Towards a theory of bisimulation for local names

Alan Jeffrey
CTI, DePaul University

- *Completeness*. We must show that contextual equivalence implies weak bisimilarity. To do this we show that each transition $\xRightarrow{\gamma}$ corresponds to a small piece of context $\mathsf{C}_\gamma[t]$ such that $t \xRightarrow{\gamma} v$ iff $\mathsf{C}_\gamma[t] \Rightarrow (v, \mathtt{true})$. (We call such lts's *contextual*: the notion that transition labels should correspond to small contexts appears to be folklore, and has only recently been investigated formally by Sewell [20].) This formal relationship between labelled observations and reduction in contexts yields completeness because non-bisimilar terms have a distinguishing trace of labelled actions, yielding a distinguishing context.

- For the converse, *soundness*, we must show that bisimilarity implies contextual equivalence, for which it is sufficient to demonstrate that bisimilarity is a congruence.

We note that our approach to characterising contextual equivalence is already in sharp contrast to Pitts and Stark. They propose logical relations as an operational proof technique for establishing contextual equivalence of ν-calculus terms. The logical relation can easily be construed as a form of bisimulation on an lts, but the labels which would have to be used are not contextual—this compromises completeness in order to obtain a direct proof of soundness for their technique.

In the case of the λthis cotactlcoo obti di

A (strong) bisimulation is a (strong) simulation whose inverse is a simulation.

Let $\approx$ be the largest bisimulation, and let $\sim$ be the largest strong bisimulation.

We can extend these relations from closed terms to open terms by closing with any appropriately typed values. A type-indexed relation R on closed terms can be extended to a relation R° on open terms:

$$\begin{array}{l}\Gamma \vDash t\; R^\circ\; t' : \sigma \\ \quad \text{iff for all } \vdash [\overline{v}/\overline{x}] : \Gamma \text{ we have:} \\ \quad\quad \vDash (t[\overline{v}/\overline{x}])\; R\; (t'[\overline{v}/\overline{x}]) : \sigma\end{array}$$

where we write $\vdash [\overline{v}/\overline{x}] : (\overline{x} : \overline{\sigma})$ whenever $\vdash \overline{v} : \overline{\sigma}$.

. ' Example

Let *not* be defined:

$$\mathit{not} \overset{\text{def}}{=} \lambda x : \mathtt{bool}\,.\, \mathtt{if}\; x\; \mathtt{then}\; \mathtt{false}\; \mathtt{else}\; \mathtt{true}$$

then one sample reduction of *not* is:

$$\begin{array}{lll}
\mathit{not} & \xRightarrow{\mathtt{copy}} & (\mathit{not}, \mathit{not}) \\
 & \xRightarrow{\mathtt{l.@true}} & (\mathit{not}(\mathtt{true}), \mathit{not}) \\
 & \xRightarrow{\tau} & (\mathtt{false}, \mathit{not}) \\
 & \xRightarrow{\mathtt{r.@false}} & (\mathtt{false}, \mathit{not}(\mathtt{false})) \\
 & \xRightarrow{\tau} & (\mathtt{false}, \mathtt{true}) \\
 & \xRightarrow{\mathtt{l.false}} & ((), \mathtt{true}) \\
 & \xRightarrow{\mathtt{r.true}} & ((), ())
\end{array}$$

showing how *not* evaluates when applied to `true` or `false`.

. Completeness

In this section, we shall show that bisimulation is *complete*, that is:

$$\text{if } t \approx_{ctx} t' \text{ then } t \approx^\circ t'$$

First we observe that the λ-calculus is deterministic and normalizing, and so bisimulation and trace equivalence coincide.

We then show that contextual equivalence implies trace equivalence by constructing a context $\mathsf{C}_{\overline{\gamma}}$ for each sequence of labels $\overline{\gamma}$ so that the context induces reductions for each label:

Lemma .1 *For every sequence $\overline{\gamma}$ of transition labels there is a context $\mathsf{C}_{\overline{\gamma}}$ such that:*

$$(\vdash t : \sigma) \xRightarrow{\overline{\gamma}} \Rightarrow (\vdash v : \sigma') \quad \textit{iff} \quad (\vdash \mathsf{C}_{\overline{\gamma}}[t] : (\sigma' \times \mathtt{bool})) \Rightarrow (\vdash (v, \mathtt{true}) : (\sigma' \times \mathtt{bool}))$$

$$\begin{aligned}
\mathsf{C}_{\mathtt{true}}[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t\, \mathtt{in}\, (x,x) \\
\mathsf{C}_{\mathtt{false}}[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t\, \mathtt{in}\, (x, \mathit{not}(x)) \\
\mathsf{C}_{@v}[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t(v)\, \mathtt{in}\, (x, \mathtt{true}) \\
\mathsf{C}_{\mathtt{copy}}[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t\, \mathtt{in}\, ((x,x), \mathtt{true}) \\
\mathsf{C}_{\mathtt{discard}}[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t\, \mathtt{in}\, ((), \mathtt{true}) \\
\mathsf{C}_{\mathtt{l}.\gamma}[t] &\overset{\text{def}}{=} \mathtt{let}\, (x_1,x_2) = t \\
&\qquad \mathtt{in}\, \mathtt{let}\, (x_1',x_2') = \mathsf{C}_\gamma[x_1]\, \mathtt{in}\, ((x_1',x_2),x_2') \\
\mathsf{C}_{\mathtt{r}.\gamma}[t] &\overset{\text{def}}{=} \mathtt{let}\, (x_1,x_2) = t \\
&\qquad \mathtt{in}\, \mathtt{let}\, (x_1',x_2') = \mathsf{C}_\gamma[x_2]\, \mathtt{in}\, ((x_1,x_1'),x_2')
\end{aligned}$$

We then prove that C_γ has the required property, by induction on γ. This is straightforward, as an example we demonstrate the case where the label is $\mathtt{l}.\gamma$.

Suppose $t \xRightarrow{\mathtt{l}.\gamma} \Rightarrow v$. We know that t must converge to a value, and by construction, we know that this value must be a pair, (v_1,v_2) say, such that $v_1 \xRightarrow{\gamma} \Rightarrow v'$, where $v = (v',v_2)$. Now,

$$\mathsf{C}_{\mathtt{l}.\gamma}[t] \Rightarrow \mathtt{let}\, (x_1',x_2') = \mathsf{C}_\gamma[v_1]\, \mathtt{in}\, ((x_1',v_2),x_2').$$

By induction we know that $\mathsf{C}_\gamma[v_1] \Rightarrow (v', \mathtt{true})$, thus we have

$$\mathtt{let}\, (x_1',x_2') = \mathsf{C}_\gamma[v_1]\, \mathtt{in}\, ((x_1',v_2),x_2') \Rightarrow ((v',v_2),\mathtt{true})$$

which is to say $\mathsf{C}_{\mathtt{l}.\gamma}[t] \Rightarrow (v, \mathtt{true})$.

Conversely, suppose that $\mathsf{C}_{\mathtt{l}.\gamma}[t] \Rightarrow (v, \mathtt{true})$. By inspection of the context we note that $t \Rightarrow (v_1,v_2)$ for some values and $\mathsf{C}_\gamma[v_1] \Rightarrow (v', \mathtt{true})$ such that v is (v',v_2). From this we know by induction that $v_1 \xRightarrow{\gamma} \Rightarrow v'$, whence $t \Rightarrow (v_1,v_2) \xRightarrow{\mathtt{l}.\gamma} ((v',v_2),\mathtt{true})$ as required.

For a sequence of labels, we define:

$$\begin{aligned}
\mathsf{C}_\varepsilon[t] &\overset{\text{def}}{=} \mathtt{let}\, x = t\, \mathtt{in}\, (x, \mathtt{true}) \\
\mathsf{C}_{\overline{\gamma},\overline{\gamma}'}[t] &\overset{\text{def}}{=} \mathtt{let}\, (x_1,x_2) = \mathsf{C}_{\overline{\gamma}}[t] \\
&\qquad \mathtt{in}\, \mathtt{let}\, (x_1',x_2') = \mathsf{C}_{\overline{\gamma}'}[x_1]\, \mathtt{in}\, (x_1', x_2 \wedge x_2')
\end{aligned}$$

The result follows by induction on the length of $\overline{\gamma}$. □

Theorem . (completeness for λ-calculus) *If* $\Gamma \vDash t \approx_{ctx} t' : \sigma$ *then* $\Gamma \vDash t \approx^\circ t' : \sigma$.

Proof. It suffices to show the result for closed terms. Let $\overline{\gamma}$ be a trace of t:

$$(\vdash t : \sigma) \xRightarrow{\overline{\gamma}}$$

so	$(\vdash t : \sigma) \xRightarrow{\overline{\gamma}} \Rightarrow (\vdash v : \sigma')$	(λ-calculus is terminating)
so	$(\vdash \mathsf{C}_{\overline{\gamma}}[t] : (\sigma' \times \mathtt{bool})) \Rightarrow (\vdash (v,\mathtt{true}) : (\sigma' \times \mathtt{bool}))$	(Lemma 2.1)
so	$(\vdash \mathtt{snd}\, (\mathsf{C}_{\overline{\gamma}}[t]) : \mathtt{bool}) \Rightarrow (\vdash \mathtt{true} : \mathtt{bool})$	(Defn of snd)
so	$(\vdash \mathtt{snd}\, (\mathsf{C}_{\overline{\gamma}}[t']) : \mathtt{bool}) \Rightarrow (\vdash \mathtt{true} : \mathtt{bool})$	($t \approx_{ctx} t'$)
so	$(\vdash \mathsf{C}_{\overline{\gamma}}[t'] : (\sigma' \times \mathtt{bool})) \Rightarrow (\vdash (v',\mathtt{true}) : (\sigma' \times \mathtt{bool}))$	(Defn of snd)
so	$(\vdash t' : \sigma) \xRightarrow{\overline{\gamma}} \Rightarrow (\vdash v' : \sigma')$	(Lemma 2.1)

Similarly, any trace of t' is a trace of t, so the terms are trace equivalent. Since the λ-calculus is deterministic, trace equivalence and bisimulation coincide, so $t \approx t'$. □

. Soundness

In this section, we shall show that bisimulation is *sound*, that is:

$$\text{if } t \approx^{\circ} t' \text{ then } t \approx_{ctx} t'$$

This result is immediate from the result that bisimulation is a congruence, for which we adopt Howe's technique [10], following Gordon [6].

For any type-indexed relation R, let $\widehat{R}$ be defined such that for each type rule in the language:

$$\frac{\overline{\Gamma} \vdash \overline{t} : \overline{\sigma}}{\Gamma \vdash op(\overline{t}) : \sigma}$$

we have:

$$\frac{\overline{\Gamma} \vDash \overline{t}\; R\; \overline{t}' : \overline{\sigma}}{\Gamma \vDash op(\overline{t})\; \widehat{R}\; op(\overline{t}') : \sigma}$$

For any type-indexed relation R, let $R^{\bullet}$ be defined:

$$\frac{t_1\; \widehat{R^{\bullet}}\; t_2\; R^{\circ}\; t_3}{t_1\; R^{\bullet}\; t_3}$$

Howe's proof depends first on showing that $\approx^{\bullet}$ is *subst*[illegible]

. Comments

The astute reader will notice that the `copy` and `discard` transitions are redundant in this setting. In fact, it is a well known property of pure functional languages that ‘operational extensionality’ holds, that is, contextual equivalence can be verified by using applicative contexts alone. This does certainly not hold true of the extensions to the λ-calculus which we will consider later in this paper where operational extensionality fails.

In a similar vein, we notice that the use of `l.` and `r.` tags rather than Gordon’s `fst` and `snd` transitions is also unnecessary here because pairing forms a product on values. In later sections, because of the presence of side-effects, the pairing operator is no longer a product, but is symmetric monoidal.

It is an important feature of the transition systems being used here, and also those of [6, 2] that they are *applicative* in nature. That is, any arbitrary pieces of code being carried in the label is always of lower order type than the term under scrutiny.

ν-calculus

We now extend the λ-calculus with unique name generation and equality testing, in order to investigate Pitts and Stark’s [13] ν-calculus.

Pitts and Stark have demonstrated that finding a sound and complete semantics for the ν-calculus is a difficult open problem. They provide a sound (but incomplete) semantics using logical relations. In this section, we provide an ‘upper bound’ to complement their ‘lower bound’ by presenting a bisimulation which is complete (but only sound up to first-order). We observe that our complete bisimulation provides a more investigative proof method for establishing contextual inequivalence which allows one to construct distinguishing contexts in a piecemeal fashion. This useful feature of the semantics avoids the need to build these, sometimes elaborate, contexts completely by making much of the construction automatic.

.1 Syntax and type rules

Extend the grammar of types with:

$$\sigma ::= \cdots \mid \mathtt{name}$$

Extend the grammar of values with:

$$v ::= \cdots \mid n$$

Extend the grammar of terms with:

$$t ::= \cdots \mid \nu n\,.\,t \mid v = v$$

Extend the type judgements $\Gamma \vdash t : \sigma$ to include a *name context* Δ of the form $n_1, \ldots, n_n$ for distinct n_i, so judgements are now of the form $\Gamma; \Delta \vdash t : \sigma$. The type rules for the new terms are:

$$\frac{}{\Gamma; \Delta, n, \Delta' \vdash n : \mathtt{name}} \qquad \frac{\Gamma; \Delta, n \vdash t : \sigma}{\Gamma; \Delta \vdash \nu n\,.\,t : \sigma}$$

$$\frac{\Gamma; \Delta \vdash v : \mathtt{name} \quad \Gamma; \Delta \vdash v' : \mathtt{name}}{\Gamma; \Delta \vdash v = v' : \mathtt{bool}}$$

The other rules do not change the name context.

. Reduction semantics

Terms no longer reduce to values, instead they now reduce to *prevalues* of the form:

$$p ::= \nu\overline{n}\,.\,v$$

Extend the reduction relation with (when $n \neq n'$):

$$\begin{array}{rcl} n = n & \xrightarrow{\tau} & \mathtt{true} \\ n = n' & \xrightarrow{\tau} & \mathtt{false} \end{array}$$

Extend the grammar of evaluation contexts by:

$$\mathsf{E} ::= \cdots \mid \nu n\,.\,\mathsf{E}$$

Replace the let-β reduction rule by:

$$\mathtt{let}\,x\ =\ \nu\overline{n}\,.\,v\ \mathtt{in}\,t \quad \xrightarrow{\tau} \quad \nu\overline{n}\,.\,t[v/x]$$

where we α-convert $\nu\overline{n}\,.\,v$ if necessary to ensure that none of the free names in t are captured. It is in this rule that *scope extrusion* of the static name binder occurs. There is an obvious translation from Pitts and Stark's ν-calculus into ours (theirs does not include pairing), and it is routine to show that this translation is adequate.

The definition of contextual equivalence remains the same, except that the results of a test can include some private names: $t \approx_{ctx} t'$ whenever for all closing contexts C of type `bool`, we have $\mathsf{C}[t] \Rightarrow \nu\overline{n}\,.\,\mathtt{true}$ iff $\mathsf{C}[t'] \Rightarrow \nu\overline{n}'\,.\,\mathtt{true}$.

. Labelled transition system semantics

We can no longer define the lts semantics as judgements $(\ \vdash v : \sigma) \xrightarrow{\gamma} (\ \vdash t$

and free names:

$$fn(n) = \{n\} \qquad fn(@v) = fn(v) \qquad fn(\overline{\gamma}, \overline{\gamma}') = fn(\overline{\gamma}) \cup fn(\overline{\gamma}') \setminus bn(\overline{\gamma})$$

A public name can be announced:

$$(\Delta \vdash n : \mathtt{name}) \quad \xrightarrow{n} \quad (\Delta \vdash () : \mathtt{unit})$$

The context $\nu n \,.\, \cdot$ is an observation context:

$$\frac{(\Delta, n \vdash p : \sigma) \xrightarrow{\gamma} (\Delta, n, \Delta' \vdash t : \sigma')}{(\Delta \vdash \nu n \,.\, p : \sigma) \xrightarrow{\gamma} (\Delta, \Delta' \vdash \nu n \,.\, t : \sigma')} [n \text{ not in } \gamma]$$

These are not bisimilar because the first term has the reduction:

$$\begin{array}{rcl}
\nu n\,.\,\lambda x:\texttt{unit}\,.\,n & \xRightarrow{\texttt{copy}} & \nu n\,.\,(\lambda x:\texttt{unit}\,.\,n, \lambda x:\texttt{unit}\,.\,n) \\
& \xRightarrow{\texttt{l}.@()} \Rightarrow & \nu n\,.\,(n, \lambda x:\texttt{unit}\,.\,n) \\
& \xRightarrow{\texttt{r}.@()} \Rightarrow & \nu n\,.\,(n, n) \\
& \xRightarrow{\texttt{l}.\nu n} & ((), n) \\
& \xRightarrow{\texttt{r}.n} & ((), ())
\end{array}$$

which the second term can only match:

$$\begin{array}{rcl}
\lambda x:\texttt{unit}\,.\,\nu n\,.\,n & \xRightarrow{\texttt{copy}} & (\lambda x:\texttt{unit}\,.\,\nu n\,.\,n, \lambda x:\texttt{unit}\,.\,\nu n\,.\,n) \\
& \xRightarrow{\texttt{l}.@()} \Rightarrow & (\nu n\,.\,n, \lambda x:\texttt{unit}\,.\,\nu n\,.\,n) \\
& \xRightarrow{\texttt{r}.@()} \Rightarrow & \nu n\,.\,(n, \nu n'\,.\,n') \\
& \xRightarrow{\texttt{l}.\nu n} & \nu n'\,.\,((), n')
\end{array}$$

At this point the term cannot match the last $\xRightarrow{\texttt{r}.n}$ transition performed by the first term because its only move is:

$$\nu n'\,.\,((), n') \xRightarrow{\texttt{r}.\nu n'} ((), ())$$

Note that this example relies crucially on the use of `copy`, $\texttt{l}.\gamma$ and using some syntax sugar such as:

$$\texttt{let}\, n = t\, \texttt{in}\, t' \overset{\text{def}}{=} \texttt{let}\, x = t\, \texttt{in}\, (t'[x/n])$$

The result then follows by induction on $\overline{\gamma}$. $\square$

Theorem . **(completeness for ν-calculus)** *If* $\Gamma;\Delta \vDash t \approx_{ctx} t' : \sigma$ *then* $\Gamma;\Delta \vDash t \approx^{\circ} t' : \sigma$.

. Partial soundness

It is a fairly simple matter to show that bisimulation is sound for the ν-calculus at first order, by but in order to complete the diagram we need to know that $\approx^{\bullet}$ is despite the fact that some ‘foreign’ code f is being applied to n. By adding assignment, f can leak the secret name n to the environment.

We believe that any form of side-effect which allows secrets to leak like this will help to make bisimulation sound and complete, for example call-cc, communication channels or imperative objects. Although the extent to which any additional features are required is as yet unclear. We have chosen to investigate global assignment as it is the simplest addition which is still deterministic and terminating.

.1 Syntax and type rules

Extend the grammar of terms by:

$$t ::= \cdots \mid r := v \,.\, t \mid ?r$$

where r ranges over an infinite set of ***references***. These operations allow a name to be written to, or read from, a reference. We do not introduce references themselves as values and thus have no need for introducing a type of references.

We introduce a ***use-def*** type system to ensure that all references are written to before they (where the bound names in d do not clash with free names in t') and:

$$\frac{t_1 \Rrightarrow t_2}{\mathsf{E}[t_1] \Rrightarrow \mathsf{E}[t_2]}$$

Let $\equiv$ be the least equivalence generated by $\Rrightarrow$.

Extend the evaluation contexts to include assignment:

$$\mathsf{E} ::= \cdots \mid r := v \,.\, \mathsf{E}$$

Extend the reduction semantics with a rule for dereferencing:

$$r := n \,.\, ?r \quad \xrightarrow{\tau} \quad r := n \,.\, n$$

Since we have modified the prevalues, we need to modify the let-β rule:

$$\mathtt{let}\, x \;=\; d \,.\, v\, \mathtt{in}\, t \quad \xrightarrow{\tau} \quad d \,.\, t[v/x]$$

Add a structural equivalence rule:

$$\frac{t_1 \equiv t_2 \quad t_2 \xrightarrow{\tau} t_3 \quad t_3 \equiv t_4}{t_1 \xrightarrow{\tau} t_4}$$

The definition of contextual equivalence remains the same, except that the results of a test can include some assignments: $t_1 \approx_{ctx} t_2$ whenever for all ref-closing contexts C of type `bool`, we have $\mathsf{C}[t_1] \Rightarrow d_1 \,.\, \mathtt{true}$ iff $\mathsf{C}[t_2] \Rightarrow d_2 \,.\, \mathtt{true}$.

Lemma .1 *Any derivation $t \xrightarrow{\tau} t'$ can be deduced $t \Rrightarrow t'' \xrightarrow{\tau} t''' \equiv t'$ where $t'' \xrightarrow{\tau} t'''$ can be deduced without using structural equivalence.*

Proof A simple analysis of the rules which generate $\Rrightarrow$ suffices to show that any reduction which may occur on the left-hand side of a rule may also occur on the right, so naught is to be gained by *cooling*. $\square$

The reader may like to note that the νref-calculus contains closed terms which may not necessarily converge to a prevalue, such as $\mathtt{let}\; x \;=\; ?r \;\mathtt{in}\; t$. However, all such terms are ref-open, and our reduction semantics is only used for ref-closed terms.

. Labelled transition system semantics

We need to provide a semantics for terms with references, so judgements are now of the form $(\Delta;\mathrm{R};\mathrm{W} \vdash p : \sigma) \xrightarrow{\gamma} (\Delta, \Delta';\mathrm{R};\mathrm{W} \vdash t : \sigma')$. Note that since terms cannot generate new references, that the reference environments are not changed by transitions.

Extend the grammar of labels with:

$$\gamma ::= \cdots \mid r{:=}n \mid ?r$$

The new transitions allow a name to be assigned:

$$(\Delta;\mathrm{R};\, \vdash () : \mathtt{unit}) \xrightarrow{r:=n} (\Delta;\mathrm{R};\, \vdash r := n \,.\, () : \mathtt{unit}) \qquad (\text{where } n \in \Delta)$$

and to be read:

$$(\Delta; \mathrm{R};\ \vdash () : \mathtt{unit}) \xrightarrow{?r} (\Delta; \mathrm{R};\ \vdash ?r : \mathtt{name}) \qquad (\text{where } r \in \mathrm{R})$$

We weaken the side-condition on application to allow the argument to include free references:

$$(\Delta; \mathrm{R};\ \vdash v : \sigma \to \sigma') \xrightarrow{@v'} (\Delta; \mathrm{R};\ \vdash v(v') : \sigma') \qquad (\text{where } \Delta; \mathrm{R};\ \vdash v' : \sigma)$$

Transitions are allowed in assignment contexts:

$$\frac{(\Delta; \mathrm{R} \cup r; \mathrm{W} \setminus r \vdash p : \sigma) \xrightarrow{\gamma} (\Delta, \Delta'; \mathrm{R} \cup r; \mathrm{W} \setminus r \vdash t : \sigma')}{(\Delta; \mathrm{R}; \mathrm{W} \vdash}$$

Proposition .1 *If* Π *is passive in* $(\Gamma;\Delta;\mathrm{R};\vdash v:\sigma)$ *and in* $(\Gamma$

(b) n is passive in p_1, so p_2 can match it by ignoring the name (which is added to the passive name environment Π).

3. Π only contains passive names.

Overt bisimulation is a partial equivalence relation, and we can show a generalization of transitivity, as evidenced in the following lemma.

Lemma . *If* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \approx_o^{\Pi_0,\Pi_1 \circ} \approx_o^{\Pi_0,\Pi_2} u : \sigma$ *then* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \approx_o^{\Pi_0,\Pi_1,\Pi_2 \circ} u : \sigma$.

Proof. It suffices to show the result for ref-closed terms, since we can then close up under all closing substitutions and ref-closing assignments. Define:

$$R^{\Pi'} = \left\{ (t,u) \mid t \approx_o^{\Pi_0,\Pi_1} \approx_o^{\Pi_0,\Pi_2} u, \text{ and } \Pi' = \Pi_0, \Pi_1, \Pi_2 \right\}.$$

It is not difficult to check that R forms an overt bisimulation. □

Since an overt simulation is a simulation, it is easy to see that $\approx_o$ is a finer relation than $\approx$. In fact, we can show that overt bisimulation coincides with bisimulation.

Proposition . $\approx$ *is the same as* $\approx_o$

Proof. Define $\Delta,\Pi;;\mathrm{W} \vDash t_1 \approx^{\Pi} t_2 : \sigma$ whenever $\Delta;;\mathrm{W} \vDash \nu\Pi \,.\, t_1 \approx \nu\Pi \,.\, t_2 : \sigma$ and Π is passive in t_1 and t_2. It is routine to verify that this is an overt bisimulation, and that it coincides with bisimulation when Π is empty. □

. Congruence of overt bisimulation

The proof that overt bisimulation is a congruence uses Howe's technique, but the definition of $\approx^{\bullet}$ is rather more complex, since we have to allow names to move between the passive and active name environments.

Define $\approx_o^{\Pi\bullet}$ by two rules:

$$\frac{\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t_1 \widehat{\approx_o^{\Pi\bullet}} t_2 \qquad \Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t_2 \approx_o^{\Pi,\Pi'\circ} t_3}{\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t_1 \approx_o^{\Pi,\Pi'\bullet} t_3}$$

and:

$$\frac{\Gamma;\Delta,n;\mathrm{R};\mathrm{W} \vDash t_1 \approx_o^{\Pi,n\bullet} t_2 \qquad \Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash \nu n \,.\, t_2 \approx_o^{\Pi\circ} t_3}{\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash \nu n \,.\, t_1 \approx_o^{\Pi\bullet} t_3}$$

This relation satisfies the usual [6] properties required of this relation, that is it contains the ˆ closure of itself and it contains overt bisimulation.

Lemma . '*If* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \approx_o^{\Pi_0,\Pi_1\bullet} \approx_o^{\Pi_0,\Pi_2\circ} u : \sigma$ *then* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \approx_o^{\Pi_0,\Pi_1,\Pi_2\bullet} u : \sigma$.

Proof. Suppose $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \approx_o^{\Pi_0,\Pi_1\bullet} t_0 \approx_o^{\Pi_0,\Pi_2} u : \sigma$ and proceed by induction on the structure of t. There are two main cases to consider based on how the Howe relation decomposes.

FirstlydSuppose:T/R-16-1.997880(t)00-114435o1212Tf60Td[71

Secondly, consider the case in which t is $\nu n\,.\,t'$ and the latter Howe rule is used. This means that there is some t'_0 such that

$$\Gamma; \Delta, n; \mathrm{R}; \mathrm{W} \vDash t' \approx_o^{\Pi_0,\Pi_1,n\bullet} t'_0 : \sigma \quad \text{and} \quad \Gamma; \Delta; \mathrm{R}; \mathrm{W} \vDash \nu n\,.\,t'_0 \approx_o^{\Pi_0,\Pi_1{}^\circ} t_0 : \sigma.$$

We can apply the induction hypothesis to

$$\Gamma; \Delta, n; \mathrm{R}; \mathrm{W} \vDash t' \approx_o^{\Pi_0,\Pi_1,n\bullet} t'_0 \approx_o^{\Pi_0,\Pi_1,\Pi_2,n} t'_0 : \sigma$$

to yield $\Gamma; \Delta, n; \mathrm{R}; \mathrm{W} \vDash t' \approx_o^{\Pi_0,\Pi_1,\Pi_2,n\bullet} t'_0 : \sigma$, and use Lemma 5.2 again to obtain

$$\Gamma; \Delta; \mathrm{R}; \mathrm{W} \vDash \nu n\,.\,t'_0 \approx_o^{\Pi_0,\Pi_1,\Pi_2{}^\circ} u : \sigma.$$

From here we apply the second Howe rule to finish. □

First, we show some technical lemmas, which extend obvious properties of bisimulation on for some value v_3. By Lemma 5.6 we have:

$$\Gamma; \Delta, \overline{n}; \mathrm{R}; \vDash v_3 \approx_o^{\Pi, \overline{n}^{\circ}} v_2 : \sigma$$

so by weakening and definition of $\approx_o^{\Pi^{\bullet}}$ we have:

$$\Gamma; \Delta, \overline{n}; \mathrm{R}; \vDash v_1 \approx_o^{\Pi, \overline{n}^{\bullet}_{\mathrm{o}}} v$$

1. $d_2.v_2 \equiv \nu n_0 \,.\, d_5.v_4$ and $\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\bar{r} \vDash d_4.v_3 \approx_o^{\Pi^\circ} d_5.v_4 : \sigma$.
   In this case, Lemma 5.4 gives us:

$$\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\bar{r} \vDash d_3.v_1 \approx_o^{\Pi^\bullet} d_5.v_4 : \sigma$$

   so by induction:

$$\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\mathrm{W}' \vDash d_3.$$

2. $\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\overline{r}\vDash d_4.v_3 \approx_o^{\Pi,n_0{}^\circ} d_2.v_2 : \sigma$.

   In this case, Lemma 5.4 gives us:

$$\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\overline{r}\vDash d_3.v_1 \approx_o^{\Pi,n_0{}^\bullet} d_2.v_2 : \sigma$$

   so by induction:

$$\Gamma;\Delta,n_0;\mathrm{R};\mathrm{W}\cup\mathrm{W}'\vDash d_3.t_1[v_1/x] \approx_o^{\Pi,n_0{}^\bullet} d_2.t_2[v_2/x] : \sigma'$$

   and so:

$$\Gamma;\Delta;\mathrm{R};\mathrm{W}\cup\mathrm{W}'\vDash \nu n_0\,.\,d_3.t_1[v_1/x] \widehat{\approx_o^{\Pi^\bullet}} \nu n_0\,.\,d_2.t_2[v_2/x] \approx_o^{\Pi^\circ} d_2.t_2[v_2/x] : \sigma'$$

   (where the latter equivalence holds since n_0 does not occur free in $d_2.t_2[v_2/x]$) and so we can use the definition of $\approx_o^{\Pi^\bullet}$ to conclude. □

We can then show that $\approx^\bullet$ is a bisimulation up to $(\equiv,=)$ [19], from which it is routine to show that overt bisimulation, and hence bisimulation, is a congruence.

Proposition .10 *On ref-closed terms, $\approx_o{}^\bullet$ is an overt bisimulation up to $(\equiv,=)$.*

Proof. Take $\Delta;;\mathrm{W}\vDash t \approx_o^{\Pi^\bullet} u : \sigma$. It is fairly easy to see that the latter two conditions for being an overt bisimulation are satisfied, and we concentrate on showing that any transition of t can be matched by a transition of u.

We will show a slightly more general result, which is that if:

$$\Delta;\overline{r};\mathrm{W}\setminus\overline{r}\vDash t \approx_o^{\Pi^\bullet} u : \sigma \qquad (\Delta;;\mathrm{W}\vdash \overline{r}:=\overline{n}\,.\,t:\sigma) \xrightarrow{\alpha} (\Delta,\Delta';;\mathrm{W}\vdash t':\sigma)$$

then we can find u' such that:

$$\Delta,\Delta';;\mathrm{W}\vDash t' \equiv\approx_o^{\Pi^\bullet} u' : \sigma' \qquad (\Delta;;\mathrm{W}\vdash \overline{r}:=\overline{n}\,.\,u:\sigma) \overset{\hat{\alpha}}{\Longrightarrow} (\Delta,\Delta';;\mathrm{W}\vdash u':\sigma')$$

In particular, note that we can take $\overline{r}$ to be empty and get the desired result.

We proceed by induction on the proof of $\approx_o^{\Pi^\bullet}$. For most of the cases this is a completely standard rule induction so we only detail the situations which vary from the usual approach.

In fact, we shall prove this property for a variant transition system, wher

Since we have:

$$(\Delta, n_0;;\mathrm{W} \vdash \overline{r} := \overline{n} \,.\, t : \sigma) \xrightarrow{\iota.n_0} (\Delta, n_0;;\mathrm{W} \vdash \overline{r} :$$

Case Suppose $(\Delta;;\mathrm{W} \vdash \overline{r} := \overline{n}\,.\,t : \sigma) \xrightarrow{\tau} (\Delta;;\mathrm{W} \vdash t' : \sigma)$. The most interesting case occurs when this is an instance of the let block β-reduction, that is:

$$(\Delta;;\mathrm{W} \vdash \overline{r} := \overline{n}\,.\,\mathtt{let}\,x\;=\;d_1.v_1\;\mathtt{in}\,t_1 : \sigma) \xrightarrow{\tau} (\Delta;;\mathrm{W} \vdash \overline{r} := \overline{n}\,.\,d_1.t_1[v_1/x] : \sigma)$$

We know, by definition of the Howe relation, that there exists some t_0, t_2 such that:

$$\Delta;\overline{r};\mathrm{W}' \cup \overline{r}' \vDash d_1.v_1 \approx_o^{\Pi'^\bullet} t_0 : \sigma' \qquad x:\sigma';\Delta;\overline{r}\cup\overline{r}';\mathrm{W}'' \vDash t_1 \approx_o^{\Pi'^\bullet} t_2 : \sigma$$
$$\Delta;\overline{r};\mathrm{W}\setminus\overline{r} \vDash \mathtt{let}\,x\;=\;t_0\;\mathtt{in}\,t_1 \approx_o^{\Pi} u : \sigma$$

for some $\Pi' \subseteq \Pi$, and $\mathrm{W}' \cup \mathrm{W}'' = \mathrm{W}\setminus\overline{r}$. Since:

$$(\Delta;;\mathrm{W}' \cup \overline{r}' \vdash \overline{r} := \overline{n}\,.\,d_1.v_1 : \sigma') \xrightarrow{\mathtt{id}} (\Delta;;\mathrm{W}' \cup \overline{r}' \vdash \overline{r} := \overline{n}\,.\,d_1.v_1 : \sigma')$$

by induction we can find d_2 and v_2 such that:

$$(\Delta;;\mathrm{W}' \cup \overline{r}' \vdash \overline{r} := \overline{n}\,.\,t_0 : \sigma') \xRightarrow{\mathtt{id}} (\Delta;;\mathrm{W}' \cup \overline{r}' \vdash d_2.v_2 : \sigma')$$

and

$$\Delta;;\mathrm{W}' \cup \overline{r}' \vDash \overline{r} := \overline{n}\,.\,d_1.v_1 \equiv\approx_o^{\Pi'^\bullet} d_2.v_2 : \sigma'$$

and so:

$$\begin{array}{rcl}
(\Delta;;\mathrm{W} \vdash \overline{r} := \overline{n}\,.\,\mathtt{let}\,x\;=\;t_0\;\mathtt{in}\,t_1 : \sigma) & \equiv & (\Delta;;\mathrm{W} \vdash \mathtt{let}\,x\;=\;\overline{r} := \overline{n}\,.\,t_0\;\mathtt{in}\,t_1 : \sigma) \\
& \Rightarrow & (\Delta;;\mathrm{W} \vdash \mathtt{let}\,x\;=\;d_2.v_2\;\mathtt{in}\,t_1 : \sigma) \\
& \xrightarrow{\tau} & (\Delta;;\mathrm{W} \vdash d_2.t_1[v_2/x] : \sigma)
\end{array}$$

and so we can find a u' such that:

$$\Delta;;\mathrm{W} \vDash d_2.t_1[v_2/x] \approx_o^{\Pi} u' : \sigma' \qquad (\Delta;;\mathrm{W} \vdash \overline{r} := \overline{n}\,.\,u : \sigma) \Rightarrow (\Delta;;\mathrm{W} \vdash u' : \sigma)$$

We can now apply Proposition 5.9 to observe that:

$$\Delta;;\mathrm{W} \vDash \overline{r} := \overline{n}\,.\,d_1.t_1[v_1/x] \equiv\approx_o^{\Pi'^\bullet} d_2.t_2[v_2/x] \approx_o^{\Pi} u' : \sigma$$

and we use Lemma 5.4 to finish.

Case We demonstrate how the Howe relation is preserved by structural congruence. In fact, we know by Lemma 4.1 that we need only consider the heating rules and show that if $t \Rrightarrow t'$ and $t \approx_o{}^\bullet u$ then $t' \approx_o{}^\bullet u$ also. We use the following case as a typical example. Suppose:

$$(\Delta;\mathrm{R};\mathrm{W} \vdash r := n\,.\,\mathtt{let}\,x\;=\;t\;\mathtt{in}\,t' : \sigma) \Rrightarrow (\Delta;\mathrm{R};\mathrm{W} \vdash \mathtt{let}\,x\;=\;r := n\,.\,t\;\mathtt{in}\,t' : \sigma)$$

We know that there is some t_0 such that:

$$\Delta;\mathrm{R}\cup r;\mathrm{W}\setminus r \vDash \mathtt{let}\,x\;=\;t\;\mathtt{in}\,t' \approx_o^{\Pi'^\bullet} t_0 : \sigma \qquad \Delta;\mathrm{R};\mathrm{W} \vDash r := n\,.\,t_0 \approx_o^{\Pi^\circ} u : \sigma$$

where $\Pi' \subseteq \Pi$. We decompose the former further to obtain terms t_0' and t_0'' and W, W'' such that $\mathrm{W}' \cup \mathrm{W}'' = \mathrm{W}$ and:

$$\Delta;\mathrm{R}\cup r;\mathrm{W}'\setminus r \vDash t \approx_o^{\Pi''^\bullet} t_0' : \sigma' \qquad x:\sigma;\Delta;\mathrm{R}\cup r;\mathrm{W}'' \vDash t' \approx_o^{\Pi''^\bullet} t_0'' : \sigma$$
$$\Delta;\mathrm{R}\cup r;\mathrm{W}\setminus r \vDash \mathtt{let}\,x\;=\;t_0'\;\mathtt{in}\,t_0'' \approx_o^{\Pi'^\circ} t_0 : \sigma$$

We observe that $\approx_o$ is easily seen to be congruent with respect to assignment so we can obtain:

$$\begin{array}{rcl}
\Delta; \mathrm{R}; \mathrm{W} \vDash \texttt{let}\ x\ =\ r := n\ .\ t\ \texttt{in}\ t' & \approx_o^{\Pi''^{\bullet}} & \texttt{let}\ x\ =\ r := n\ .\ t'_0\ \texttt{in}\ t''_0 \\
& \equiv & r := n\ .\ \texttt{let}\ x\ =\ t'_0\ \texttt{in}\ t''_0 \\
& \approx_o^{\Pi'^{\circ}} & r := n\ .\ t_0 \\
& \approx_o^{\Pi^{\circ}} & u : \sigma
\end{array}$$

and so we are finished. □

Corollary .11 $\approx^\circ$ *is a congruence for the* ν*ref-calculus.*

. Comments

Earlier in the paper we described the logical relations of [14] as an *overt* proof technique for ν-calculus. We can see now that there are similarities between our overt bisimulation and the logical relations. In particular, both techniques make use of a predicate to track the private names

Extend the definition of passivity from terms to capture-free evaluation contexts E with typing:

$$\frac{\Gamma;\Delta;\mathrm{R}';\mathrm{W}' \vdash \cdot : \sigma'}{\Gamma;\Delta;\mathrm{R};\mathrm{W} \vdash \mathsf{E}[\cdot] : \sigma}$$

Define $\overline{n}$ are passive from $\Psi \subseteq \Theta$ in E iff $\overline{n}$ are passive from $\Psi \subseteq \Theta$ in $\Gamma;\Delta,\Delta';\mathrm{R};\mathrm{W} \vdash \mathsf{E}[t] : \sigma$ for any $\Gamma;\Psi,\Delta';\mathrm{R}';\mathrm{W}' \vdash t : \sigma'$.

A partial equivalence relation (PER) on names R is a transitive, symmetric relation. We shall write $R : \overline{n} \leftrightarrow \overline{n}$ whenever $\overline{n}$ is the domain of R.

Given a PER R: $\overline{n} \leftrightarrow \overline{n}$, define the *passive bisimulation* $\sim_R$ to be the type-indexed relation given by:

- If $\overline{n}$ are passive from $\Psi \subseteq \Theta$ in $(\Gamma;\Delta;\mathrm{R};\mathrm{W} \vdash t : \sigma)$

  then $\Gamma;\Psi \subseteq \Theta \subseteq \Delta;\mathrm{R};\mathrm{W} \vDash t \sim_R t : \sigma$.

- If $\Gamma;\Psi \subseteq \Theta \subseteq \Delta;\mathrm{R};\vDash v \sim_R v' : \sigma$ and $\Gamma,x:\sigma;\Psi \subseteq \Theta \subseteq \Delta;\mathrm{R};\mathrm{W} \vDash t \sim_R t' : \sigma'$

  then $\Gamma;\Psi \subseteq \Theta \subseteq$

(where Π, Π' are not free in $\overline{\gamma}$) as:

$$\begin{tikzcd}
(\Delta, \Delta';;\mathrm{W} \vdash t[v/x] : \sigma') \arrow[r, leftrightarrow, "\sim_R"] \arrow[d, Rightarrow, "\overline{\gamma}"'] & (\Delta, \Delta';;\mathrm{W} \vdash t[v[\Pi'/\Pi]/x] : \sigma') \arrow[d, Rightarrow, "\overline{\gamma}"'] \\
\cdot \arrow[r, leftrightarrow, "\sim_R"] & \cdot
\end{tikzcd}$$

so by Proposition A.2 we have that Π is passive in $t[v/x]$. □

This proof relies on the fact that if $t \sim_R u$ then t cannot perform a n transition for any n in the domain of R:

Proposition A. *For any $R : \overline{n} \leftrightarrow \overline{n}$, if $\Delta; \mathrm{R}; \mathrm{W} \vDash t \sim_R u : \sigma$ and $(\Delta; \mathrm{R}; \mathrm{W} \vdash t : \sigma) \xrightarrow{\iota.n}$ then $n \notin \overline{n}$.*

Proof. A straightforward induction on the proof of $\sim_R$. □

Proposition A. *If* $\Gamma;\Psi \subseteq \Theta \subseteq \Delta;\mathrm{R};\mathrm{W} \vDash t \sim_R u : \sigma$ *then* $\Gamma;\Psi \subseteq \Theta \subseteq \Delta;\mathrm{R};\mathrm{W} \vDash t \sim_R^1 u : \sigma$.

Proof. We proceed by induction on the proof of $\sim_R$ and notice that the type index $\Psi \subseteq \Theta$ plays

Case $t = x$, so the result follows by the definition of $\sim_R^1$.

Case $(\Delta; \mathrm{R}; \mathrm{W} \vdash t[$

Case. The transition is a reduction of the form:

$$(\Delta; R; W \vdash \mathsf{E}[x = x'][\overline{v}/\overline{x}] : \sigma) \xrightarrow{\tau} (\Delta; R; W \vdash \mathsf{E}[b][\overline{v}/\overline{x}] : \sigma)$$

which is handled similarly to the previous case.

Case. The transition is a reduction of the form:

$$(\Delta; R; W \vdash \mathsf{E}[\mathtt{if}\ x\ \mathtt{then}\ t_1\ \mathtt{else}\ t_2][$$

Proposition A. *For any* E *with* $\overline{n}$ *passive typed:*

$$\frac{\Gamma;\Delta;\mathrm{R}';\mathrm{W}' \vdash \cdot : \sigma'}{\Gamma;\Delta;\mathrm{R};\mathrm{W} \vdash \mathsf{E}[\cdot] : \sigma}$$

if $\Gamma;\Delta;\mathrm{R}';\mathrm{W}' \vDash t \sim^1_R t' : \sigma'$ *then* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash \mathsf{E}[t] \sim^1_R \mathsf{E}[t'] : \sigma$.

Proof. Similar to the proof of Proposition A.4 with the addition of two cases which arise as an interaction between E and t.

Case. E is $\mathsf{E}_1[r := v.\,[\cdot]]$ and t is $\mathsf{E}_2[?r]$ so that

$$\Gamma;\Delta;\mathrm{R}',r;\mathrm{W}' \vDash \mathsf{E}_2[?r] \sim_R \mathsf{E}'_2[?r]$$

with r not assigned to in $\mathsf{E}_2, \mathsf{E}'_2$. We observe that v cannot be a name in $\overline{n}$ and we easily get $\Gamma;\Delta;;\vDash v \sim_R v$. By definition of $\sim_R$ it follows that

$$\Gamma, y : \mathtt{name};\Delta;\mathrm{R}',r;\mathrm{W}' \vDash \mathsf{E}_2[y] \sim_R \mathsf{E}'_2[y]$$

because any n which can be instantiated for y can be supplied to $?r$ using a closing assignment. Given this it is a simple matter to use the definition of $\sim_R$ to yield

$$\Gamma, y : \mathtt{name};\Delta;\mathrm{R};\mathrm{W}' \vDash \mathsf{E}[\mathsf{E}_2[y] \sim_R \mathsf{E}[\mathsf{E}'_2[y]$$

and the result follows.

Case. E is $\mathsf{E}'[\mathtt{let}\ x = [\cdot]\ \mathtt{in}\ u]$ and t is $d_1.v_1$ so that $\Gamma;\Delta;\mathrm{R}';\mathrm{W}' \vDash d_1.v_1 \sim_R d_2.v_2$. We use Proposition A.10 to observe that $\Gamma;\Delta,\Delta';\mathrm{R}',\mathrm{R}'';\vDash v_1 \sim_R v_2$ for appropriate Δ',R''. It is easy to see that the hypothesis tells us that $\overline{n}$ are passive in $\Gamma, x : \sigma';\Delta;\mathrm{R};\mathrm{W} \vdash \mathsf{E}'[u]$ therefore

$$\Gamma;\Delta,\Delta';\mathrm{R},\mathrm{R}'',\mathrm{W} \vDash \mathsf{E}'[u[v_1/x]] \sim_R \mathsf{E}'[u[v_2/x].$$

We use the definition of $\sim_R$ to obtain

$$\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash d_1.\mathsf{E}'[u[v_1/x] \sim_R d_2.\mathsf{E}'[u[v_2/x]$$

and structural congruence to finish. □

Proposition A. *If* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \sim^1_R t' : \sigma$ *and* $[\overline{n}'/\overline{n}] \subseteq R : \overline{n} \leftrightarrow \overline{n}$ *is a bijective substitution then* $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \sim^1_R t'[\overline{n}'/\overline{n}] : \sigma$.

Proof. For closed terms, this goes through immediately, since transitions are invariant under bijective substitutions.

For open terms, consider any substitutions $\Delta;\mathrm{R};\, \vDash [\overline{v}/\overline{x}] \sim^1_R [\overline{w}/\overline{x}] : \Gamma$. Since $\overline{v}$ and $\overline{w}$ are closed, we have that:

$$\Delta;\mathrm{R};\, \vDash [\overline{v}/\overline{x}] \sim^1_R [\overline{w}[\overline{n}/\overline{n}']/\overline{x}] : \Gamma$$

and so since $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vDash t \sim^1_R t' : \sigma$ we have:

$$\Delta;\mathrm{R};\mathrm{W} \vDash t[\overline{v}/\overline{x}] \sim^1_R t'[\overline{w}[\overline{n}/\overline{n}']/\overline{x}] : \sigma$$

and again we have closed terms, so:

$$\Delta;\mathrm{R};\mathrm{W} \vDash t[\overline{v}/\overline{x}] \sim^1_R t'[\overline{w}[\overline{n}/\overline{n}']/\overline{x}][\overline{n}'/\overline{n}] = t'[\overline{n}'/\overline{n}][\overline{w}/\overline{x}] : \sigma$$

as required. □

Proposition A. *If $\overline{n}$ are passive from $\Psi \subseteq \Theta$ in $\Gamma;\Delta;\mathrm{R};\mathrm{W} \vdash \mathsf{E}[x(v)] : \sigma$ then $\overline{n}$ are passive from $\Psi \subseteq \Theta$ in E and $\Gamma;\Delta;\mathrm{R}';\vdash v : \sigma'$.*

Proof.