Refinements for open automata

Rabéa[1][1111−2222−3333−4444], Quentin[1,2][1111−2222−3333−4444], Ludo[2][0000−1111−2222−3333], and Eric[3][2222−−3333−4444−5555]

[1] blabla lncs@fff
http://www.springer.com/gp/computer-science/lncs
[2] blibli
{abc,def}@fff

Abstract. **[TODO: The abstract should briefly summarize the contents of the paper in 150–250 words.]**
Establishing equivalences and refinement relations between programs is an important mean for verifying the correctness of programs, by formally proving the relation between a specification and an implementation, proving that two implementation are equivalent, or justifying optimisations and transformations, by establishing that the behaviours of a modified program simulate those of to the source one.
In this article, we discuss a notion of refinement between so-called "open automata", which are symbolic behavioural models for communicating systems. Open automata may have "holes" modelling elements of their context, and can be composed by instantiation of the holes. This allows for a compositional approach for verification of their behaviour, essential to address proofs about large realistic systems.
We define several variants of refinement between systems including either equal or different sets of holes, and show under which conditions these refinements are preserved by composition of open automata. We also discuss the relations between these refinements and the existence of deadlocks. We illustrate these notions on several simple use-cases.

Keywords: Labelled transition systems · Refinement · Composition.

1 Introduction

[TODO:1.5 pages]

Contributions:

We first: introduce open automata (NOT NEW), their composition, and the deadlock-free composition.

Then we define a refinement relation for open automata that has the following characteristics:

- good behaviour wrt composition
- refinement does not introduce deadlock
- a first relation that focuses on the automaton part and a second that deals with open behaviour

we will see that having at the same time composition and transitivity also raises challenges; we introduce a new form of refinement relation that addresses this challenge.

2 Related Work

[TODO:1 page]

The notion of refinement aims captures the relation between a specification and an implementation of the same component. It is usually defined as trace inclusion or simulation [**?**,**?**]; this ensures that all behaviours of the implementation must be also behaviours of the specification. This definition, which is based on systems whose behaviour is fully defined, is not well-suited for open systems, as it requires to reason also about unspecified behaviours.

There are some works that have focused on the refinement of open systems. Defining refinement of open systems as trace inclusion is addressed as a notion of subtyping in type theory [**?**,**?**]. Such refinement is instead based on interface-oriented approach, it allows the expression more internal choices and less external choices. The refinement of open systems is also defined in terms of alternating simulation [**?**,**?**], which deals with game-based models. Alternating simulation that is originating from the game theory [**?**] allows the study of relation between individual components by viewing them as alternating transition systems. In particular, a refinement of game-based automata expresses that the refined component can offer more services (input actions) and fewer service demands (output actions). However, the composition of such automata may lead to illegal states, where one automaton issues an output that is not acceptable as input in the other one. The theory of alternating simulation provides an optimistic approach to compute compatibility between automata based on the fact that each automaton expects the other to provide legal inputs, i.e, two components can be composed if there is an environment where they can work together. As we shall see in this paper, our approach to design refinement has some commonalities with that of the above mentioned [**?**]: both are process-oriented approach even if they are not based on the same notion of simulation and they are based on optimistic approach to composition. For the composability, we shall see that we use the notion of comparability of holes (similar to the notion of compatibility), which is explicitly encompassed in the definition of composition.

Previous work on open automata focused on equivalence relations compatible with composition. In an article by Hou, Zechen and Madelaine [**?**], a computable bisimulation is introduced and proved equivalent to the previous bisimulation already introduced. In a more recent work by Ameur-Boulifa, Henrio and Madelaine [**?**], a weak version of the bisimulation on open automata is introduced. These works differ from ours because the relation introduced in this report is a refinement relation in the form of a simulation and not a bisimulation. Also we do not have results as strong as computability neither a weak version able to tackle silent actions.

[TODO:Some related work on other models than open automata introduce refinement relations. In a chapter by Bellegarde, Julliand and Kouchnarenko [?], a simulation relation on transition systems is introduced. This simulation encompass action refinement, is able to deal with silent actions and is compatible with parallel composition. Here the refinement relation does not consider action refinement as valid but it should be done in future work. Also they check how LTL properties are preserved or combined using their refinement which we do not do. However their model is less expressive: the transition system model is less expressive than open automata and the parallel composition is less expressive than composition on open automata. In a later report by Kouchnarenko and Lanoix [?], the refinement relation they introduce is on LTS (labelled transition systems). Their relation additionally prevents deadlock and livelocks. The composition is also extended to synchronised composition which is more expressive. In our work we also deal with deadlocks but not with livelocks since the latter arise only with silent actions. This work is closer than the previous one to what we do here, still open automata are more expressive than LTS and composition is more general than synchronised composition.]

A refinement relation on models nearer to open automata is introduced in an article by Zhang, Meng and Lo [**?**]. In their article they work with transition systems with variable which makes the state space potentially infinite. This aspect is also present in open automata. They show how invariants, a notion near to our reachability predicates, are composed. By relation on these invariants they introduce several refinement relations. We could have done something similar for non-blocking composition reachability predicates which are introduced in Section **??**.

On the deadlock prevention aspect, an article by Dihego, Sampaio and Oliveira [**?**] present a refinement relation on process algebra (translated to LTS). This refinement relation is a special case of inheritance and prevents the introduction of deadlocks. Their refinement and inheritance are quite the opposite of our refinement in terms of new behaviours. They have channels, interfaces, inputs and outputs, which in the open automata model can be compared to action labels, holes and action data for both inputs and outputs. They have a rich composition as open automata but the introduction of deadlock is already prevented by a well chosen set of composing operations. Also their composition is slightly different than the one on open automata because they can cause loops by linking two channels of the same process, where in open automata the composition makes an oriented tree. In their model there is an explicit deadlock and a successful termination where in open automata there are no explicit termination. We define a deadlock as a configuration without possible transition and assume what is a deadlock when comparing the open automata. To define their refinement and inheritance relation they use trace and failure semantics, which are weaker than (bi)simulations [**?**] and could break with open automata composition.

3 Background: Open Automata and their Composition

[TODO:3.5 pages]

This section presents our notations and the principles of automata. Except for minor changes in the notations, compared to previous works [**?**] the only new contribution is the definition of a composition operator for open automata.

Families 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 disjoint union on set is noted $\uplus$[3]. Disjoint union is also used on maps. In a formula, a quantifier followed by a finite set will be used as a shorthand for the quantification on every variable in the set: $\forall \{a_1, \ldots, a_n\}, \exists \{b_1, \ldots, b_m\}, P$ means $\forall a_1, \ldots, \forall a_n, \exists b_1, \ldots, \exists b_m, P$.

An expression algebra E is the disjoint union of terms, actions, and formulas $E = \mathcal{T} \uplus \mathcal{A} \uplus \mathcal{F}$. $\mathcal{T}$ and $\mathcal{A}$ are term algebras **[TODO:do we explain term algebras]**. The formulas $\mathcal{F}$ contain at least first order formulas and equality[4] over $\mathcal{T}$ and $\mathcal{A}$.

$vars(e)$ is the set of variables in $e \in E$ that are not bound by any binder. An expression is closed if $vars(e) = \varnothing$. The set $\mathcal{P}$ denote values which is a subset of closed terms. $\mathcal{F}_V$ is the set of formulas f that only use variables in V, i.e. the formulas such that $vars(f) \subseteq V$.

The substitution in $e \in E$ of $x \in vars(e)$ by $t \in \mathcal{T}$, is denoted $e[t/x]$, and its generalisation to the parallel substitution of variables in V by $\psi : V \to \mathcal{T}$ is denoted $e\{\!\{\psi\}\!\}$.

We suppose given a decidable satisfiability relation on formulas, $\vdash f$ is the satisfiability over closed formulas. We will use two satisfiability relations:

- The satisfiability of a formula $f \in \mathcal{F}$ under some valuation $\sigma : V \to \mathcal{P}$ is defined as follows: $\sigma \vdash f \iff \vdash \exists vars(f\{\!\{\sigma\}\!\}), f\{\!\{\sigma\}\!\}$
- The satisfiability of a formula $f \in \mathcal{F}$ with some variable set V as context is defined as follows: $V \vdash f \iff \vdash \forall V, \exists (vars(f) \setminus V), f$

3.1 Open Automata

Open automata (abbreviated OA) are labelled transition systems with variables that can be used to compose other automata: they are made of transitions that are dependent of the actions of "holes", a composition operation consists in filling a hole with another automaton to obtain a more complete automaton. The variables makes the OA symbolic, and the holes allow for a partial definition of the behaviour.

Definition 1 (Open transition, Open automaton (OA)). *An* open automaton *is a tuple* $\langle S, s_0, V, \sigma_0, J, T \rangle$ *with* S *the set of states,* $s_0 \in S$ *the initial state,* V *the finite set of variable names,* $\sigma_0 : V' \to \mathcal{P}$ *the initial valuation of*

[3] $\uplus$ notation either supposes that the sets are disjoint or rename conflicting objects depending on the context

[4] Equality does not need to be only syntactic.

variables where $V' \subseteq V$, J the set of hole names and T the set of open transitions.

An open transition *is a tuple* $\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'}$ *with $s, s' \in S$ the source and target states, $\alpha \in \mathcal{A}$ the produced action, $J' \subseteq J$ the holes involved in the transition, $\beta_j \in \mathcal{A}$ the actions of the holes, $g \in \mathcal{F}$ the guard and $\psi : V \to \mathcal{T}$ the variable assignments. To be well-formed, an open transition should use only variables of the automaton and variables appearing in the involved actions, formally:*

$$vars(g) \subseteq vars(\alpha) \cup \bigcup_{j\in J'} vars(\beta_j) \cup V$$
$$\forall v \in V.\, vars(\psi(v)) \subseteq vars(\alpha) \cup \bigcup_{j\in J'} vars(\beta_j) \cup V$$

Definition 2 (Guard, Out-transition, Transition variables). *Let V be the variable names of the considered automaton, T its transitions and r one of its states. $\mathrm{OT}_T(r) \in T$ are called the out-transitions of the state r. $\mathrm{IT}_T(r) \in T$ are called the in-transitions of the state r. When the transition set is clear from the context, it will be omitted. The local variables of a transition $vars(t)$ are all variables appearing in transition t except the global variables of the automaton.*

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

$$guard\left(\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'}\right) = g \qquad vars\left(\frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'}\right) = \left(vars(\alpha) \cup \bigcup_{j\in J'} vars(\beta_j)\right) \setminus V$$

The following terminology will be used to reason on open automata

Definition 3 (Configuration, instantiated transition). *A pair consisting of a state and a valuation is called a* configuration. *An* instantiated transition *of an automaton $\langle S, s_0, V, \sigma_0, J, T\rangle$ is a transition $t\psi$ where $t \in T$ and ψ is a well-formed substitution with $\mathrm{dom}(\psi) = vars(t)$.*

Open automaton composition Open automata are partially specified automata, that partiality comes mostly from the holes. A hole is an interface in which we can plug an open automaton. The plugging operation is called composition. The composition of open automata was already implicitly defined by the means of composition on pNets in previous work [**?**] but never completely formalised on open automata. The definition of composition below is a direct translation of what happens with pNets composition without the need of introducing pNets.

[TODO:peut etre echanger c et p? ou c devient p et p devient?]

Definition 4 (Composition of open automata). *The composition of $A_c = \langle S_c, s_{0c}, V_c, \sigma_{0c}, J_c, T_c\rangle$ in the hole $k \in J_p$ of $A_p = \langle S_p, s_{0p}, V_p, \sigma_{0p}, J_p, T_p\rangle$ is the OA defined as follows:*

$$A_p[A_c/k] ::= \langle S_p \times S_c, (s_{0p}, s_{0c}), V_p \uplus V_c, \sigma_{0p} \uplus \sigma_{0c}, J_c \uplus J_p \setminus \{k\}, T\rangle$$

with

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

The first OA decides when the second can evolve by involving its hole in a transition: The action emitted when A_c makes a transition is synchronised with the action of the hole k in transitions of A_p .

Relations between open automata A relation (bisimulation, refinement, etc.) between open automata requires to compare their states. To do so we will suppose that the variables of the two OAs are disjoint (a renaming of variables may have to be applied before comparing OA states)

Definition 5 (Relation on OA states). *Suppose V_1 and V_2 are disjoint. A relation on states of $\langle S_1, s_{01}, V_1, \sigma_{01}, J_1, T_1 \rangle$ and $\langle S_2, s_{02}, V_2, \sigma_{02}, J_2, T_2 \rangle$ is a function $R : S_1 \times S_2 \to \mathcal{F}_{V_1 \uplus V_2}$.*

The idea is that two states are related if the variables they refer to verify a certain formula. Additionally, we may check that initial states of the automata are related by checking that: $\sigma_{01} \uplus \sigma_{022} \vdash R(s_{01}, s_{02})$.

[TODO:I do not think we reason on configurations for the moment, if we want to, we should introduce this:] Two configurations $(s_1, \sigma_1) \in S_1 \times (V_1 \to \mathcal{P}), (s_2, \sigma_2) \in S_2 \times (V_2 \to \mathcal{P})$ are related iff $\sigma_1 \uplus \sigma_2 \vdash R(s_1, s_2)$.

A bisimulation for open automata TODO OR NOT?

3.2 Example

As an example, the traffic light system that controls traffic at an intersection. The open automaton modeling this system is illustrated in Figure 1. This automaton has three states remembering which coloured light is on (Red, Yellow or Green). It includes two holes: a controller (*ctl*) and a counter (*cnt*) depicting together the behaviour of the timer. The color switches when the counter and the controller component agree that the time is over. The new time limit can be set by the counter component and the exposed action to the environment is an unobservable action τ.

The open automata shown in Figure 2 models the timer. On the left, the controller component designed to be connected in the hole *ctl*. Its role is to decide the duration before switching the lights. We control the time interval for each light by setting them by prior knowledge: 17s for the first duration, 3s for the second, and 20s for the third. On the right, the tick counter component designed to be connected in the hole *cnt*.

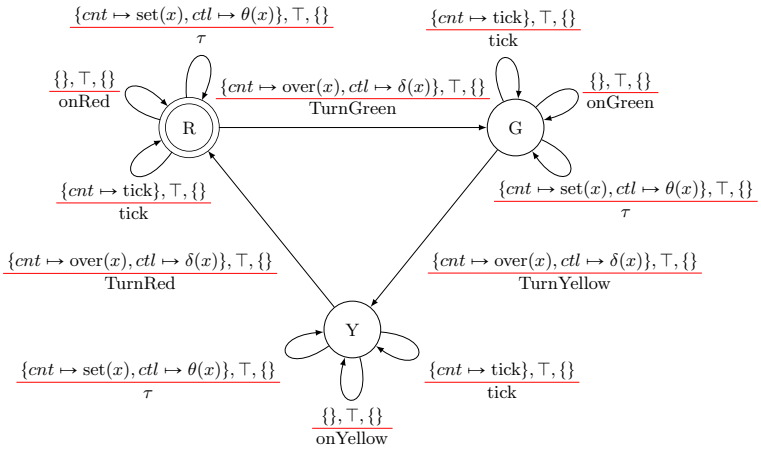


Fig. 1. The specification of a Traffic Light system

In Figure 4 we present the composition of the three automata. Each state of the result of the composition consists of a state of traffic light system together with a state of controller component and one of counter component. The composed automaton takes over the same steps as the traffic light automaton but it also includes new steps, indicting the change of states for the setting of timer. Its τ transitions involve both the traffic light automaton and the hole automata, they correspond to a joint step of sending time thresholds of the controller, the time setting of the counter. For instance, the τ transition starting from $R1S$, it is obtained by composition of both $\frac{\{cnt \mapsto \text{set}(x), ctl \mapsto \theta(x)\}, \top, \{\}}{R \xrightarrow{\tau} R}$, $\frac{\{\}, \top, \{\}}{1 \xrightarrow{\theta(17)} 2}$ and $\frac{\{\}, \top, \{t \leftarrow x, c \leftarrow 0\}}{S \xrightarrow{\text{set}(x)} C}$.

However, the joint composition of transitions $\frac{\{cnt \mapsto \text{set}(x), ctl \mapsto \theta(x)\}, \top, \{\}}{R \xrightarrow{\tau} R}$, $\frac{\{\}, \top, \{\}}{2 \xrightarrow{\delta(x)} 3}$ and $\frac{\{\}, c < t, \{c \leftarrow c + 1\}}{C \xrightarrow{\text{tick}} C}$ is not agreed because it would produce a transition whose guard is not satisfiable. It will produce the following transition: $\frac{\{\}, \top \wedge \top \wedge c < t \wedge \text{set}(x) = \text{tick} \wedge \theta(x) = \delta(x), \{c \leftarrow c + 1\}}{R2C \xrightarrow{\tau} R3C}$ which requires to be triggered the equality of completely different actions.

4 A First Refinement Relation

Similarly to FH-bisimulation [**?**] we are interested in finding relations between states of to open automata that contain variables and holes. However here we want to build a refinement relation that also guarantees that no deadlock is introduced when refining the automaton. The refinement relation should be conditioned by the internal states of the automata. More precisely, a refinement

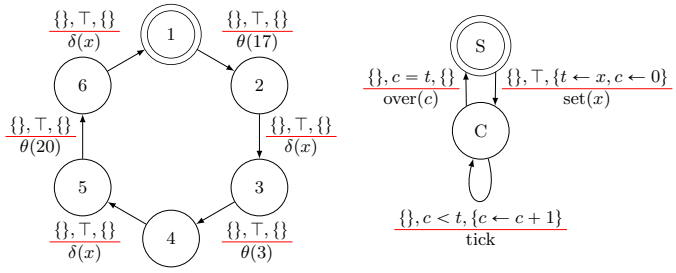


Fig. 2. (a) An example of controller component (b) An example of counter component

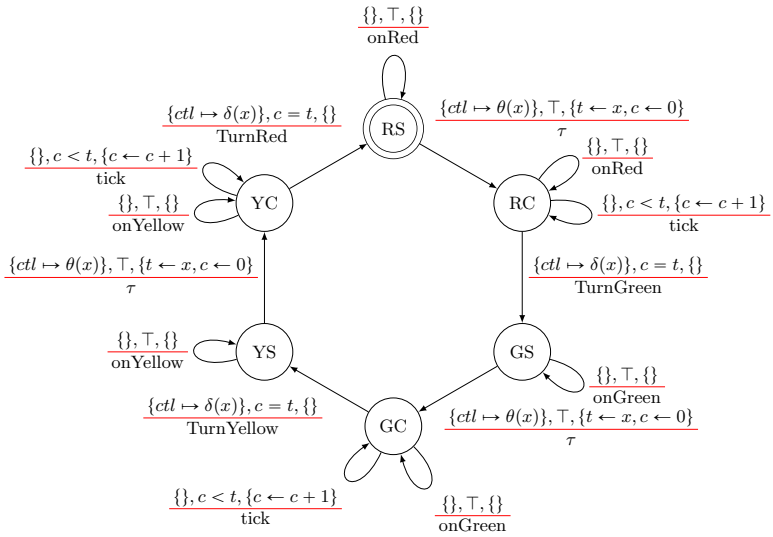


Fig. 3. The incomplete Specification Traffic Lights system

relation characterizes when two states are related, this characterisation is expressed as a predicate on the variables of the two automata. Thus, a refinement relation has the following signature: $S_1 \times S_2 \rightarrow \mathcal{F}_{V_1 \uplus V_2}$.

[TODO:check deadlock reduction/reducing in the literature] We first define when a relation is deadlock reducing, before characterising a first simple refinement relation to compare automata with identical holes.

4.1 Deadlock Reduction, Reachability, and Composition

A notion that is often used in the context of refinement is the notion of deadlock reduction. This property considers that two states related by a given relation and states that if one state can do a transition, then the other can do a transition too. This notion is not much interesting in the general case as there is a priori no relation between the two transitions. However, when the relation that relates

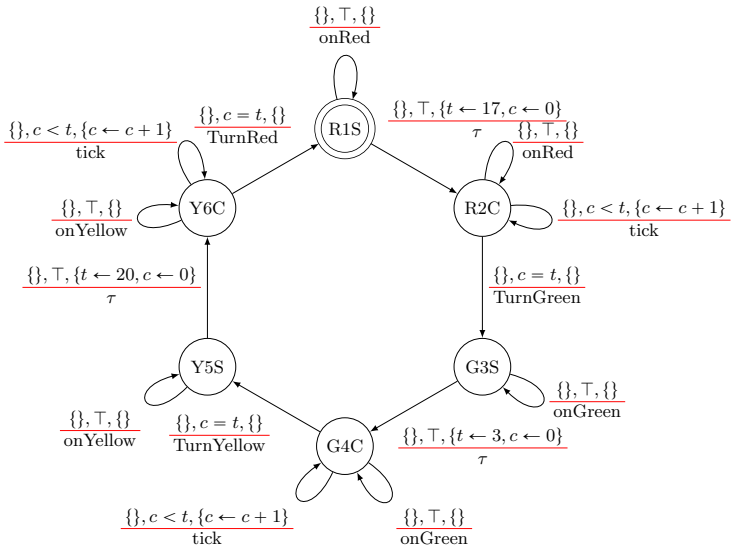


Fig. 4. The full Traffic Lights system

states is a simulation, this will relate the possible transitions and the deadlock reduction will become a valuable property.

Next definition states that if T_2 is not in a deadlock position then T_1 can do a transition but not necessarily with the same input values. This is very weak but sufficient when composed with the def of simulation.

Definition 6 (Deadlock reduction).
Let $\langle S_1, s_{01}, V_1, \sigma_{01}, J_1, T_1\rangle$ and $\langle S_2, s_{02}, V_2, \sigma_{02}, J_2, T_2\rangle$ be two OAs. A relation on OA configurations $R : S_1 \times S_2 \to \mathcal{F}_{V_1 \uplus V_2}$ is deadlock reducing if it satisfies the following[5]*:*

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

$$V_1 \uplus V_2 \uplus \biguplus_{t_2 \in \mathrm{OT}(s_2)} vars(t_2) \vdash \left(R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} guard(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \right)$$

Because of the symbolic nature of OAs and their structure, in this definition, the fact that a guard is true is sufficient to reason on the possible paths. This slightly simplifies the definition and makes the characterisation of transition that can be triggered very symbolic. An equivalent but more classical definition of deadlock reduction could be expressed as well. It states that if there is a transition that can be triggered in the first automaton, then there is a transition from the related state in the second automaton that can be triggered. This second

[5] Note that variables of T_1 are existentially quantified in the proposition.

definition is more verbose, in particular because of the multiple quantifiers over transitions and automaton state. The alternative definition is equivalent to this one when reasoning on a refinement relation (but not in general).

Unfortunately, this property is not compositional: the composition operator can itself introduce a deadlock. In other words, when filling the hole of two related automata with a third one, even if there is a deadlock reduction between the two original automata , there might not be a deadlock reduction in the composed ones. The same problem may arise when two related automata are composed in the same hole of a third one.

This creates a conflict between deadlock reduction and the properties involving composition. One possible solution to avoid this conflict is to only consider a composition that do not introduce deadlocks, we will define such a composition below. But before defining non-blocking composition, we first introduce a notion of state reachability for open automata.

Definition 7 (Reachability). *For any open automata $A = \langle S, s_0, V, \sigma_0, J, T\rangle$, a reachability predicate $\checkmark_A : S \to \mathcal{F}_V$ is any predicate on states that is valid on initial state, and preserved across transitions:*

$$\sigma_0 \vdash \checkmark_A(s_0) \quad \wedge \quad \forall t = \frac{\beta_j^{j\in J'}, g, \psi}{s \xrightarrow{\alpha} s'} \in T, vars(t) \vdash \left(\checkmark_A(s) \wedge g \implies \checkmark_A(s')\{\!\{\psi\}\!\}\right)$$

Reachability takes into account all paths, and can over-approximate the reachable configurations. From an automation point of view, finding the most precise reachability predicate for a given automata is not decidable because of the symbolic nature of open automata, but only an over-approximation is necessary. An automatic tool would only need to find an over approximation of reachability to reason on composition that is compatible with deadlock reduction. We call *non-blocking composition* a composition that can be safely be used to compose open automata that are involved in a deadlock reducing relation.

Definition 8 (Non-blocking composition). *Let $A_i = \langle S_i, s_{0i}, V_i, \sigma_{0i}, J_i, T_i\rangle$ with $0 \leqslant i \leqslant n$ be a family of open automata. Let $A = A_0[\![j_i \mapsto A_i | 1 \leqslant i \leqslant n]\!]$ and $A = \langle S, s_0, V, \sigma_0, J, T\rangle$.*

The composition $A_0[\![j_i \mapsto A_i | 1 \leqslant i \leqslant n]\!]$ is non-blocking if A has a reachability predicate such that, for each reachable configuration, if there is a possible transition in A_0 then there is a possible transition in A:

$$\forall s = (s_0, s_1, .., s_n) \in S, V \uplus \biguplus_{t_0 \in \mathrm{OT}(s_0)} vars(t_0) \vdash \left(\checkmark_A(s) \wedge \bigvee_{t_0 \in \mathrm{OT}(s_0)} guard(t_0) \implies \bigvee_{t \in \mathrm{OT}(s)} guard(t)\right)$$

Again we use guards to ensure that the transition can occur. It is not sufficient to ensure the existence of equivalent transitions in general, but it will be sufficient in the context of refinement.

4.2 Refinement Relations for automata with the same holes

We define here simulation between labelled transition systems to Hole-equal simulation between open automata that have exactly the same holes.

Definition 9 (Hole-equal simulation). *Consider two OAs* $\langle S_1, s_{01}, V_1, \sigma_{01}, J_1, T_1\rangle$ *and* $\langle S_2, s_{02}, V_2, \sigma_{02}, J_2, T_2\rangle$ *with* $J_1 = J_2$*, the relation on configurations* $R : S_1 \times S_2 \to \mathcal{F}_{V_1 \uplus V_2}$ *is a hole-equal simulation from* S_1 *to* S_2 *if the following conditions hold:*

(1) $\sigma_{01} \uplus \sigma_{02} \vdash R(s_{01}, s_{02})$

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

$$\forall t_1 = \frac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in \mathrm{OT}(s_1), \exists \left(t_{2x} = \frac{\beta_{2xj}^{j\in J_{2x}'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in \mathrm{OT}(s_2) \right)^{x\in X},$$
$$\left(\forall x \in X, J_{2x}' = J_1'\right)$$
$$\wedge\ V_1 \uplus V_2 \uplus \mathit{vars}(t_1) \vdash$$
$$\left(R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \begin{pmatrix} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J_{2x}'} \beta_{1j} = \beta_{2xj} \\ \wedge\ g_{2x} \wedge R\left(s_1', s_{2x}'\right) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \end{pmatrix} \right)$$

(3) R is deadlock reducing.

Note in this definition, instead of matching an instantiated transition of the first automata to another instantiated transition of the second, it matches an open transition t_1 to a family of covering open transitions $t_{2x}^{x\in X}$.

Intuitively, this means that for every pair of related states (s_1, s_2) of the two automata, and for every transition of the first automaton from s_1, there is a set of matching transitions of the second automaton from s_2 such that the produced action match, the actions of the same holes and the successors are related after variable update. Our definition captures a simple kind of sub-classing of open automata with the same holes. It is stronger than a strict simulation since it matches a transition with a family of transitions. With such a relation we are able to check the refinement between two open automata with the same level of abstraction but specified differently, for example, by duplicating states, removing transitions, reinforcing guards, modifying variables.

We will show in Section 6 that this refinement relation has good properties in terms of transitivity, compositionality, reflexivity, etc.

The refinement relation defined above is insufficient in the setting of composition which is the main advantage of the open automaton-based approach. Indeed, it should be possible to refine an automaton by filling its hole, providing a concrete view of a part of the application that was not specified originally. More generally, it should be possible to relate automata that do not have the same holes because composition is a crucial part of system specification. Thus, we believe composition should also be a form of refinement and call this feature *refinement through composition*. However, filling holes can result in a system with more or less holes than the original system because the plugged subsystem can

contain itself many holes. Next section will define a more powerful refinement relation able to reason on automata with different sets of holes.

[TODO:add an example]

5 A Refinement Relation that Takes Holes into Account

[TODO:3.5 pages]

This section extends the preceding relation to automata where the set of holes is not the same. A simple use-case for this is filling a hole with a completely defined automaton. In this case, we want to ensure that the automaton with a filled hole is a refinement of the other automata: actions of the identical holes will be taken into account the same way, and filling the hole partially reduces the behaviour of the automaton.

The major challenge in the design of this relation is to maintain a form of transitivity while being able to take into account the actions of some of the holes. A naive definition of refinement would ensure that the holes that are identical in the two open automata are taken into account in the simulation. Unfortunately considering all the common holes does not ensure transitivity of the simulation for the following reason. If A_1 simulates A_2 and A_2 simulates A_3, and one hole j appears in A_3 and in A_1 but not in A_2 then we have no guarantee on the way A_1 and A_3 take the actions of this his hole into account, thus a refinement between and A_1 and A_3 would require conditions involving actions of the hole j which cannot be ensured. The way we solve this issue is to remember in the simulation relation which holes have been compared. This way the relation is parameterized by the set of holes that belong to the two automata and are taken into account. This way, in the example above, we would have no guarantee on actions the hole j by transitivity but can state a refinement relation with guarantees on the actions of the other holes.

In the following definition we add a parameter H which is the set of holes tracked by the refinement relation and adapt the definition by ignoring actions of the holes that are not in H.

Definition 10 (Open automata refinement). *For two open automata* $A_1 \coloneqq \langle S_1, s_{01}, J_1, V_1, \sigma_{01}, T_1\rangle$ *and* $A_2 \coloneqq \langle S_2, s_{02}, J_2, V_2, \sigma_{02}, T_2\rangle$, A_1 *is a refinement of* A_2 *tracking holes* H, *noted* $A_1 \leqslant_H A_2$, *with* $H \subseteq J_1 \cap J_2$, *if there is a relation* $R : (S_1 \times S_2) \to \mathcal{F}_{V_1 \uplus V_2}$ *such that:*

(1) $\sigma_{01} \uplus \sigma_{02} \vdash R(s_{01}, s_{02})$

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

$$\forall \frac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in \mathrm{OT}(s_1), \exists \left(\frac{\beta_{2xj}^{j\in J_{2x}'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in \mathrm{OT}(s_2) \right)^{x\in X},$$
$$\left(\forall x \in X, J_{2x}' \cap H = J_1' \cap H\right)$$
$$\wedge\ V_1 \uplus V_2 \uplus vars(t_1) \vdash$$
$$\left(R(s_1, s_2) \wedge g_1 \implies \bigvee_{x\in X} \left(\begin{array}{c} \alpha_1 = \alpha_{2x} \wedge \bigwedge_{j\in J_{2x}'\cap H} \beta_{1j} = \beta_{2xj} \\ \wedge\ g_{2x} \wedge R\left(s_1', s_{2x}'\right) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \end{array} \right) \right)$$

(3) R is deadlock reducing.

Giving R and H is sufficient to characterise the refinement, so we call (R, H) a hole-tracking simulation of A_1 by A_2. Hole in H are called tracked holes.

Note that every action of the holes outside H is unconstrained in the related automata.

[TODO:example]

6 Properties of our Refinement Relations

[TODO:1.5 pages] We now state the properties of our refinement relations, proofs of the properties can be found in the appendices. We express these properties in terms of open automata refinement because hole-equal simulation is a particular case of Definition 10 when $J_1 = J_2 = H$, and thus most of the properties shown here are easy to adapt to hole-equal simulation.

The first crucial property of refinement is that it is a preorder on the set of open automata. This property enables stepwise refinement.

Theorem 1 (Refinement is a preorder). *Refinement is reflexive and transitive: it is a preorder on the set of open automata.*

The relation $\leqslant_H$ is reflexive, $A \leqslant_H A$, by taking $R(s_1, s_2) \mapsto \bigwedge_{v\in vars(s_1)} v = v$ to be $s_1 = s_2$ and checking the above conditions (Definition 10), we can see that the relation is indeed reflexive. Appendix A presents the proof of transitivity. It is done classically by identifying the relation between A_1 and A_3 that is a refinement. What is less classical is the definition of this relation because it is a boolean formula. For each couple of states s_1 and s_3 of A_1 and A_3 we build a a formula that defines the refinement relation. To do this, we take the disjunction of formulas relating s_1 and s_3, and passing by all states s_2 of A_2. More precisely, we define a relation of the following form:

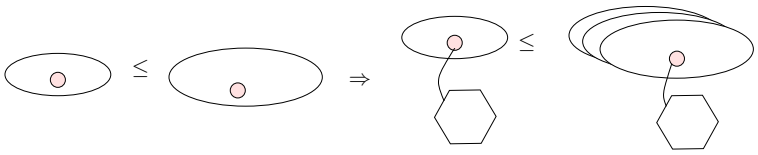
$$R_{13}(s_1, s_3) = \bigvee_{s_2\in S_2} \left(R_{12}(s_1, s_2) \wedge R_{23}(s_2, s_3)\right)$$

We then prove that this relation defines is a refinement, according to Definition 10.

The next to theorems state that refinement is compositional in the sense that it is sufficient to prove refinement for the composed automata separately to obtain a refinement relation. This can be split into theorems that can be trivially composed by transitivity.

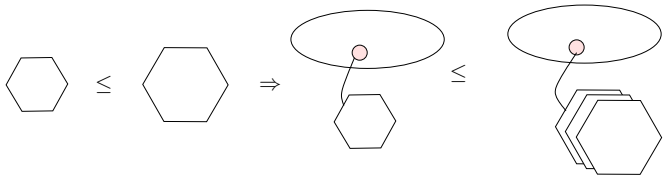
Theorem 2 (Context refinement). *Let A_1, A_2 and A_3 be three open automata with $A_1 \leqslant_H A_2$. Let J_3 be the set of holes of A_3. Suppose that $k \in H$ and that $A_1[A_3/k]$ is non-blocking. We have:*

$$A_1[A_3/k] \leqslant_{J_3 \uplus H \setminus \{k\}} A_2[A_3/k]$$



Theorem 3 (Congruence). *Let A_1, A_2 and A_3 be three open automata with $A_2 \leqslant_H A_3$. Suppose that $k \in H$ and that $A_1[A_2/k]$ is non-blocking. We have:*

$$A_1[A_2/k] \leqslant_{J_1 \uplus H \setminus \{k\}} A_1[A_3/k]$$



[TODO:i did not put the global composition theorem, de we state it?]

- composition is a refinement (under which condition? non-blocking composition?)
- what is the property wrt deadlocks?
- relation with fh bisimulation

7 Conclusion

[TODO:0.5 pages]

8 a recuperer si on se rend compte que c'est utile

When we will introduce refinements in Section **??**, setting an undefined variable will be considered a valid refinement, for instance a 5 bits register is a particular n bits register.

A Proof of Transitivity for Refinement

If $A_1 \leqslant_H A_2$ and $A_2 \leqslant_{H'} A_3$, then $A_1 \leqslant_{H \cap H'} A_3$.

Proof. If $A_1 \leqslant_H A_2$ then there is R_{12} a relation between states of A_1 and of A_2; If $A_2 \leqslant_{H'} A_3$ then there is R_{23} a relation between states of A_2 and of A_3. We build a relation between states of A_1 and of A_3 as follows: for each pair of states s_1, s_3, for each state s_2 such that R_{12} relates s_1 and s_2, and R_{23} relates s_2 and s_3. Let R_{13} be the relation:

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

We will show that $A_1 \leqslant_{H \cap H'} A_3$ by exhibiting R_{13} as a hole-tracking simulation of A_1 by A_3.

We have to prove that the relation R_{13} satisfies the three conditions of the definition of a refinement of open automata.

1. Firstly, we have to R_{13} satisfies initial configurations:

   $$\sigma_{01} \uplus \sigma_{03} \vdash R_{13}(s_{01}, s_{03})$$

   By knowing that substitutions only have an effect on the variables of the open automaton they belong to, they also produce terms containing only variables of the open automaton they belong to. We have:

   $$\begin{aligned}
   (\sigma_{01} \uplus \sigma_{02} \vdash R_{12}(s_{01}, s_{02})) \wedge (\sigma_{02} \uplus \sigma_{03} \vdash R_{23}(s_{02}, s_{03})) &\implies \\
   R_{12}(s_{01}, s_{02})\{\!\{\sigma_{01} \uplus \sigma_{02}\}\!\} \wedge R_{23}(s_{02}, s_{03})\{\!\{\sigma_{02} \uplus \sigma_{03}\}\!\} &\implies \\
   R_{12}(s_{01}, s_{02})\{\!\{\sigma_{01} \uplus \sigma_{02} \uplus \sigma_{03}\}\!\} \wedge R_{23}(s_{02}, s_{03})\{\!\{\sigma_{01} \uplus \sigma_{02} \uplus \sigma_{03}\}\!\} &\implies \\
   \underbrace{R_{12}(s_{01}, s_{02}) \wedge R_{23}(s_{02}, s_{03})}_{\implies R_{13}(s_{01}, s_{03})}\{\!\{\sigma_{01} \uplus \sigma_{02} \uplus \sigma_{03}\}\!\} &
   \end{aligned}$$

   Since σ_{02} has no effect on variables of s_{01} and s_{03} thus we get the expected result.
2. Secondly, we need to prove that for any open transition t_1 in T_1 originating from s_1:

   $$\frac{\beta_{1j}^{j \in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in \mathrm{OT}(s_1)$$

   there exists an indexed family of OTs originating from s_3:

   $$V_1 \uplus V_3 \uplus \mathit{vars}(t_1) \vdash \left(R_{13}(s_1, s_3) \wedge g_1 \implies \bigvee_{z \in Z} \left(\begin{array}{c} \alpha_1 = \alpha_{3z} \wedge \bigwedge\limits_{j \in J'_{3z} \cap (H \cap H')} \beta_{1j} = \beta_{3jz} \wedge \\ g_{3z} \wedge R_{12}(s_1', s_{3z}')\{\!\{\psi_1 \uplus \psi_{3z}\}\!\} \end{array} \right) \right)$$

Consider $(s_1, s_3) \in R_{13}$ then there is a set of states $(s_{2p})^{p\in P}$ of A_2 relating s_1 and s_3:

$$R_{13}(s_1, s_3) = \bigvee_{p\in P} (R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3))$$

Let's consider any $s_{2p} \in (s_{2p})^{p\in P}$. We have on one side, for any open transition t_1 in T_1 originating from s_1:

$$\frac{\beta_{1j}^{j\in J'_1}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s'_1} \in \mathrm{OT}(s_1)$$

there exists an indexed family of OTs originating from s_{2p}:

$$\left(\frac{\beta_{2pxj}^{j\in J'_{2px}}, g_{2px}, \psi_{2px}}{s_{2p} \xrightarrow{\alpha_{2px}} s'_{2px}} \in \mathrm{OT}(s_{2p})\right)^{x\in X_p}$$

such that $\forall x \in X, J'_{2px} \cap H = J'_1 \cap H$ and

$$V_1 \uplus V_2 \uplus \mathit{vars}(t_1) \vdash$$
$$\left(R_{12}(s_1, s_{2p}) \wedge g_1 \implies \bigvee_{x\in X_p} \left(\begin{array}{c} \alpha_1 = \alpha_{2px} \wedge \bigwedge_{j\in J'_{2px}\cap H} \beta_{1j} = \beta_{2pxj} \wedge \\ g_{2px} \wedge R_{12}\left(s'_1, s'_{2px}\right)\{\!\{\psi_1 \uplus \psi_{2px}\}\!\} \end{array}\right)\right)$$

Adding to both sides of the implication the predicate $R_{23}(s_{2p}, s_3)$ we get:

$$V_1 \uplus V_2 \uplus V_3 \uplus \mathit{vars}(t_1) \vdash$$
$$R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge g_1 \implies$$
$$\bigvee_{x\in X} \left(\begin{array}{c} \alpha_1 = \alpha_{2px} \wedge \bigwedge_{j\in J'_{2px}\cap H} \beta_{1j} = \beta_{2pxj} \wedge \\ g_{2px} \wedge R_{12}\left(s'_1, s'_{2px}\right)\{\!\{\psi_1 \uplus \psi_{2px}\}\!\} \wedge R_{23}(s_{2p}, s_3) \end{array}\right) \quad (*)$$

On the other side, according the relation between A_2 and A_3 we have for any open transition t_{2px} in T_2 originating from s_{2p}:

$$\frac{\beta_{2pxj}^{j\in J'_{2px}}, g_{2px}, \psi_{2px}}{s_{2p} \xrightarrow{\alpha_{2px}} s'_{2px}} \in \mathrm{OT}(s_{2p})$$

there exists an indexed family of OTs originating from s_3:

$$\left(\frac{\beta_{3pxyj}^{j\in J'_{3pxy}}, g_{3pxy}, \psi_{3pxy}}{s_3 \xrightarrow{\alpha_{3pxy}} s'_{3pxy}} \in \mathrm{OT}(s_3)\right)^{y\in Y}$$

such that $\forall y \in Y, J'_{2px} \cap H' = J'_{3pxy} \cap H'$ and

$$V_2 \uplus V_3 \uplus vars(t_{2px}) \vdash$$
$$R_{23}(s_{2p}, s_3) \wedge g_{2px} \implies$$
$$\bigvee_{y \in Y} \begin{pmatrix} \alpha_{2px} = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap H'} \beta_{2pxj} = \beta_{3pxyj} \wedge \\ g_{3pxy} \wedge R_{23}\big(s'_{2px}, s'_{3xy}\big)\{\!\{\psi_{2px} \uplus \psi_{3pxy}\}\!\} \end{pmatrix} \qquad (**)$$

From the two previous cases: $\forall x \in X, J'_{2px} \cap H = J'_1 \cap H$ and $\forall y \in Y, J'_{2px} \cap H' = J'_{3pxy} \cap H'$ we conclude: $\forall x \in X, \forall y \in Y, J'_1 \cap (H \cap H') = J'_{3pxy} \cap (H \cap H')$

In addition, by combining formula $(**)$ and $(*)$, we get:

$$V_1 \uplus V_2 \uplus V_3 \uplus vars(t_1) \uplus vars(t_{2px}) \vdash$$
$$R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge g_1 \implies$$
$$\bigvee_{x \in X} \left(\begin{array}{l} \alpha_1 = \alpha_{2px} \wedge \bigwedge_{j \in J'_{2px} \cap H} \beta_{1j} = \beta_{2pxj} \wedge R_{12}\big(s'_1, s'_{2px}\big)\{\!\{\psi_1 \uplus \psi_{2px}\}\!\} \\ \wedge \bigvee_{y \in Y} \begin{pmatrix} \alpha_{2px} = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap H'} \beta_{2pxj} = \beta_{3pxyj} \wedge \\ g_{3pxy} \wedge R_{23}\big(s'_{2px}, s'_{3pxy}\big)\{\!\{\psi_{2px} \uplus \psi_{3pxy}\}\!\} \end{pmatrix} \end{array} \right)$$

With the rearrangement of the formula, we get:

$$V_1 \uplus V_2 \uplus V_3 \uplus vars(t_1) \uplus vars(t_{2px}) \vdash$$
$$R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge g_1 \implies$$
$$\bigvee_{x \in X} \bigvee_{y \in Y} \begin{pmatrix} \alpha_1 = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap (H \cap H')} \beta_{1j} = \beta_{3jpxy} \wedge g_{3pxy} \wedge \\ R_{12}\big(s'_1, s'_{2px}\big)\{\!\{\psi_1 \uplus \psi_{2px}\}\!\} \wedge R_{23}\big(s'_{2px}, s'_{3pxy}\big)\{\!\{\psi_{2px} \uplus \psi_{3pxy}\}\!\} \end{pmatrix}$$

Because of the domain of the substitutions of the relations, the formula can be re-written:

$$V_1 \uplus V_2 \uplus V_3 \uplus vars(t_1) \uplus vars(t_{2px}) \vdash$$
$$R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge g_1 \implies$$
$$\bigvee_{x \in X} \bigvee_{y \in Y} \begin{pmatrix} \alpha_1 = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap (H \cap H')} \beta_{1j} = \beta_{3jpxy} \wedge g_{3pxy} \wedge \\ \big(R_{12}\big(s'_1, s'_{2px}\big) \wedge R_{23}\big(s'_{2px}, s'_{3pxy}\big)\big)\{\!\{\psi_1 \uplus \psi_{2px} \uplus \psi_{3pxy}\}\!\} \end{pmatrix}$$

The formula is valid for all $s_{2p} \in (s_{2p})^{p \in P}$. To build R_{13} we need to rely on the disjunction of all possible paths to relate s_1 and s_3, which leads to $R_{13}(s_1, s_3) = \bigvee_{p \in P} (R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3))$.

$$V_1 \uplus V_2 \uplus V_3 \uplus vars(t_1) \uplus \biguplus_{p \in P} vars(t_{2px}) \vdash$$

$$\bigvee_{p \in P} (R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge g_1) \implies$$

$$\bigvee_{p \in P} \bigvee_{x \in X} \bigvee_{y \in Y} \left(\begin{array}{l} \alpha_1 = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap (H \cap H')} \beta_{1j} = \beta_{3jpxy} \wedge g_{3pxy} \wedge \\ \underbrace{R_{12}(s'_1, s'_{2px}) \wedge R_{23}(s'_{2px}, s'_{3pxy})}_{\implies R_{13}(s'_1, s'_{3pxy})} \{\!\{\psi_1 \uplus \psi_{2px} \uplus \psi_{3pxy}\}\!\} \end{array} \right)$$

Indeed, since ψ_{2px} has no effect on variables of A_1 and A_3 we have:

$$\begin{aligned} R_{12}(s'_1, s'_{2px}) \wedge R_{23}(s'_{2px}, s'_{3pxy}) \{\!\{\psi_1 \uplus \psi_{2px} \uplus \psi_{3pxy}\}\!\} \\ \implies R_{13}(s'_1, s'_{3pxy}) \{\!\{\psi_1 \uplus \psi_{2px} \uplus \psi_{3pxy}\}\!\} \\ \implies R_{13}(s'_1, s'_{3pxy}) \{\!\{\psi_1 \uplus \psi_{3pxy}\}\!\} \end{aligned}$$

This allows us to conclude that the family of open transitions of A_3 indexed over p, x, and y simulates the original open transition with note that ψ_{2px} has no effect on variables of A_1 and A_3:

$$V_1 \uplus V_3 \uplus vars(t_1) \vdash$$

$$R_{13}(s_1, s_3) \wedge g_1 \implies$$

$$\bigvee_{p \in P} \bigvee_{x \in X} \bigvee_{y \in Y} \left(\begin{array}{l} \alpha_1 = \alpha_{3pxy} \wedge \bigwedge_{j \in J'_{3pxy} \cap (H \cap H')} \beta_{1j} = \beta_{3jpxy} \\ \wedge g_{3pxy} \wedge R_{13}(s'_1, s'_{3pxy}) \{\!\{\psi_1 \uplus \psi_{3pxy}\}\!\} \end{array} \right)$$

Overall, we have a family of open transitions $t_{pxy}^{p \in P, x \in X, y \in Y} \subseteq T_3$ that should simulate t_1. All combinations of elements in P, X, and Y provide a set Z of open transitions. This allows us to get the desired conclusion, that there is a set of open transitions indexed over Z:

$$V_1 \uplus V_3 \uplus vars(t_1) \vdash$$

$$\left(R_{13}(s_1, s_3) \wedge g_1 \implies \bigvee_{z \in Z} \left(\begin{array}{l} \alpha_1 = \alpha_{3z} \wedge \bigwedge_{j \in J'_{3z} \cap (H \cap H')} \beta_{1j} = \beta_{3jz} \wedge \\ g_{3z} \wedge R_{13}(s'_1, s'_{3z}) \{\!\{\psi_1 \uplus \psi_{3z}\}\!\} \end{array} \right) \right)$$

3. Finally, we need to prove the satisfaction of the deadlock reducing condition. According to the refinement relation relating A_2 and A_3 we have for all $(s_{2p}, s_3) \in S_2 \times S_3$:

$$V_2 \uplus V_3 \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash \left(R_{23}(s_{2p}, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p}) \right)$$

By adding the term $R_{12}(s_1, s_{2p})$ to both sides of the implication, we get:

$$V_1 \uplus V_2 \uplus V_3 \uplus vars(t_{2p}) \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash$$
$$R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies R_{12}(s_1, s_{2p}) \wedge \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p})$$

This is valid for all $s_{2p} \in (s_{2p})^{p \in P}$ then we have:

$$V_1 \uplus V_2 \uplus V_3 \uplus \biguplus_{p \in P} vars(t_{2p}) \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash$$
$$\bigvee_{p \in P} \left(R_{12}(s_1, s_{2p}) \wedge R_{23}(s_{2p}, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies R_{12}(s_1, s_{2p}) \wedge \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p}) \right)$$

This results in:

$$V_1 \uplus V_2 \uplus V_3 \uplus \biguplus_{p \in P} vars(t_{2p}) \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash$$
$$R_{13}(s_1, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies \bigvee_{p \in P} \left(R_{12}(s_1, s_{2p}) \wedge \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p}) \right)$$

Yet according to the relation between A_1 and A_2 we have for all $(s_1, s_{2p}) \in S_1 \times S_2$:

$$V_1 \uplus V_2 \uplus \biguplus_{t_{2p} \in \mathrm{OT}(s_{2p})} vars(t_{2p}) \vdash R_{12}(s_1, s_{2p}) \wedge \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p}) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1)$$

This formula is valid for all $s_{2p} \in (s_{2p})^{p \in P}$, thus we can state the following:

$$V_1 \uplus V_2 \uplus \biguplus_{\substack{t_{2p} \in \mathrm{OT}(s_{2p}) \\ p \in P}} vars(t_{2p}) \vdash \bigvee_{p \in P} \left(R_{12}(s_1, s_{2p}) \wedge \bigvee_{t_{2p} \in \mathrm{OT}(s_{2p})} guard(t_{2p}) \right) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1)$$

By using transitive property of implication on formula obtained form both relations and by removing the A_2 variables that have not effect on A_1 et A_3, we get the desired result:

$$V_1 \uplus V_3 \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash R_{13}(s_1, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1)$$

□

B Proof of Context Refinement (Theorem 2)

Suppose that $A_1 \leqslant_H A_2$, $k \in H$ and that $A_1[A_3/k]$ is non-blocking. We have:

$$A_1[A_3/k] \leqslant_{J_3 \uplus H \setminus \{k\}} A_2[A_3/k]$$

[TODO:The subtlety here is that the original relation implies that valuations in A3 are equal (ie the value for each variable are the same in both valuations, modulo renaming), after a transition we should obtain "equal" valuations because Post are deterministic]

Proof. Let us denote by A_{13} (resp. A_{23}) the open automaton resulting from $A_1[A_3/k]$ (resp. $A_2[A_3/k]$), to prove the theorem it is sufficient to prove that there exists a relation between states of the two open automata that satisfies the conditions of the Definition 10.

We denote $A_1 = \langle S_1, s_{01}, J_1, V_1, \sigma_{01}, T_1 \rangle$ and $A_2 \coloneqq \langle S_2, s_{02}, J_2, V_2, \sigma_{02}, T_2 \rangle$ and $A_3 = \langle S_3, s_{03}, J_3, V_3, \sigma_{03}, T_3 \rangle$. The proof requires to rename the variables of one instance of the two A_3 automata to avoid clashes in variable names (this is required by the definition of refinement). In practice we will use superscripts ^{1} and ^{2} to distinguish elements of the two instances of A_3.

Let R be the refinement relation relating states of A_1 and A_2. Let us denote with t^1 and t^2 the elements of A_1 and A_2 respectively. Consider any two states $s_{13} = \left(s_1, s_3^1\right)$ and $s_{23} = \left(s_2, s_3^2\right)$ (s_3^1 and s_3^2 are the same with renaming). We define a relation R' relating states of s_{13} and s_{23} as follows:

$$R'(s_{13}, s_{23}) = R(s_1, s_2) \wedge \checkmark_{A_{13}}(s_{13}) \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2$$

We want to prove that $(R', H \uplus J_3 \setminus \{k\})$ is a hole-tracking simulation of A_{13} and A_{23}. In the following we denote $H' = H \cup J_3 \setminus \{k\}$.

1. First, we have to prove the relation for initial states:

$$\sigma_{013} \uplus \sigma_{023} \vdash R'(s_{013}, s_{023})$$

with $\sigma_{013} = \sigma_{01} \uplus \sigma_{03}^1$, $\sigma_{023} = \sigma_{02} \uplus \sigma_{03}^2$, $s_{013} = (s_{01}, s_{03}^1)$, and $s_{023} = (s_{02}, s_{02}^2)$.
By using the fact that R relates initial configurations of A_1 and A_2, we

have the following statement: $(\sigma_{01} \uplus \sigma_{02} \vdash R(s_{01}, s_{02}))$, and based on the fact that initial states are reachable $\sigma_{013} \vdash \checkmark_{A_{13}}(s_{013})$ we have

$$(\sigma_{01} \uplus \sigma_{02} \vdash R(s_{01}, s_{02})) \land (\sigma_{013} \vdash \checkmark_{A_{13}}(s_{013}))$$

Considering that initial valuations σ^1_{03} and σ^2_{03} associate the same values to the "same" variables modulo renaming, so the following holds:
$\left(\bigwedge_{v_3 \in V_3} v^1_3 = v^2_3 \land s^1_3 = s^2_3\right) \{\!\{\sigma^1_{03} \uplus \sigma^2_{03}\}\!\}$.
Additionally, because the domains of the substitution function are disjoint, the substitution function has an effect only on the related elements, we get:

$$\begin{aligned}
&(\sigma_{01} \uplus \sigma_{02} \vdash R(s_{01}, s_{02})) \land (\sigma_{013} \vdash \checkmark_{A_{13}}(s_{013})) \\
\Longrightarrow\ & R(s_{01}, s_{02})\{\!\{\sigma_{01} \uplus \sigma_{02}\}\!\} \land \checkmark_{A_{13}}(s_{013})\{\!\{\sigma_{013}\}\!\} \land \left(\bigwedge_{v_3 \in V_3} v^1_3 = v^2_3 \land s^1_3 = s^2_3\right) \{\!\{\sigma^1_{03} \uplus \sigma^2_{03}\}\!\} \\
\Longrightarrow\ & \left(R(s_{01}, s_{02}) \land \checkmark_{A_{13}}(s_{013}) \land \bigwedge_{v_3 \in V_3} v^1_3 = v^2_3 \land s^1_3 = s^2_3\right) \{\!\{\sigma_{01} \uplus \sigma_{02} \uplus \sigma^1_{03} \uplus \sigma^2_{03} \uplus \sigma_{013}\}\!\} \\
\Longrightarrow\ & \sigma_{013} \uplus \sigma_{023} \vdash R'(s_{013}, s_{023})
\end{aligned}$$

2. Second, we need to prove for any OT t_{13} in T_{13} originating from s_{13}:

$$\frac{\beta^{j \in J'_{13}}_{13j}, g_{13}, \psi_{13}}{s_{13} \xrightarrow{\alpha_{13}} s'_{13}} \in \mathrm{OT}(s_{13})$$

there exists an indexed family t_{23x} of OTs originating from s_{23} that simulate it:

$$\left(\frac{\beta^{j \in J'_{23x}}_{23xj}, g_{23x}, \psi_{23x}}{s_{23} \xrightarrow{\alpha_{23x}} s'_{23x}} \in \mathrm{OT}(s_{23})\right)^{x \in X}$$

such that $\left(\forall x \in X, J'_{23x} \cap H' = J'_{13} \cap H'\right)$ and

$$V_{13} \uplus V_{23} \uplus \mathit{vars}(t_{13}) \vdash R'(s_{13}, s_{23}) \land g_{13} \implies \bigvee_{x \in X} \begin{pmatrix} \alpha_{13} = \alpha_{23x} \land \bigwedge_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \land \\ g_{23x} \land R'\left(s'_{13}, s'_{23x}\right)\{\!\{\psi_{13} \uplus \psi_{23x}\}\!\} \end{pmatrix}$$

Recall that by definition of composition and open automata refinement we have:

$$\begin{aligned}
&V_{13} = V_1 \uplus V^1_3 \text{ and } V_{23} = V_2 \uplus V^2_3 \\
&H' \subseteq J_3 \uplus (J_1 \cap J_2) \setminus \{k\} = (J_3 \uplus J_1 \setminus \{k\}) \cap (J_3 \uplus J_2 \setminus \{k\})
\end{aligned}$$

First of all, we have by hypothesis $A_1 \leqslant_H A_2$, then for any open transition t_1 in T_1 originating from s_1:

$$\frac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'} \in \mathrm{OT}(s_1)$$

there exists an indexed family of OTs originating from s_2:

$$\left(\frac{\beta_{2xj}^{j\in J_{2x}'}, g_{2x}, \psi_{2x}}{s_2 \xrightarrow{\alpha_{2x}} s_{2x}'} \in \mathrm{OT}(s_2)\right)^{x\in X}$$

such that $\forall x \in X, J_{2x}' \cap H = J_1' \cap H$ and

$$V_1 \uplus V_2 \uplus vars(t_1) \vdash$$
$$\left(R(s_1,s_2) \wedge g_1 \implies \bigvee_{x\in X}\left(\begin{array}{c}\alpha_1 = \alpha_{2x} \wedge \bigwedge\limits_{j\in J_{2x}'\cap H} \beta_{1j} = \beta_{2xj} \wedge \\ g_{2x} \wedge R\big(s_1', s_{2x}'\big)\{\!\{\psi_1 \uplus \psi_{2x}\}\!\}\end{array}\right)\right) \qquad (*)$$

Consider any transition t_{13} in A_{13}. Based on the definition of composition t_{13} can be obtained from two different cases, we will consider the two cases separately.

First case: Both automata perform a transition. The transition t_{13} is obtained by the composition of transitions $t_1 = \frac{\beta_{1j}^{j\in J_1'}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s_1'}$ and

$$t_3^1 = \frac{(\beta_{3j}^1)^{j\in J_3^{1\prime}}, g_3^1, \psi_3^1}{s_3^1 \xrightarrow{\alpha_3^1} s_3^{1\prime}} \qquad \text{when } k \in J_1'$$

The result is:

$$t_{13} = \frac{\beta_{1j}^{j\in J_1'\setminus\{k\}} \uplus (\beta_{3j}^1)^{j\in J_3^{1\prime}}, g_1 \wedge g_3^1 \wedge \alpha_3^1 = \beta_{1k}, \psi_1 \uplus \psi_3^1}{(s_1, s_3^1) \xrightarrow{\alpha_1} \left(s_1', s_3^{1\prime}\right)} \qquad \text{where } k \in J_1'$$

We then obtain a family of OTs by the simulation of A_1 by A_2 (as stated above). By hypothesis we have $k \in H$, so in the case where $k \in J_1'$, we deduce that $k \in J_{2x}'$ we can then build a family of OTs $t_{23x}^{x\in X}$ with the same transitions of A_3 (up to renaming) as those used to build t_{13}.

$$t_{23x} = \left(\frac{\beta_{2xj}^{j\in J_{2x}'\setminus\{k\}} \uplus (\beta_{3j}^2)^{j\in J_3^{2\prime}}, g_{2x} \wedge g_3^2 \wedge \alpha_3^2 = \beta_{2xk}, \psi_{2x} \uplus \psi_3^2}{(s_2, s_3^2) \xrightarrow{\alpha_{2x}} \left(s_{2x}', s_3^{2\prime}\right)}\right)^{x\in X}$$

Recall that in this case $k \in J'_1$, so $\forall x \in X$ we have

$$\begin{aligned}
J'_{23x} \cap H' &= ((J'_{2x} \setminus \{k\}) \uplus J_3^{2\prime}) \cap (H \uplus J_3 \setminus \{k\}) \\
&= ((J'_{2x} \cap (H \uplus J_3)) \uplus (J_3^{2\prime} \cap (H \uplus J_3))) \setminus \{k\} \\
&= ((J'_{2x} \cap H) \uplus (J_3^{2\prime} \cap J_3)) \setminus \{k\} \text{ since } J_3 \cap J'_{2x} = \varnothing \text{ and } H \cap J_3^{2\prime} = \varnothing \\
&= ((J'_{2x} \cap H) \uplus J_3^{2\prime}) \setminus \{k\} \text{ since } J_3^{2\prime} \subseteq J_3 \\
&= ((J'_1 \cap H) \uplus J_3^{1\prime}) \setminus \{k\} \quad \text{since } J_3^{1\prime} = J_3^{2\prime} \text{ and } J'_1 \cap H = J'_{2x} \cap H \\
&= ((J'_1 \cap H) \uplus (J_3^{1\prime} \cap J_3)) \setminus \{k\} \text{ since } J_3^{1\prime} \subseteq J_3 \\
&= (J'_1 \cap (J_3 \uplus H)) \uplus ((J_3^{1\prime} \cap (J_3 \uplus H)) \setminus \{k\} \text{ since } J_3 \cap J'_1 = \varnothing \text{ and } H \cap J_3^{1\prime} = \varnothing \\
&= \big((J'_1 \uplus J_3^{1\prime}) \setminus \{k\}\big) \cap ((J_3 \uplus H) \setminus \{k\}) \\
&= J'_{13} \cap H'
\end{aligned}$$

In this case the composition gives:

$$g_{13} \Leftrightarrow g_1 \wedge g_3^1 \wedge \alpha_3^1 = \beta_{1k} \text{ and } g_{23x} \Leftrightarrow g_{2x} \wedge g_3^2 \wedge \alpha_3^2 = \beta_{2xk}$$

As $k \in H$ we have $\beta_{1k} = \beta_{2xk}$ then we deduce:

$$g_3^1 \wedge \alpha_3^1 = \beta_{1k} \Leftrightarrow g_3^2 \wedge \alpha_3^2 = \beta_{2xk}$$

The proof of the rest is based on the following facts:

(a) Because composition doesn't change the resulting actions, nor their variables, we can extend the valuation context of the variables to cover the variables of the transition t_3. By construction of t_{13} and t_{23} we have $\alpha_{13} = \alpha_1$ and $\alpha_{23x} = \alpha_{2x}$. So we deduce: $\alpha_1 = \alpha_{2x} \Rightarrow \alpha_{13} = \alpha_{23x}$.

(b) By composition we have also:
$\beta_{13j}^{j \in J'_{13}} = \beta_{1j}^{j \in J'_1 \setminus \{k\}} \uplus (\beta_{3j}^1)^{j \in J_3^{1\prime}}$ and $\beta_{23xj}^{j \in J'_{23}} = \beta_{2xj}^{j \in J'_{2x} \setminus \{k\}} \uplus (\beta_{3j}^2)^{j \in J_3^{2\prime}}$
Therefore, we have for all $j \in J'_{13}$ (recall that $J'_{13} = J'_{23}$):

$$\beta_{13j} = \beta_{23xj} \Rightarrow (j \in J_1 \wedge \beta_{1j} = \beta_{2xj}) \vee (j \in J_3^1 \wedge \beta_{3j}^1 = \beta_{3j}^2)$$

(c) Considering β_{3j}^1 and β_{3j}^2 are the same (up to renaming) we have:
$V_3^1 \uplus V_3^2 \uplus vars(t_{13}) \vdash \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \implies \bigwedge_{j \in J_3^{2\prime}} \beta_{3j}^1 = \beta_{3j}^2$
The disjunction with the following hypothesis:
$V_1 \uplus V_2 \uplus vars(t_1) \vdash \bigwedge_{j \in J'_{2x} \cap H} \beta_{1j} = \beta_{2xj}$ will give:

$$\begin{aligned}
V_1 \uplus V_2 \uplus V_3^1 \uplus V_3^2 \uplus vars(t_{13}) \vdash \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 &\implies \bigwedge_{j \in J'_{2x} \cap H} \beta_{1j} = \beta_{2xj} \vee \bigwedge_{j \in J_3^{2\prime}} \beta_{3j}^1 = \beta_{3j}^2 \\
\Rightarrow V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 &\implies \bigwedge_{j \in (J'_{2x} \cap H) \uplus J_3^{2\prime}} \beta_{13j} = \beta_{23xj} \\
\Rightarrow V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 &\implies \bigwedge_{j \in \Rightarrow ((J'_{2x} \cap H) \uplus J_3^{2\prime}) \setminus \{k\}} \beta_{13j} = \beta_{23xj} \\
\Rightarrow V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 &\implies \bigwedge_{j \in (J'_{23x} \cap H')} \beta_{13j} = \beta_{23xj}
\end{aligned}$$

By the extension of the valuation context mentioned above in the formula $(*)$ and by using the statements resulting from the cases (a), (b) and (c), we get:

$$V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash$$
$$R(s_1, s_2) \wedge \big(g_1 \wedge g_3^1 \wedge \alpha_3^1 = \beta_{1k}\big) \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \implies$$
$$\bigvee_{x \in X} \left(\begin{array}{c} \alpha_{13} = \alpha_{23x} \wedge \bigwedge\limits_{j \in (J'_{23x} \cap H')} \beta_{13j} = \beta_{23xj} \\ \wedge\ g_{2x} \wedge R\big(s'_1, s'_{2x}\big) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \end{array} \right) \wedge g_3^2 \wedge \alpha_3^2 = \beta_{2xk}$$

That can be re-written as follows:

$$V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash$$
$$R(s_1, s_2) \wedge g_{13} \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \implies \bigvee_{x \in X} \left(\begin{array}{c} \alpha_{13} = \alpha_{23x} \wedge \bigwedge\limits_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \\ \wedge\ g_{23x} \wedge R\big(s'_1, s'_{2x}\big) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \end{array} \right)$$

Moreover, we have for any transition t_3 in A_3 relying s_3 and s'_3 the following:

$$V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash$$
$$\bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2 \implies \bigwedge_{v_3 \in V_3} \psi_{13}(v_3^1) = \psi_{13}(v_3^2) \wedge s_3^{1\prime} = s_3^{2\prime}$$

From the two previous formulas, we get:

$$V_{13} \uplus V_{23} \uplus vars(t_{13}) \vdash$$
$$R(s_1, s_2) \wedge g_{13} \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2 \implies$$
$$\bigvee_{x \in X} \left(\begin{array}{l} \alpha_{13} = \alpha_{23x} \wedge \bigwedge\limits_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \wedge\ g_{23x} \wedge \\ R\big(s'_1, s'_{2x}\big) \{\!\{ \psi_1 \uplus \psi_{2x} \}\!\} \wedge \bigwedge\limits_{v_3 \in V_3} \psi_{13}(v_3^1) = \psi_{13}(v_3^2) \wedge s_3^{1\prime} = s_3^{2\prime} \end{array} \right)$$

And finally by hypothesis we have $A_1[A_3/k]$ is non-blocking, by definition of reachability (Definition 7) we obtain $\sigma_{013} \vdash \checkmark_{A_{13}}(s_{013})$ and for all $t_{13} \in T_{13}$:

$$vars(t_{13}) \vdash \big(\checkmark_{A_{13}}(s_{13}) \wedge g_{13} \implies \checkmark_{A_{13}}\big(s'_{13}\big) \{\!\{ \psi_{13} \}\!\}\big)$$

Thus, we get:

$$
\begin{array}{l}
V_{13} \uplus V_{23} \uplus \mathit{vars}(t_{13}) \vdash \\
\quad R(s_1, s_2) \wedge g_{13} \wedge \checkmark_{A_{13}}(s_{13}) \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2 \implies \\
\bigvee_{x \in X} \left(\begin{array}{l} \alpha_{13} = \alpha_{23x} \wedge \bigwedge_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \wedge g_{23x} \wedge R\big(s'_1, s'_{2x}\big)\{\!\{\psi_1 \uplus \psi_{2x}\}\!\} \\ \wedge \checkmark_{A_{13}}\big(s'_{13}\big)\{\!\{\psi_{13}\}\!\} \wedge \bigwedge_{v_3 \in V_3} \psi_{13}(v_3^1) = \psi_{13}(v_3^2) \wedge s_3^{1\prime} = s_3^{2\prime} \end{array} \right)
\end{array}
$$

Because of the independence of the substitution domains, we simplify and get the expected formula:

$$
\begin{array}{l}
V_{13} \uplus V_{23} \uplus \mathit{vars}(t_{13}) \vdash \\
\quad R'(s_{13}, s_{23}) \wedge g_{13} \implies \bigvee_{x \in X} \left(\begin{array}{l} \alpha_{13} = \alpha_{23x} \wedge \bigwedge_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \\ \wedge\ g_{23x} \wedge R'\big(s'_{13}, s'_{23x}\big)\{\!\{\psi_{13} \uplus \psi_{23x}\}\!\} \end{array} \right)
\end{array}
$$

Second case: Only the encompassing automaton performs a transition t_{13} is obtained by the transition $t_1 = \dfrac{\beta_{1j}^{j \in J'_1}, g_1, \psi_1}{s_1 \xrightarrow{\alpha_1} s'_1}$ alone with the state s_3^1 unchanged, if $k \notin J'_1$

$$
t_{13} = \frac{\beta_{1j}^{j \in J'_1}, g_1, \psi_1}{\big(s_1, s_3^1\big) \xrightarrow{\alpha_1} \big(s'_1, s_3^1\big)}
$$

We then obtain a family of OTs by the simulation of A_1 by A_2 (stated above). As $k \in H$ we have $k \notin J'_{2x}$ and we build:

$$
t_{23x} = \left(\frac{\beta_{2xj}^{j \in J'_{2x}}, g_{2x}, \psi_{2x}}{\big(s_2, s_3^2\big) \xrightarrow{\alpha_{2x}} \big(s'_{2x}, s_3^2\big)} \right)^{x \in X}
$$

These two cases define (depending on how t_{13} is built) the family t_{23x} of OTs we are looking for. Thus, for both cases we have to prove the following:

$$
\begin{array}{l}
\big(\forall x \in X, J'_{23x} \cap H' = J'_{13} \cap H'\big) \text{ and} \\
V_{13} \uplus V_{23} \uplus \mathit{vars}(t_{13}) \vdash \\
R'(s_{13}, s_{23}) \wedge g_{13} \implies \bigvee_{x \in X} \left(\begin{array}{l} \alpha_{13} = \alpha_{23x} \wedge \bigwedge_{j \in J'_{23x} \cap H'} \beta_{13j} = \beta_{23xj} \\ \wedge\ g_{23x} \wedge R'\big(s'_{13}, s'_{23x}\big)\{\!\{\psi_{13} \uplus \psi_{23x}\}\!\} \end{array} \right)
\end{array}
$$

Recall in this case $k \notin J'_1$, $\forall x \in X$ we have

$$\begin{aligned} J'_{23x} \cap H' &= J'_{2x} \cap (J_3 \uplus H \setminus \{k\}) \\ &= \left(J'_{2x} \cap H\right) \text{ since } J'_{2x} \cap J_3 = \varnothing \wedge k \notin J'_{2x} \\ &= \left(J'_1 \cap H\right) \text{ since } J'_1 \cap H = J'_{2x} \cap H \\ &= \left(J'_{13} \cap H'\right) \text{ since } J_3 \cap J'_1 = \varnothing \wedge k \notin J'_1 \end{aligned}$$

The proof of the rest of the formula follows the same steps as the previous case the only argument that changes is that by composition we obtain: $g_{13} \Leftrightarrow g_1$ and $g_{23x} \Leftrightarrow g_{2x}$.
The last formula we need to prove is the following: for all $(s_{13}, s_{23}) \in S_{13} \times S_{23}$

$$V_{13} \uplus V_{23} \uplus \biguplus_{t_{23} \in \mathrm{OT}(s_{23})} vars(t_{23}) \vdash \left(R'(s_{13}, s_{23}) \wedge \bigvee_{t_{23} \in \mathrm{OT}(s_{23})} guard(t_{23}) \implies \bigvee_{t_1 \in \mathrm{OT}(s_{13})} guard(t_{13}) \right)$$

Let's start with the hypothesis of the theorem stating that $A_1 \leqslant_H A_2$, thus we have for all $(s_1, s_2) \in S_1 \times S_2$

$$V_1 \uplus V_2 \uplus \biguplus_{t_2 \in \mathrm{OT}(s_2)} vars(t_2) \vdash \left(R(s_1, s_2) \wedge \bigvee_{t_2 \in \mathrm{OT}(s_2)} guard(t_2) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \right)$$

We extend the valuation to cover the variables of t_3^2 in t_{23} and using their value in t_3 and because the following statement holds: $vars(t_{13}) \vdash \checkmark_{A_{13}}(s_{13})$, we get:

$$V_1 \uplus V_2 \uplus \biguplus_{t_{23} \in \mathrm{OT}(s_{23})} vars(t_{23}) \vdash$$

$$\left(R(s_1, s_2) \wedge \bigvee_{t_{23} \in \mathrm{OT}(s_{23})} guard(t_{23}) \wedge \checkmark_{A_{13}}(s_{13}) \implies \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \wedge \checkmark_{A_{13}}(s_{13}) \right)$$

We have a second hypothesis stating that the result of the composition $A_1[A_3/k]$ is non-blocking, by Definition 8 we have:

$$V_{13} \uplus \biguplus_{t_1 \in \mathrm{OT}(s_1)} vars(t_1) \vdash \left(\checkmark_{A_{13}}(s_{13}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \implies \bigvee_{t_{13} \in \mathrm{OT}(s_{13})} guard(t_{13}) \right)$$

By transitivity we obtain:

$$V_1 \uplus V_2 \uplus \biguplus_{t_{23} \in \mathrm{OT}(s_{23})} vars(t_{23}) \vdash \left(R(s_1, s_2) \wedge \checkmark_{A_{13}}(s_{13}) \wedge \bigvee_{t_{23} \in \mathrm{OT}(s_{23})} guard(t_{23}) \implies \bigvee_{t_{13} \in \mathrm{OT}(s_{13})} guard(t_{13}) \right)$$

Considering that each valuations σ_3^1 and σ_3^2 associate the same values to the "same" variables modulo renaming, so the following holds:
$\left(\bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2\right) \{\!\{\sigma_3^1 \uplus \sigma_3^2\}\!\}$, we obtain:

$$V_1 \uplus V_2 \uplus \biguplus_{t_{23} \in \mathrm{OT}(s_{23})} vars(t_{23}) \vdash$$

$$\left(\underbrace{R(s_1, s_2) \wedge \checkmark_{A_{13}}(s_{13}) \wedge \bigwedge_{v_3 \in V_3} v_3^1 = v_3^2 \wedge s_3^1 = s_3^2}_{R'(s_{13}, s_{23})} \wedge \bigvee_{t_{23} \in \mathrm{OT}(s_{23})} guard(t_{23}) \implies \bigvee_{t_{13} \in \mathrm{OT}(s_{13})} guard(t_{13}) \right)$$

which gives us what we needed to demonstrate. □

C Proof of Congruence (Theorem 4)

Suppose that $A_2 \leqslant_H A_3$, $k \in H$ and that $A_1[A_2/k]$ is non-blocking. We have:

$$A_1[A_2/k] \leqslant_{J_1 \uplus H \setminus \{k\}} A_1[A_3/k]$$

Proof. Let us denote by A_{12} (resp. A_{13}) the open automaton resulting from $A_1[A_2/k]$ (resp. $A_1[A_3/k]$), to prove the theorem it is sufficient to prove that there exists a relation between states of the two open automata that satisfies the conditions of the Definition 10. We denote $A_1 = \langle S_1, s_{01}, J_1, V_1, \sigma_{01}, T_1 \rangle$ and $A_2 = \langle S_2, s_{02}, J_2, V_2, \sigma_{02}, T_2 \rangle$ and $A_3 = \langle S_3, s_{03}, J_3, V_3, \sigma_{03}, T_3 \rangle$.

Let R be the refinement relation relating states of A_2 and A_3. Let us denote with t^2 and t^3 the elements of A_2 and A_3 respectively. Consider any two states $s_{12} = \left(s_1^1, s_2\right)$ and $s_{13} = \left(s_1^2, s_3\right)$ (s_1^1 and s_1^2 are the same with renaming). We define a relation R' relating states of s_{12} and s_{13} as follows:

$$R'(s_{12}, s_{13}) = R(s_2, s_2) \wedge \checkmark_{A_{12}}(s_{12}) \wedge \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2$$

Let us denote $H' = H \uplus J_1 \setminus \{k\}$. We want to prove that (R', H') is a hole-tracking simulation of A_{12} and A_{13}. As for the previous proof we proceed by cases:

1. We have to prove that it holds for the initial configurations:

   $$\sigma_{012} \uplus \sigma_{013} \vdash R'(s_{012}, s_{013})$$

   with $\sigma_{012} = \sigma_{01}^1 \uplus \sigma_{02}$, $\sigma_{013} = \sigma_{01}^2 \uplus \sigma_{03}$, $s_{012} = (s_{01}^1, s_{02})$, and $s_{013} = (s_{01}^2, s_{03})$. By using the fact that R relates initial configurations of A_2 and A_3, we have the following statement: $(\sigma_{02} \uplus \sigma_{03} \vdash R(s_{02}, s_{03}))$, and based on the fact that initial states are reachable $\sigma_{012} \vdash \checkmark_{A_{12}}(s_{012})$ we have:

   $$(\sigma_{02} \uplus \sigma_{03} \vdash R(s_{02}, s_{03})) \wedge (\sigma_{012} \vdash \checkmark_{A_{12}}(s_{012}))$$

Because initial valuations σ_{01}^1 and σ_{01}^2 associate the same values to the "same" variables modulo renaming, so the following holds:
$\left(\bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2\right) \{\!\{\sigma_{01}^1 \uplus \sigma_{01}^2\}\!\}$.
Furthermore, because the domains of the substitution function are disjoint, the substitution function has an effect only on the related elements, we get:

$$
\begin{aligned}
&(\sigma_{02} \uplus \sigma_{03} \vdash R(s_{02}, s_{03})) \wedge (\sigma_{012} \vdash \checkmark_{A_{12}}(s_{012})) \\
&\Longrightarrow R(s_{02}, s_{03})\{\!\{\sigma_{02} \uplus \sigma_{03}\}\!\} \wedge \checkmark_{A_{12}}(s_{012})\{\!\{\sigma_{012}\}\!\} \wedge \left(\bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2\right) \{\!\{\sigma_{01}^1 \uplus \sigma_{01}^2\}\!\} \\
&\Longrightarrow \left(R(s_{02}, s_{03}) \wedge \checkmark_{A_{12}}(s_{012}) \wedge \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2\right) \{\!\{\sigma_{02} \uplus \sigma_{03} \uplus \sigma_{01}^1 \uplus \sigma_{01}^2 \uplus \sigma_{012}\}\!\} \\
&\Longrightarrow \sigma_{012} \uplus \sigma_{013} \vdash R'(s_{012}, s_{013})
\end{aligned}
$$

2. Second, we need to prove for any OT t_{12} in T_{12} originating from s_{12}

$$\frac{\beta_{12j}^{j \in J'_{12}}, g_{12}, \psi_{12}}{s_{12} \xrightarrow{\alpha_{12}} s'_{12}} \in \mathrm{OT}(s_{12})$$

there exists an indexed family t_{13x} of OTs originating from s_{13} that simulate it:

$$\left(\frac{\beta_{13xj}^{j \in J'_{13x}}, g_{13x}, \psi_{13x}}{s_{13} \xrightarrow{\alpha_{13x}} s'_{13x}} \in \mathrm{OT}(s_{13})\right)^{x \in X}$$

such that $(\forall x \in X, J'_{13x} \cap H' = J'_{12} \cap H')$ and

$$
\begin{aligned}
&V_{12} \uplus V_{13} \uplus \mathit{vars}(t_{12}) \vdash \\
&\qquad R'(s_{12}, s_{13}) \wedge g_{12} \implies \bigvee_{x \in X} \begin{pmatrix} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j \in J'_{13x} \cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{13x} \wedge R'\left(s'_{12}, s'_{13x}\right)\{\!\{\psi_{12} \uplus \psi_{13x}\}\!\} \end{pmatrix}
\end{aligned}
$$

By the definition of composition and open automata refinement we have:

$V_{12} = V_1^1 \uplus V_2$ and $V_{13} = V_1^2 \uplus V_3$
$H' = H_{23} \uplus J_1 \setminus \{k\} \subseteq J_1 \uplus (J_2 \cap J_3) \setminus \{k\} = \left(J_1^1 \uplus J_2 \setminus \{k\}\right) \cap \left(J_1^2 \uplus J_3 \setminus \{k\}\right) = J_{12} \cap J_{13}$

We have by hypothesis $A_2 \leqslant_H A_3$, then for any open transition t_2 in T_2 originating from s_2:

$$\frac{\beta_{2j}^{j \in J'_2}, g_2, \psi_2}{s_2 \xrightarrow{\alpha_2} s'_2} \in \mathrm{OT}(s_2)$$

there exists an indexed family of OTs originating from s_2:

$$\left(\frac{\beta_{3xj}^{j\in J'_{3x}}, g_{3x}, \psi_{3x} \in \mathrm{OT}(s_3)}{s_3 \xrightarrow{\alpha_{3x}} s'_{3x}}\right)^{x\in X}$$

such that $\forall x \in X, J'_{3x} \cap H = J'_2 \cap H$ and

$$V_2 \uplus V_3 \uplus vars(t_2) \vdash$$
$$\left(R(s_2, s_3) \wedge g_2 \implies \bigvee_{x\in X} \left(\begin{array}{c} \alpha_2 = \alpha_{3x} \wedge \bigwedge_{j\in J'_{3x}\cap H} \beta_{2j} = \beta_{3xj} \wedge \\ g_{3x} \wedge R\left(s'_2, s'_{3x}\right)\{\!\{\psi_2 \uplus \psi_{3x}\}\!\} \end{array}\right)\right) \qquad (*)$$

Consider any transition t_{12} in A_{12}, t_{12} is obtained (by case analysis on the definition of composition).

First case: Both automata perform a transition.

The transition t_{12} is obtained by the composition of transitions $t_2 = \frac{\beta_{2j}^{j\in J'_2}, g_2, \psi_2}{s_2 \xrightarrow{\alpha_2} s'_2}$ and

$$t_1^1 = \frac{(\beta_{1j}^1)^{j\in J_1^{1\prime}}, g_1^1, \psi_1^1}{s_1^1 \xrightarrow{\alpha_1^1} s_1^{1\prime}} \qquad \text{when } k \in J_1^{1\prime}$$

The result is the transition:

$$t_{12} = \frac{\beta_{1j}^{1\ j\in J_1^{1\prime}\setminus\{k\}} \uplus (\beta_{2j})^{j\in J'_2}, g_1^1 \wedge g_2 \wedge \alpha_2 = \beta_{1k}^1, \psi_1^1 \uplus \psi_2}{\left(s_1^1, s_2\right) \xrightarrow{\alpha_1^1} \left(s_1^{1\prime}, s'_2\right)} \qquad \text{where } k \in J_1^{1\prime}$$

We then obtain a family of OTs by the simulation of A_2 by A_3 (stated above). By hypothesis we have $k \in H$, so in the case where $k \in J_1^{1\prime}$, we deduce that $k \in J'_{3x}$ we can then build a family of OTs $t_{13x}^{x\in X}$ with the same transitions of A_1 (up to renaming) as those used to build t_{12}.

$$t_{13x} = \left(\frac{(\beta_{1j}^2)^{j\in J_1^{2\prime}\setminus\{k\}} \uplus \beta_{3xj}^{j\in J'_{3x}}, g_1^2 \wedge g_{3x} \wedge \alpha_{3x} = \beta_{1k}^2, \psi_{3x} \uplus \psi_1^2}{\left(s_1^2, s_3\right) \xrightarrow{\alpha_1^2} \left(s_1^{2\prime}, s'_{3x}\right)}\right)^{x\in X}$$

Recall that in this case $k \in J_1^{1\prime}$

$$\begin{aligned}
J'_{13} \cap H' &= ((J_1^{2\prime} \uplus J'_{3x}) \setminus \{k\}) \cap ((H \uplus J_1) \setminus \{k\}) \\
&= ((J_1^{2\prime} \cap (H \uplus J_1)) \uplus (J'_{3x} \cap (H \uplus J_1))) \setminus \{k\} \\
&= ((J_1^{2\prime} \cap J_1)) \uplus (J'_{3x} \cap H)) \setminus \{k\} \text{ since } J_1^{2\prime} \cap H = \varnothing \text{ and } J'_{3x} \cap J_1 = \varnothing \\
&= (J_1^{2\prime} \uplus (J'_{3x} \cap H)) \setminus \{k\} \text{ since } J_1^{1\prime} \subseteq J_1 \\
&= (J_1^{1\prime} \uplus (J'_2 \cap H)) \setminus \{k\} \text{ since } J_1^{1\prime} = J_1^{2\prime} \text{ and } J'_2 \cap H = J'_{3x} \cap H \\
&= ((J_1^{1\prime} \cap J_1) \uplus (J'_2 \cap H)) \setminus \{k\} \text{ since } J_1^{1\prime} \subseteq J_1 \\
&= (J'_2 \cap (J_1 \uplus H)) \uplus (J_1^{1\prime} \cap (J_1 \uplus H)) \setminus \{k\} \text{ since } J_1 \cap J'_2 = \varnothing \text{ and } H \cap J_1^{1\prime} = \varnothing \\
&= ((J'_2 \uplus J_1^{1\prime}) \setminus \{k\}) \cap ((J_1 \uplus H) \setminus \{k\}) \\
&= J'_{12} \cap H'
\end{aligned}$$

In this case the composition gives:

$$g_{12} \Leftrightarrow g_1^1 \wedge g_2 \wedge \alpha_2 = \beta_{1k}^1 \text{ and } g_{13x} \Leftrightarrow g_1^2 \wedge g_{3x} \wedge \alpha_{3x} = \beta_{1k}^2$$

Furthermore, as $\beta_{1k}^1 = \beta_{1k}^2$ then we get:

$$g_1^1 \wedge \alpha_2 = \beta_{1k}^1 \Leftrightarrow g_1^2 \wedge \alpha_{3x} = \beta_{1k}^2$$

As the previous case, the rest of the proof is based on the fact:

(a) The composition doesn't change the resulting actions, nor their variables, we can extend the valuation context of the variables to cover the variables of the transition t_3. By construction of t_{12} and t_{13} we have $\alpha_{12} = \alpha_1^1$ and $\alpha_{13x} = \alpha_1^2$. As $\alpha_1^1 = \alpha_1^2$ we deduce: $\alpha_{12} = \alpha_{13x}$.

(b) By composition we have also:
$\beta_{12j}^{j \in J'_{12}} = (\beta_{1j}^1)^{j \in J_1^{1\prime} \setminus \{k\}} \uplus \beta_{2j}^{j \in J'_2}$ and $\beta_{13xj}^{j \in J'_{13}} = (\beta_{1j}^2)^{j \in J_1^{2\prime} \setminus \{k\}} \uplus \beta_{3xj}^{j \in J'_{3x}}$
Therefore, we have for all $j \in J'_{12}$ (recall that $J'_{12} = J'_{13}$):

$$\beta_{12j} = \beta_{13xj} \Rightarrow (j \in J_1^{1\prime} \wedge \beta_{1j}^1 = \beta_{1j}^2) \vee (j \in J'_2 \wedge \beta_{2j} = \beta_{3xj})$$

(c) And considering β_{1j}^1 and β_{1j}^2 are the same (up to renaming) we have:
$V_1^1 \uplus V_1^2 \uplus vars(t_1) \vdash \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigwedge_{j \in J'_1} \beta_{1j}^1 = \beta_{1j}^2$.
Suppose that $V_2 \uplus V_3 \uplus vars(t_{3x}) \vdash \bigwedge_{j \in J'_{3x} \cap H} \beta_{2j} = \beta_{3xj}$ holds then we can deduce:

$$V_1^1 \uplus V_1^2 \uplus V_2 \uplus V_3 \uplus vars(t_{13x}) \vdash \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigwedge_{j \in J'_1} \beta_{1j}^1 = \beta_{1j}^2 \vee \bigwedge_{j \in J'_{3x} \cap H} \beta_{2j} = \beta_{3xj}$$

$$\Rightarrow V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigwedge_{j \in (J'_{3x} \cap H) \uplus J_1^{1\prime}} \beta_{12j} = \beta_{13xj}$$

$$\Rightarrow V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigwedge_{j \in ((J'_{3x} \cap H) \uplus J_1^{1\prime}) \setminus \{k\}} \beta_{12j} = \beta_{13xj}$$

$$\Rightarrow V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigwedge_{j \in (J'_{13x} \cap H')} \beta_{12j} = \beta_{13xj}$$

From the statements resulting the cases (a), (b) and (c), and by the formula $(*)$ we get:

$$V_2 \uplus V_3 \uplus vars(t_2) \vdash$$

$$R(s_2, s_3) \wedge g_2 \wedge \left(g_1^1 \wedge \alpha_2 = \beta_{1k}^1\right) \wedge \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies$$

$$\bigvee_{x \in X} \begin{pmatrix} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j \in J'_{13x} \cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{3x} \wedge R\left(s'_2, s'_{3x}\right) \{\!\!\{\psi_2 \uplus \psi_{3x}\}\!\!\} \end{pmatrix} \wedge \left(g_1^2 \wedge \alpha_{3x} = \beta_{1k}^2\right)$$

which can be simplified and re-written as follows:

$$V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash$$
$$R(s_2, s_3) \wedge g_{12} \wedge \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \implies \bigvee_{x \in X} \begin{pmatrix} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j \in J'_{13x} \cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{13x} \wedge R\big(s'_2, s'_{3x}\big) \{\!\{ \psi_2 \uplus \psi_{3x} \}\!\} \end{pmatrix}$$

Furthermore, we have for any transition t_1 in A_1 relying s_1 and s'_1 and for each $x \in X$:

$$V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash$$
$$\bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2 \implies \bigwedge_{v_1 \in V_1} \big(v_1^1 = v_1^2\big) \{\!\{ \psi_{13x} \}\!\} \wedge s_1^{1\prime} = s_1^{2\prime}$$

We can deduce for all $x \in X$ the following statement:

$$V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash$$
$$\bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2 \implies \bigvee_{x \in X} \left(\bigwedge_{v_1 \in V_1} \big(v_1^1 = v_1^2\big) \{\!\{ \psi_{13x} \}\!\} \wedge s_1^{1\prime} = s_1^{2\prime} \right)$$

Therefore, we get:

$$V_{12} \uplus V_{13} \uplus vars(t_{13x}) \vdash$$
$$R(s_2, s_3) \wedge g_{12} \wedge \bigwedge_{v_1 \in V_1} v_1^1 = v_1^2 \wedge s_1^1 = s_1^2 \implies$$
$$\bigvee_{x \in X} \left(\begin{array}{l} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j \in J'_{13x} \cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{13x} \wedge R\big(s'_2, s'_{3x}\big) \{\!\{ \psi_2 \uplus \psi_{3x} \}\!\} \wedge \bigwedge_{v_1 \in V_1} \big(v_1^1 = v_1^2\big) \{\!\{ \psi_{13x} \}\!\} \wedge s_1^{1\prime} = s_1^{2\prime} \end{array} \right)$$

From the hypothesis, we have $A_1[A_2/k]$ is non-blocking. Then, by definition of the reachability (Definition 7) we have for all $t_{12} \in T_{12}$:

$$vars(t_{12}) \vdash \big(\checkmark_{A_{12}}(s_{12}) \wedge g_{12} \implies \checkmark_{A_{12}}\big(s'_{12}\big) \{\!\{ \psi_{12} \}\!\} \big)$$

With the conjunction of the formula obtained above and after simplifying, we get the expected result:

$$V_{12} \uplus V_{13} \uplus vars(t_{12}) \vdash$$
$$R'(s_{12}, s_{13}) \wedge g_{12} \implies \bigvee_{x \in X} \begin{pmatrix} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j \in J'_{13x} \cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{13x} \wedge R'\big(s'_{12}, s'_{13x}\big) \{\!\{ \psi_{12} \uplus \psi_{13x} \}\!\} \end{pmatrix}$$

Second case: Only the encompassing automaton performs a transition.

The transition t_{12} is obtained by the transition $t_1^1 = \frac{(\beta_{1j}^1)^{j\in J_1^{1\prime}}, g_1^1, \psi_1^1}{s_1^1 \xrightarrow{\alpha_1^1} s_1^{1\prime}}$ alone with the state s_2 unchanged, if $k \notin J_1^{1\prime}$

$$t_{12} = \frac{(\beta_{1j}^1)^{j\in J_1^{1\prime}}, g_1^1, \psi_1^1}{\left(s_1^1, s_2\right) \xrightarrow{\alpha_1^1} \left(s_1^{1\prime}, s_2\right)}$$

We then obtain a family of OTs by the simulation of A_2 by A_3 (stated above). As $k \in H$ we have $k \notin J'_{3x}$ and we build with the same transitions of A_1 (up renaming):

$$t_{13x} = \left(\frac{(\beta_{1j}^2)^{j\in J_1^{2\prime}}, g_1^2, \psi_1^2}{\left(s_1^2, s_3\right) \xrightarrow{\alpha_1^2} \left(s_1^{2\prime}, s_3\right)} \right)^{x\in X}$$

We have to prove the following formula in both cases (depending on how the transition t_{12} is built): $(\forall x \in X, J'_{13x} \cap H' = J'_{12} \cap H')$ and

$$V_{12} \uplus V_{13} \uplus vars(t_{12}) \vdash$$

$$R'(s_{12}, s_{13}) \wedge g_{12} \implies \bigvee_{x\in X} \left(\begin{array}{c} \alpha_{12} = \alpha_{13x} \wedge \bigwedge_{j\in J'_{13x}\cap H'} \beta_{12j} = \beta_{13xj} \wedge \\ g_{13x} \wedge R'\left(s'_{12}, s'_{13x}\right) \{\!\!\{ \psi_{12} \uplus \psi_{13x} \}\!\!\} \end{array} \right)$$

Recall in this case $k \notin J_1^{1\prime}$, so for all $x \in X$ we have:

$$\begin{aligned} J'_{13x} \cap H' &= J'_{3x} \cap (J_1 \uplus H \setminus \{k\}) \\ &= \left(J'_{3x} \cap H\right) \text{ since } J'_{3x} \cap J_1 = \varnothing \wedge k \notin J'_{3x} \\ &= \left(J'_2 \cap H\right) \text{ since } J'_2 \cap H = J'_{3x} \cap H \\ &= \left(J'_{12} \cap H'\right) \text{ since } J_2 \cap J'_1 = \varnothing \wedge k \notin J_1^{1\prime} \end{aligned}$$

The proof of this case follows the same steps as the previous case the only argument that changes is that by composition we obtain: $g_{12} \Leftrightarrow g_1^1$ and $g_{13x} \Leftrightarrow g_2^1$.

It remains to prove the satisfaction of the deadlock reducing condition: for all $(s_{12}, s_{13}) \in S_{12} \times S_{13}$

$$V_{12} \uplus V_{13} \uplus \biguplus_{t_{13}\in \mathrm{OT}(s_{13})} vars(t_{13}) \vdash$$

$$\left(R'(s_{12}, s_{13}) \wedge \bigvee_{t_{13}\in \mathrm{OT}(s_{13})} guard(t_{13}) \implies \bigvee_{t_{12}\in \mathrm{OT}(s_{12})} guard(t_{12}) \right)$$

Let's start with the hypothesis stating that $A_2 \leqslant_H A_3$, thus we have for all $(s_2, s_3) \in S_2 \times S_3$:

$$V_2 \uplus V_3 \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash \left(R(s_2, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \implies \bigvee_{t_{2p} \in \mathrm{OT}(s_2)} guard(t_2) \right)$$

By adding the same term $\checkmark_{A_{12}}(s_{12}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1)$ to both sides of the implication, we get:

$$\begin{aligned} V_2 \uplus V_3 \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash \\ R(s_2, s_3) \wedge \bigvee_{t_3 \in \mathrm{OT}(s_3)} guard(t_3) \wedge \checkmark_{A_{12}}(s_{12}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \implies \\ \bigvee_{t_{2p} \in \mathrm{OT}(s_2)} guard(t_2) \wedge \checkmark_{A_{12}}(s_{12}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \end{aligned}$$

that can be re-written

$$\begin{aligned} V_2 \uplus V_3 \uplus \biguplus_{t_3 \in \mathrm{OT}(s_3)} vars(t_3) \vdash \\ R(s_2, s_3) \wedge \bigvee_{t_{13} \in \mathrm{OT}(s_{13})} guard(t_{13}) \wedge \checkmark_{A_{12}}(s_{12}) \implies \bigvee_{t_2 \in \mathrm{OT}(s_2)} guard(t_2) \wedge \checkmark_{A_{12}}(s_{12}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \end{aligned}$$

Yet we have by hypothesis $A_1[A_2/k]$ is non-blocking, by definition of non-blocking composition (Definition 8) we have:

$$V_{12} \uplus \biguplus_{t_1 \in \mathrm{OT}(s_1)} vars(t_1) \vdash \checkmark_{A_{12}}(s_{12}) \wedge \bigvee_{t_1 \in \mathrm{OT}(s_1)} guard(t_1) \implies \bigvee_{t_{12} \in \mathrm{OT}(s_{12})} guard(t_{12})$$

From the two previous formulas and considering that each valuations σ_1^1 and σ_1^2 associate the same values to the "same" variables modulo renaming we obtain the expected result.

□