PIOA Simulations

June 1, 2018

Given a PIOA P, let Tr_P be the set of all valid transitions of P: transitions $s \xrightarrow{a} t$ such that $t \in \mathsf{supp}(\mathsf{tr}\ s\ a)$. Given a task T and a state s such that T is enabled in s, write Ts to be the induced distribution on Tr_P. (By input enabledness, we may also write is for any input action i to be the corresponding induced distribution.) Lift this to execution fragments α such that T is enabled in $\mathsf{lstate}\ \alpha$ in the obvious way. Then, lift this to distributions of execution fragments μ, by sampling α from μ, and returning $T\ \alpha$ if T is enabled in $\mathsf{lstate}\ \alpha$, and α otherwise. (Note by this definition, $T\ \mu$ is equal to $\mathsf{apply}(T, \mu)$.)

Given a task sequence σ, say that σ is disabled in s if all tasks in σ are disabled in s, and that σ is enabled in s if there exists a task in σ enabled in s.

Given two comparable (not necessarily closed) task-structured PIOAs P and P', a state simulation from P to P' is a tuple of functions $\phi_S : S_P \to S_{P'}$, $\phi_T : \mathsf{Tr}_P \to \mathsf{Frag}_{P'}$, $C : \mathsf{Task}_P \to \mathsf{Task}^*_{P'}$ such that:

1. $\phi_S\ \mathsf{start}_P = \mathsf{start}_{P'}$
2. for all $t \in \mathsf{Tr}_P$, $\phi_S(\mathsf{fstate}\ t) = \mathsf{fstate}\ (\phi_T\ t)$
3. for all t, $\phi_S(\mathsf{lstate}\ t) = \mathsf{lstate}\ (\phi_T\ t)$
4. for all t, $\mathsf{tr}\ t = \mathsf{tr}\ (\phi_T\ t)$
5. for all s and tasks T, T is enabled in s implies $C(T)$ is enabled in $\phi_S\ s$
6. for all tasks T and states s such that $a \in T$ is enabled in s, $\mathcal{D}(\phi_T)(Ts) = \mathcal{D}(\phi_T)(as) = C(T)\ (\phi_S\ s)$
7. for all input actions i, $\phi_t(s \xrightarrow{i} s') = (\phi_S s) \xrightarrow{i} (\phi_S s')$.

Theorem 1. *If there exists a state simulation from P to P' then $P \leq_0 P'$.*

The above is proved in two parts:

Lemma 1 (Soundness)**.** *If P and P' are closed and there exists a state simulation from P to P', then there exists a simulation from P to P'.*

Lemma 2 (Compositionality)**.** *If there exists a state simulation from P to P', then for all compatible environments E, there exists a state simulation from $P||E$ to $P'||E$.*

1 Soundness

Let (ϕ_S, ϕ_T, C) be a state simulation from P to P'. Define $\overline{\phi} : \mathsf{Frag}_P \to \mathsf{Frag}_{P'}$ such that $\overline{\phi}(s) = \phi_S\ s$, and otherwise inductively by

$$\overline{\phi}\ \alpha = \begin{cases} \phi_T\ (s \xrightarrow{a} s') & \text{if } \alpha = s \xrightarrow{a} s' \\ (\overline{\phi}\ \alpha')^\frown(\phi_T\ (\mathsf{lstate}\ \alpha' \xrightarrow{a}\ s)) & \text{if } \alpha = \alpha' \xrightarrow{a}\ s \end{cases}$$

Define the simulation relation $R \subseteq \mathcal{D}(\mathsf{Exec}\ P) \times \mathcal{D}(\mathsf{Exec}\ P')$ to be $\mu\ R\ \eta$ iff $\mathcal{D}(\overline{\phi})(\mu) = \eta$. Observe that ϕ_T preserves traces, so the trace condition for R holds. Also, observe the start condition for R holds, by the definition of a state simulation.

Now, we must verify that the step condition for R holds. Choose the task correspondence C to be that in the state simulation. Lift C to operate on task sequences, by concatenating each image. Suppose that $\mu\ R\ \eta$ (i.e., that $\mathcal{D}(\overline{\phi})\ \mu = \eta$), μ is consistent with σ, η is consistent with $C(\sigma)$. Then we must verify that for all T, $\mathcal{D}(\overline{\phi})(T\ \mu) = C(T)\ \eta$; that is, we must show that $\mathcal{D}(\overline{\phi})(T\ \mu) = C(T)\ \mathcal{D}(\overline{\phi})\ \mu$.

Lemma 3. *For all execution fragments α such that T is enabled in $\mathsf{lstate}\ \alpha$, $\mathcal{D}(\overline{\phi})(T\alpha) = C(T)(\overline{\phi}\ \alpha)$.*

The above follows from condition 6 in the definition of a state simulation.

Lemma 4. *For all distributions μ of execution fragments, $\mathcal{D}(\overline{\phi})(T\ \mu) = C(T)\ (\mathcal{D}(\overline{\phi})\mu)$.*

Proof. Write $\mu = \sum_i p_i \alpha_i$. Then,

$$T\ \mu = \sum_i p_i \cdot \begin{cases} T\ \alpha_i & \text{if } T \text{ is enabled in } \mathsf{lstate}\ \alpha_i \\ \alpha_i & \text{otherwise} \end{cases}$$

thus

$$\mathcal{D}(\overline{\phi})(T\ \mu) = \sum_i p_i \begin{cases} \mathcal{D}(\overline{\phi})\ (T\ \alpha_i) & \text{if } T \text{ is enabled in } \mathsf{lstate}\ \alpha_i \\ \overline{\phi}\ \alpha_i & \text{otherwise} \end{cases}$$

By conditions 3, 5 and 6 in the state simulation, this is equal to

$$\sum_i p_i \begin{cases} C(T)(\overline{\phi}\ \alpha_i) & \text{if } C(T) \text{ is enabled in } \phi_S(\mathsf{lstate}\ \alpha_i) = \mathsf{lstate}\ \overline{\phi}\ \alpha_i \\ \overline{\phi}\ \alpha_i & \text{otherwise} \end{cases}$$

which is equal to $C(T)\ (\mathcal{D}(\overline{\phi})\ \mu)$.

$\square$

2 Compositionality

Let (ϕ_S, ϕ_T, C) be a state simulation from P to P', and let E be compatible with P and P'. Then, define the following state simulation from $P||E$ to $P'||E$:

$$\phi'_S(s,t) = (\phi_S\ s, t)$$

$$\phi'_T((s,t) \xrightarrow{a} (s',t')) = \begin{cases} \mathsf{lift}_t^{t'}(\phi_T(s \xrightarrow{a} s')) & \text{if } a \text{ is enabled in } s \\ (\phi_S\ s, t) \xrightarrow{a} (\phi_S\ s', t') & \text{otherwise} \end{cases}$$

$$C'T = \begin{cases} C(T) & \text{if } T \in \mathsf{Task}_P \\ T & \text{otherwise} \end{cases}$$

where $\mathsf{lift}_t^{t'}\ \alpha$ lifts each transition in $\mathsf{Tr}_{P'}$ to a transition in $\mathsf{Tr}_{P'||E}$, such that t is the initial state of E and t' is the final state. This is always possible, since we know that $(s,t) \xrightarrow{a} (s',t')$ and that ϕ_T only schedules hidden actions in P', except possibly a single external action which must be enabled in t.

We must now verify the conditions for (ϕ'_S, ϕ'_T, C') to be a state simulation. Conditions 1-5 are immediate.

For condition 7, compute that

$$\begin{aligned}
\phi'_T((s,t) \xrightarrow{i} (s',t')) &= \mathsf{lift}_t^{t'}\, \phi_T(s \xrightarrow{i} s') \text{ by input enabledness} \\
&= \mathsf{lift}_t^{t'}\, (\phi_S\ s) \xrightarrow{i} (\phi_S\ s') \text{ by condition 7 on } \phi_T \\
&= (\phi_S\ s, t) \xrightarrow{i} (\phi_S\ s', t') \\
&= \phi'_S(s,t) \xrightarrow{i} \phi'_S(s',t').
\end{aligned}$$

Now, we need to verify that for all tasks $T \in \mathsf{Task}_{P||E}$ and joint states (s,t) such that T is enabled in (s,t),

$$\mathcal{D}(\phi'_T)(T(s,t)) = C'(T)(\phi'_S\ (s,t)).$$

If $T \in \mathsf{Task}_P$ and Ts induces an action a such that a is enabled in t, then we need to check that

$$\mathcal{D}(\phi'_T)(a(s,t)) = C(T)(\phi_S\ s, t).$$

Since actions operate separately on each component, we get that $a(s,t) = \sum_i \sum_j p_i p'_j (s,t) \xrightarrow{a} (s_i, t_j)$, so that

$$\begin{aligned}
\mathcal{D}(\phi'_T)(\sum_i \sum_j p_i p'_j (s,t) \xrightarrow{a} (s_i,t_j)) &= \sum_i \sum_j p_i p'_j \phi'_T((s,t) \xrightarrow{a} (s_i,t_j)) \\
&= \sum_j p'_j \mathcal{D}(\mathsf{lift}_t^{t_j}) \sum_i \phi_T(s \xrightarrow{a} s_i) \\
&= \sum_j p'_j \mathcal{D}(\mathsf{lift}_t^{t_j}) C(T)(\phi_S\ s) \\
&= C(T)(\phi_S\ s, t).
\end{aligned}$$

If a is not enabled in t, then similarly compute that

$$\begin{aligned}\mathcal{D}(\phi'_T)(\sum_i p_i(s,t) \xrightarrow{a} (s_i,t)) &= \mathcal{D}(\mathsf{lift}^t_t)\sum_i p_i\phi_T(s \xrightarrow{a} s_i)\\ &= \mathcal{D}(\mathsf{lift}^t_t)C(T)(\phi_S\ s)\\ &= C(T)(\phi_S\ s,t).\end{aligned}$$

If $T \in \mathsf{Task}_E$ and Tt induces an action a such that a is enabled in s, we must check that

$$\mathcal{D}(\phi'_T)(T(s,t)) = T(\phi_S\ s,t).$$

As above, write $a(s,t) = \sum_i \sum_j p_i p'_j(s,t) \xrightarrow{a} (s_i,t_j)$ so that

$$\begin{aligned}\mathcal{D}(\phi'_T)(\sum_i\sum_j p_i p'_j(s,t) \xrightarrow{a} (s_i,t_j)) &= \sum_i\sum_j p_i p'_j \phi'_T((s,t) \xrightarrow{a} (s_i,t_j))\\ &= \sum_i\sum_j p_i p'_j \mathsf{lift}^{t_j}_t \phi_T(s \xrightarrow{a} s_i).\end{aligned}$$

Now, since a arises from a task in E, a must be an input action for s, thus we get

$$\begin{aligned}\sum_i\sum_j p_i p'_j \mathsf{lift}^{t_j}_t \phi_T(s \xrightarrow{a} s_i) &= \sum_i\sum_j p_i p'_j \mathsf{lift}^{t_j}_t (\phi_S s) \xrightarrow{a} (\phi_S\ s_i)\\ &= \sum_i\sum_j p_i p'_j (\phi_S\ s,t) \xrightarrow{a} (\phi_S\ s_i, t_j)\\ &= T(\phi_S\ s,t).\end{aligned}$$

.

Finally, if a is not enabled in s, write $a(s,t) = \sum_j p'_j(s,t) \xrightarrow{a} (s,t_j)$ so that

$$\begin{aligned}\mathcal{D}(\phi'_T)(\sum_j p'_j(s,t) \xrightarrow{a} (s,t_j)) &= \sum_j p'_j \phi'_T((s,t) \xrightarrow{a} (s,t_j))\\ &= \sum_j p'_j((s,t) \xrightarrow{a} (s,t_j)) \text{ since } a \text{ is disabled in } s\\ &= T(\phi_S\ s,t).\end{aligned}$$