The composition may look like a handshake, with both automata running in parallel (product of states and joint variables) but it is actually not symmetric.

Example 2.

5 What is a refinement for Open Automata

There are several properties that we may want from a refinement relation. Depending on these properties, several kind of refinement may be used. For example if we are interested in being able to produce the same sequence of action we may want to use trace set inclusion as a refinement. Here the main properties that we want are related to composition and action refinement. A suitable kind of refinement relation for this kind of properties is simulation refinement: A bisimulation has already been proposed for open automata that is compatible with composition.

cite an article here

Other kind of refinement relation that I may explore are control refinement/hole refinement[2], (meet semi)lattice refinement[3], weak-simulation refinement[4], composition-correct refinement[5].

This is a note for possible path, some may be explored, some may not.

The important properties we will consider here are:

Reflexivity: $\forall a, a \leq a$;

Transitivity: $\forall a\, b\, c, a \leq b \wedge b \leq c \implies a \leq c$;

Preorder: Reflexivity and transitivity;

Composition completeness: $\forall a\, b, a[b] \leq a$;

Composition correctness: $\forall a\, b, a \leq b \implies \exists c, a =_{FH} b[c]$;

Composition context equivalence: $\forall a\, b\, c, a \leq b \implies a[c] \leq b[c]$;

Composition congruence: $\forall a\, b\, c, a \leq b \implies c[a] \leq c[b]$;

Composition compatibility: $\forall a\, b\, c\, d, a \leq b \wedge c \leq d \implies c[a] \leq d[b]$;

FH-bisimulation compatibility: $\forall a\, b\, c, d, a =_{FH} b \wedge c =_{FH} d \wedge a \leq c \implies b \leq d$;

Deadlock reduction: If $a \leq b$ then a does not introduces new deadlocks, noted $a \sqsubseteq b$ in the following;

Livelock reduction: If $a \leq b$ then a does not introduces new livelocks, noted $a \preceq b$ in the following;

τ-stuttering: .

"Deadlock reduction" conflicts with "composition completeness" by disallowing some unwanted cases where an automaton in a hole cannot produce any action that out-transitions expect. In particular filling a hole with the deadlock automaton (one state without out-transitions) is not considered a refinement for a no deadlock introductive refinement. "Composition completeness" can be modified into "Deadlock reduction completeness" to be compatible with "Deadlock reduction": $\forall a\, b, a[b] \sqsubseteq a \implies a[b] \leq a$. Only the latter will be considered. Similar issues arise on other properties and can be fixed by changing the prerequisites to insure that the most deadlock-prone composition of automata is composition complete:

[2]No good formulation atm, the idea is that $a \leq b ::= sth \wedge J_a \setminus J_b \leq_{ctrl} J_b \setminus J_a$, sth is probably a clause to ensure that they behave the same without holes involved.

[3]meet = handshake, $a \leq b ::= a = a||b$ with sync holes, I think this will be equivalent to hole-identical sim, (join=non-det choice?)

[4]Weak variation for each interesting simulation refinement

[5]Basically a stricter variant of simulation-refinement where simulation has to take place the other way for some transitions (no different holes?) in order to be correct wrt composition

Composition context equivalence modified: $\forall a\, b\, c, a \leq b \wedge a[c] \leq a \implies a[c] \leq b[c]$;

Composition congruence modified: $\forall a\, b\, c, a \leq b \wedge c[a] \leq c \implies c[a] \leq c[b]$;

Composition compatibility modified: $\forall a\, b\, c\, d, a \leq b \wedge c \leq d \wedge c[a] \leq c \implies c[a] \leq d[b]$;

On the other hand there is a property that was discussed, that may arise when designing a refinement relation and that is interesting but actually unwanted:

Daisy equivalence: $a[D] \leq a \wedge a \leq a[D]$, where D is the single state automaton that can produce any action.

This property state that composing a hole with an automaton which can do anything is not a strict refinement. While it might seem to be a natural property because this automaton is the closest to the meaning of a hole in term of automaton, there is actually a difference between the two. Daisy equivalence allows to refine by filling a hole but also by going the other way, that is creating a hole name and putting its hole action to any transition (and possibly keeping any of the unmodified transitions). The unwanted behaviour arise for instance when two independant holes are filled, then a hole is created and its actions are put where the two other actions were. This effectively merges independant holes, or equivalently allows to consider that an automaton can be plug simultaneously into several holes which is not part of the composition semantics.

Example 3.

The relations will have to be at least preorders and simulations to be called refinement simulation. However a simulation on open automata is more complex than a simulation on a LTS.

Definition 8 (Simulation refinement on open automata)**.** A relation $\leq$ is a simulation refinement if it is a preorder and let $\mathrm{OA}_1 := \langle S_1, i_1, J_1, V_1, \varphi_1, T_1 \rangle$ and $\mathrm{OA}_2 := \langle S_2, i_2, J_2, V_2, \varphi_2, T_2 \rangle$, the following holds:

$$\begin{aligned} \mathrm{OA}_1 \leq \mathrm{OA}_2 \implies \exists R : (S_1 \times S_2) \to \mathcal{F}_{V_1 \uplus V_2}, \varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2) \\ \wedge \forall (s_1, s_2) \in S_1 \times S_2, t_1 := \frac{\beta_{1j}^{j \in J'_1}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s'_1} \in T_1, \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\ \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \exists \frac{\beta_{2j}^{j \in J'_2}, g_2, \psi_2}{s_2 \xrightarrow{\alpha_2} s'_2} \in T_2, \\ \sigma \vdash \alpha_1 = \alpha_2 \wedge \bigwedge_{j \in J'_1 \cap J'_2} \beta_{1j} = \beta_{2j} \wedge g_2 \wedge R(s'_1, s'_2)\{\!\{\psi_1 \uplus \psi_2\}\!\} \end{aligned}$$

Which means a relation is a simulation refinement if when two automata are related then there must be a relation that relates initial states and valuations and for every pair of state and valuation of the automaton variables that are related, and every possible transitions with valuation of its free variables from the first automaton, there is a possible transition and its valuation such that the produced action matches, the holes actions with same names match and the target states are related.

This is a natural extenion of the notion of simulation on LTS, which is the same definition without valuation and holes (and guards). It will serve as a watchdog for any relation that will be defined.

6 Preliminary refinement relations

The main objective of the refinement relation in this section is to be able to say that a composition of two automata is a refinement of the base automaton ($a[b] \leq a$). However composing an open automaton can give an automaton where not much can be said in terms of hole indicies. So looking at restricted cases where holes are related in a specific manner can help understanding the general case. For two open automata $\mathrm{OA}_1 := \langle S_1, i_1, J_1, V_1, \varphi_1, T_1 \rangle$ and $\mathrm{OA}_2 := \langle S_2, i_2, J_2, V_2, \varphi_2, T_2 \rangle$, the following refinement relations are defined.

This relation is the basis for all the other ones. It applies on automaton with same holes. The goal is to caracterise open automata that have a more determined (more deterministic) behaviour without adding deadlocks.

Definition 9 (Hole-identical refinement)**.** If $J_1 = J_2$ then "OA_1 is a hole-identical refinement of OA_2", noted $\mathrm{OA}_1 \leq_= \mathrm{OA}_2$, is defined as:

$$\exists R : (S_1 \times S_2) \to \mathcal{F}_{V_1 \uplus V_2}, \varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2) \quad \wedge \quad R \text{ hole-identical simulation of } \mathrm{OA}_1 \text{ in } \mathrm{OA}_2$$

As in other simulation-like relations, R is a witness of $\mathrm{OA}_1 \leq_= \mathrm{OA}_2$. The standard requirements for (bi)simulations are that R relates initial states, transitions can be matched, and target of matched transitions with related sources are also related. For a bisimulation transitions are matched one-to-many from each automaton to the other, but for a simulation only one way is used. The definition above requires explicitely that initial states are related, the other requirement are encapsulated in the definition (below) of hole-identical simulation.

The specificities of open automata comes in when relating the (initial) states. The relation R relates i_1 to i_2 under the condition that the curent value of variables satisfies $R(i_1, i_2)$. If two states $s_1 \in S_1, s_2 \in S_2$ are not(/never) related then $\nvdash R(s_1, s_2)$. The predicate $R(i_1, i_2)$ is used to take into account the fact that a state is also constituted of the value of the variables. This is already used in FH-bisimulation introduced in previous articles about open automata. So the meaning of $\varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2)$ is that the initial states with the initial valuation are effectively related.

Definition 10 (Hole-identical simulation)**.** "R is a hole-identical simulation of OA_1 in OA_2" is defined as:

Any idea on how to note that? $R \vDash \mathrm{OA}_1 \leq_= \mathrm{OA}_2$ for instance.

$$\forall (s_1, s_2) \in S_1 \times S_2,$$

$$\left(\begin{array}{l} \forall t_1 := \dfrac{\beta_{1j}^{j \in J'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in T_1, \exists \left(\dfrac{\beta_{2xj}^{j \in J'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in T_2 \right)^{x \in X}, \\ \forall \sigma : (V_1 \uplus vars(t_1) \uplus V_2) \to \mathcal{P}, \\ \quad \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \bigvee_{x \in X} \left(\begin{array}{l} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j \in J'} \beta_{1j} = \beta_{2xj} \\ \wedge\, g_{2x} \wedge R(s_1', s_{2x}')\{\!\{\psi_1 \uplus \psi_{2x}\}\!\} \end{array} \right) \end{array} \right)$$

$$\wedge\, \forall \sigma : (V_1 \uplus V_2) \to \mathcal{P}, \sigma \vdash R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} guard(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1)$$

The first two lines quantify on source states, transitions (and target states) and source states valuations. The 3rd line is the condition to ensure that transitions are not matched to incompatible ones: For all variable assignement of the two automata and all hole actions (quantification of σ), assuming that the states were related and the transition in OA_1 is possible ($R(s_1, s_2) \wedge g_1$ part) then there is always a transition, not necessarily only one, which can be performed (g_{2x}), produce the same action ($\alpha_1 = \alpha_{2x}$), accept the same action from the holes ($\beta_{1j} = \beta_{2xj}$) and satisfy the predicate for relating the target states after variable update ($R(s_1', s_{2x}')\{\!\{\psi_1 \uplus \psi_{2x}\}\!\}$).

The notion of refinement here is stating that OA_1 is a refinement of OA_2 if OA_1 can be simulated in OA_2. The hole-identical part is referring to the fact that holes indicies are the same and identical holes indicies have to perform the same actions. On top of that the last line ensures no deadlock introduction. It can be interpreted "assuming the predicate holds and there is any possible transition in OA_2 (= no deadlock), then there must be a possible transition in OA_1 (= no deadlock)". It does not have to ensure that this transition isn't garbage or that the target states are related because the first part of the simulation already does it. Also if there was a deadlock then there is no transition because no trasition can be simulated in a deadlock, that's why it is more a deadlock equivalence than a deadlock reduction.

Example 4.

Lemma 1 (Equivalent definition)**.**

$$\left(\begin{array}{l} \forall t_1 := \dfrac{\beta_{1j}^{j\in J'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in T_1, \exists \left(\dfrac{\beta_{2xj}^{j\in J'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in T_2\right)^{x\in X}, \\ \forall \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\ \quad \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \left(\begin{array}{l} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J'} \beta_{1j} = \beta_{2xj} \\ \wedge\, g_{2x} \wedge R(s_1', s_{2x}')\{\!\{\psi_1 \uplus \psi_{2x}\}\!\} \end{array}\right) \end{array}\right)$$

$$\Longleftrightarrow$$

$$\left(\begin{array}{l} \forall t_1 := \dfrac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in T_1, \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\ \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \exists \dfrac{\beta_{2j}^{j\in J_2'}, g_2, \psi_2}{s_2 \xrightarrow{\alpha_2} s_2'} \in T_2, \\ \quad \sigma \vdash \alpha_1 = \alpha_2 \wedge \bigwedge_{j\in J_1'\cap J_2'} \beta_{1j} = \beta_{2j} \wedge g_2 \wedge R(s_1', s_2')\{\!\{\psi_1 \uplus \psi_2\}\!\} \end{array}\right)$$

The first property is one part of the hole-identical simulation, the second property is the central part of the refinement simulation requirement. If we prove the direct way of this equivalence and $\leq_=$ is a preorder then $\leq_=$ is a refinement simulation. The other way is there because I hope that it will be easier to use this formulation to prove that $\leq_=$ is a preorder. One may wonder why it was not used in the definition if it is equivalent. The reason is that the formulation used is compatible with SMT solver, the place where it will most probably be used in practise.

Proof.

$\Rightarrow$:

$\Leftarrow$: $\square$

Theorem 1. *The hole-identical refinement relation is a preorder.*

Proof.

Reflexivity: Let OA be an open automaton with variables in V. The automaton variables on the right hand side of the relation will be noted v' for the equivalent variable $v \in V$ in the automaton on the left hand side. The simulation $R = \left\{ (s,s) \mapsto \bigwedge_{v\in V} v = v' \,\middle|\, s \in S_1 \right\}$ is a witness of $\mathrm{OA} \leq \mathrm{OA}$. Checking it is a simple exercise left to the reader in order to understand how the definition works.

Transitivity: Let $\mathrm{OA}_1, \mathrm{OA}_2, \mathrm{OA}_3$ be open automata with respectively variables in V_1, V_2, V_3 and states S_1, S_2, S_3. And let R_{12} be a witness of $\mathrm{OA}_1 \leq_= \mathrm{OA}_2$ and R_{23} be a witness of $\mathrm{OA}_2 \leq_= \mathrm{OA}_3$.

$$R_{13}(s_1, s_3) := \exists V_2, \bigvee_{s_2 \in S_2} R_{12}(s_1, s_2) \wedge R_{23}(s_2, s_3)$$

Where $\exists S, P$, with $S := \left\{ v_1, v_2, \cdots, v_{|S|} \right\}$ a finite set of variables, means $\exists v_1, \exists v_2, \cdots, \exists v_{|S|}, P$.

Now let's prove that R_{13} is a witness of $\mathrm{OA}_1 \leq_= \mathrm{OA}_3$:

-

□

Theorem 2 (Hole-identical refinement correction)**.**

Proof. □

The goal of the following relation is to capture the case where holes are filled with automata that do not have any hole. It is supposed to be a relation for which filling holes with fully specified automata is a considered refinement.

Definition 11 (Hole-subset refinement)**.** If $J_1 \subseteq J_2$ then "OA_1 is a hole-subset refinement of OA_2", noted $\mathrm{OA}_1 \leq_{\subseteq} \mathrm{OA}_2$ is defined as:

$$\exists R : (S_1 \times S_2) \to \mathcal{F}_{V_1 \uplus V_2}, \varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2) \wedge R \text{ hole-subset simulation of } \mathrm{OA}_1 \text{ in } \mathrm{OA}_2$$

This definitions is essentially the same as the hole-identical one excepted the constraint on holes which has been softened.

Definition 12 (Hole-subset simulation)**.** "R is a hole-subset simulation of OA_1 in OA_2" is defined as:

$$\forall s_1 \in S_1, s_2 \in S_2,$$

$$\left(\begin{array}{l} \forall \dfrac{\beta_{1j}^{j\in J'_1}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s'_1} \in T_1, \exists \left(\dfrac{\beta_{2xj}^{j\in J'_{2x}}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s'_{2x}} \in T_2 \right)^{x\in X}, \\ (\forall x \in X, J'_1 = J'_{2x} \cap J_1) \wedge \forall \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\ \quad \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \left(\begin{array}{l} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J'_1} \beta_{1j} = \beta_{2xj} \\ \wedge\, g_{2x} \wedge R(s'_1, s'_{2x}) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \end{array} \right) \end{array} \right)$$

$$\wedge\, \forall \sigma : (V_1 \uplus V_2) \to \mathcal{P}, \sigma \vdash R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} \mathit{guard}(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} \mathit{guard}(t_2)$$

The difference with the hole-identical simulation is that the holes involved in the matched transitions don't have to be exactly the same anymore. The new constraint is that the holes in common (that is J_1') should be involved at the same time and their action have to match. What happens on the other holes is unspecified except for the fact that, by being free variables they still need to be valued such that the transition is possible.

Example 5.

An alternative version of this definition could enforce $\exists J_2' \subseteq J_2, \forall x \in X, J_{2x}' = J_2'$. Let's call it the alternative hole-subset simulation. This version would mean that the holes involved in every matched transitions must be the same. While this may seem more natural this leads to the unwanted behaviour that an automaton FH-bisimilar to a refinement of some specification might not be a refinement of that specification:

Proposition 1 (Alternative hole-subset simulation is not compatible with FH-bisimulation)**.** *TODO: collapse a transition that was differenciated by having one filled hole involved and it should be simple*

This is a sufficient reason to discard this version although it seems more natural.

Proposition 2 (Hole-subset refinement is compatible with FH-bisimulation)**.**

Example 6.

Proposition 3 (Hole-subset refinement is an extension of hole-identical refinement)**.** *Hole-subset refinement and hole-identical match when holes are identical.*

Proof. When $J_1 = J_2$, $J_1' = J_{2x}' \cap J_1 \iff J_1' = J_{2x}' \cap J_2 \iff J_1' = J_{2x}'$, which is the implicit constraint on hole actions in the hole-identical simulation. By that rewriting their definition match. □

Let's assume that this relation is also correct, the proof of correctness is given later for a more general refinement relation.

Theorem 3 (Composition is a refinement)**.**

Proof. □

Theorem 4 (Context refinement)**.**

Proof. □

Theorem 5 (Congruence with composition)**.**

Proof. □

The goal of the following relation is to capture the case where holes are filled with one hole automata. It is supposed to be a relation for which filling holes with one hole automata is a considered refinement.

Definition 13 (Hole-matching refinement)**.** If $|J_1| = |J_2|$ then "OA_1 is a hole-matching refinement of OA_2", noted $\mathrm{OA}_1 \leq_{\#} \mathrm{OA}_2$ is defined as:

$$\begin{aligned}&\exists R : (S_1 \times S_2) \to \mathcal{F}, \quad \exists f : J_1 \setminus J_2 \to J_2 \setminus J_1,\\ &\qquad \varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2) \quad \wedge \quad f(J_1 \setminus J_2) = J_2 \setminus J_1\\ &\qquad\qquad \wedge\ R \text{ hole-f-matching simulation of } \mathrm{OA}_1 \text{ in } \mathrm{OA}_2\end{aligned}$$

This definitions is essentially the same as the hole-identical except that the constraint on holes has been softened and it is compensated with a invertible map between non-shared holes.

Definition 14 (Hole-f-matching simulation)**.** R is a hole-f-matching simulation of OA_1 in OA_2 is defined as:

$$\begin{aligned}&\forall s_1 \in S_1, s_2 \in S_2,\\ &\left(\begin{array}{l} \forall \dfrac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in T_1, \exists \left(\dfrac{\beta_{2xj}^{j\in J_{2x}'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in T_2 \right)^{x\in X}, \\ (\forall x \in X, J_1' \cap J_2 = J_{2x}' \cap J_1 \wedge f(J_1' \setminus J_2) \subseteq J_{2x}') \wedge \forall \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\ \quad \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \left(\begin{array}{l} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J_1'\cap J_2} \beta_{1j} = \beta_{2xj} \\ \wedge\ g_{2x} \wedge R(s_1', s_{2x}')\{\!\{\psi_1 \uplus \psi_{2x}\}\!\} \end{array} \right) \end{array} \right)\\ &\qquad \wedge \forall \sigma : (V_1 \uplus V_2) \to \mathcal{P}, \sigma \vdash R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} \mathit{guard}(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} \mathit{guard}(t_1)\end{aligned}$$

7 Refinement relation for open automata

Definition 15 (Open automata refinement)**.** For any two open automata $\mathrm{OA}_1 \coloneqq \langle S_1, i_1, J_1, V_1, \varphi_1, T_1 \rangle$ and $\mathrm{OA}_2 \coloneqq \langle S_2, i_2, J_2, V_2, \varphi_2, T_2 \rangle$, "$\mathrm{OA}_1$ is a refinement of OA_2", noted $\mathrm{OA}_1 \leq \mathrm{OA}_2$, is defined as:

$$\begin{aligned}&\exists R : (S_1 \times S_2) \to \mathcal{F}_{V_1 \uplus V_2}, \exists f : J_1 \setminus J_2 \to J_2 \setminus J_1,\\ &\qquad\qquad \varphi_1 \uplus \varphi_2 \vdash R(i_1, i_2) \wedge R \text{ f simulation of } \mathrm{OA}_1 \text{ in } \mathrm{OA}_2\end{aligned}$$

Definition 16 (f simulation)**.** R is a f simulation of OA_1 in OA_2 is defined as:

$$
\begin{gathered}
\forall s_1 \in S_1, s_2 \in S_2, \\
\left(
\begin{array}{l}
\forall \dfrac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in T_1, \exists \left(\dfrac{\beta_{2xj}^{j\in J_{2x}'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in T_1 \right)^{x\in X}, \\
(\forall x \in X, J_1' \cap J_2 = J_{2x}' \cap J_1 \wedge f(J_1' \setminus J_2) \subseteq J_{2x}') \wedge \forall \sigma : (V_1 \uplus \mathit{vars}(t_1) \uplus V_2) \to \mathcal{P}, \\
\quad \sigma \vdash R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \left(\begin{array}{c} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J_1'\cap J_2} \beta_{1j} = \beta_{2xj} \\ \wedge\, g_{2x} \wedge R(s_1', s_{2x}')\{\!\{\psi_1 \uplus \psi_{2x}\}\!\} \end{array} \right)
\end{array}
\right) \\
\wedge\, \forall \sigma : (V_1 \uplus V_2) \to \mathcal{P}, R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} \mathit{guard}(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} \mathit{guard}(t_1)
\end{gathered}
$$

The objective of the first part of this definition is to be able to simulate OA_1 in OA_2 when holes of the same name receive the same closed automaton while other holes receive "compatible" closed automata. The objective of the other part of this definition is to prevent the appearance of new deadlocks, which should mean with the first property that the compared automaton are deadlock equivalent.

8 New equivalence relation induced by pre-order

9 Conclusion