A COMPLETE TYPE CHECKING OF EXTENSIONALITY EXAMPLE

$$
\dfrac{
\begin{array}{c}
\dfrac{
\begin{array}{c}
\dfrac{
\begin{array}{c}
\dfrac{
\begin{array}{c}
\dfrac{
\dfrac{
\dfrac{
\Gamma(\mathsf{funext}) \;=\; \forall a\,b.\mathsf{Eq}\ b \Rightarrow f:(a\to b)\to g:(a\to b)\to(x:a\to\{f\ x == g\ x\})\to\{f \simeq g\}
}{
\Gamma\vdash\mathsf{funext} :: \forall a\,b.\mathsf{Eq}\ b \Rightarrow f:(a\to b)\to g:(a\to b)\to(x:a\to\{f\ x == g\ x\})\to\{f \simeq g\}
}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\} :: \forall b.\mathsf{Eq}\ b \Rightarrow f:(\{v:\alpha\mid\kappa_\alpha\}\to b)\to g:(\{v:\alpha\mid\kappa_\alpha\}\to b)\to(x:\{v:\alpha\mid\kappa_\alpha\}\to\{f\ x == g\ x\})\to\{f \simeq g\}
}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\}\ @\{v:\beta\mid\kappa_\beta\} :: \mathsf{Eq}\ \{v:\beta\mid\kappa_\beta\} \Rightarrow f:(\{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\})\to g:(\{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\})\to(x:\{v:\alpha\mid\kappa_\alpha\}\to\{f\ x == g\ x\})\to\{f \simeq g\}
}
\\
\dfrac{\Gamma(\mathsf{d}) \;=\; \mathsf{Eq}\ \alpha}{\Gamma\vdash\mathsf{d} :: \mathsf{Eq}\ \alpha}
\qquad\qquad
\dfrac{}{\Gamma\vdash\mathsf{Eq}\ \alpha \;\preceq\; \mathsf{Eq}\ \{v:\beta\mid\kappa_\beta\}}\;\text{Sub-D}
\end{array}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\}\ @\{v:\beta\mid\kappa_\beta\}\ \mathsf{d} :: f:(\{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\})\to g:(\{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\})\to(x:\{v:\alpha\mid\kappa_\alpha\}\to\{f\ x == g\ x\})\to\{f \simeq g\}
}
\\
\dfrac{\Gamma(\mathsf{h}) \;=\; x:\{v:\alpha\mid d_{\mathsf{h}}\}\to\{v:\beta\mid r_{\mathsf{h}}\}}{\Gamma\vdash\mathsf{h} :: x:\{v:\alpha\mid d_{\mathsf{h}}\}\to\{v:\beta\mid r_{\mathsf{h}}\}}
\qquad\qquad
\dfrac{\cdots}{\Gamma\vdash x:\{v:\alpha\mid d_{\mathsf{h}}\}\to\{v:\beta\mid r_{\mathsf{h}}\} \;\preceq\; \{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\}}\;\text{Sub-H}
\end{array}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\}\ @\{v:\beta\mid\kappa_\beta\}\ \mathsf{d}\ \mathsf{h} :: g:(\{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\})\to(x:\{v:\alpha\mid\kappa_\alpha\}\to\{\mathsf{h}\ x == g\ x\})\to\{\mathsf{h} \simeq g\}
}
\\
\dfrac{\Gamma(\mathsf{k}) \;=\; x:\{v:\alpha\mid d_{\mathsf{k}}\}\to\{v:\beta\mid r_{\mathsf{k}}\}}{\Gamma\vdash\mathsf{k} :: x:\{v:\alpha\mid d_{\mathsf{k}}\}\to\{v:\beta\mid r_{\mathsf{k}}\}}
\qquad\qquad
\dfrac{\cdots}{\Gamma\vdash x:\{v:\alpha\mid d_{\mathsf{k}}\}\to\{v:\beta\mid r_{\mathsf{k}}\} \;\preceq\; \{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\}}\;\text{Sub-K}
\end{array}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\}\ @\{v:\beta\mid\kappa_\beta\}\ \mathsf{d}\ \mathsf{h}\ \mathsf{k} :: (x:\{v:\alpha\mid\kappa_\alpha\}\to\{\mathsf{h}\ x == \mathsf{k}\ x\})\to\{\mathsf{h} \simeq \mathsf{k}\}
}
\\
\dfrac{\Gamma(\mathsf{lemma}) \;=\; x:\alpha\to\{p\}}{\Gamma\vdash\mathsf{lemma} :: x:\alpha\to\{p\}}
\qquad\qquad
\dfrac{\cdots}{\Gamma\vdash x:\alpha\to\{p\} \;\preceq\; x:\{v:\alpha\mid\kappa_\alpha\}\to\{\mathsf{h}\ x == \mathsf{k}\ x\}}\;\text{Sub-L}
\end{array}
}{
\Gamma\vdash\mathsf{funext}\ @\{v:\alpha\mid\kappa_\alpha\}\ @\{v:\beta\mid\kappa_\beta\}\ \mathsf{d}\ \mathsf{h}\ \mathsf{k}\ \mathsf{lemma} :: \{h \simeq k\}
}
$$

$$
\dfrac{
\dfrac{\kappa_\alpha \Rightarrow d_{\mathsf{h}}}{\Gamma\vdash\{v:\alpha\mid\kappa_\alpha\} \;\preceq\; \{v:\alpha\mid d_{\mathsf{h}}\}}
\qquad
\dfrac{\kappa_\alpha \Rightarrow r_{\mathsf{h}} \Rightarrow \kappa_\beta}{\Gamma, x:\{v:\alpha\mid\kappa_\alpha\}\vdash\{v:\beta\mid r_{\mathsf{h}}\} \;\preceq\; \{v:\beta\mid\kappa_\beta\}}
}{
\Gamma\vdash x:\{v:\alpha\mid d_{\mathsf{h}}\}\to\{v:\beta\mid r_{\mathsf{h}}\} \;\preceq\; \{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\}
}\;\text{Sub-H}
$$

$$
\dfrac{
\dfrac{\kappa_\alpha \Rightarrow d_{\mathsf{k}}}{\Gamma\vdash\{v:\alpha\mid\kappa_\alpha\} \;\preceq\; \{v:\alpha\mid d_{\mathsf{k}}\}}
\qquad
\dfrac{\kappa_\alpha \Rightarrow r_{\mathsf{k}} \Rightarrow \kappa_\beta}{\Gamma, x:\{v:\alpha\mid\kappa_\alpha\}\vdash\{v:\beta\mid r_{\mathsf{k}}\} \;\preceq\; \{v:\beta\mid\kappa_\beta\}}
}{
\Gamma\vdash x:\{v:\alpha\mid d_{\mathsf{k}}\}\to\{v:\beta\mid r_{\mathsf{k}}\} \;\preceq\; \{v:\alpha\mid\kappa_\alpha\}\to\{v:\beta\mid\kappa_\beta\}
}\;\text{Sub-K}
$$

$$
\dfrac{
\dfrac{\kappa_\alpha \Rightarrow \mathsf{true}}{\Gamma\vdash\{v:\alpha\mid\kappa_\alpha\} \;\preceq\; \alpha}
\qquad
\dfrac{\kappa_\alpha \Rightarrow p \Rightarrow h\ x == k\ x}{\Gamma, x:\{v:\alpha\mid\kappa_\alpha\}\vdash\{p\} \;\preceq\; \{h\ x == k\ x\}}
}{
\Gamma\vdash x:\alpha\to\{p\} \;\preceq\; x:\{v:\alpha\mid\kappa_\alpha\}\to\{\mathsf{h}\ x == \mathsf{k}\ x\}
}\;\text{Sub-L}
$$

Fig. 1. Complete type checking of naïve extensionality in thEq.

$$\begin{array}{rcll} \textit{Expressions} & e & ::= & \text{as in } \lambda^{RE} \\ \textit{Types} & t & ::= & \mathsf{Bool} \mid () \mid \mathsf{PBEq}_t \mid t \to t \\ \textit{Typing Environment} & G & ::= & \emptyset \mid G, x : t \end{array}$$

Basic Type checking $\boxed{G \vdash_B e :: t}$

$$\frac{}{G \vdash_B c :: \lfloor \mathsf{TyCons}(c) \rfloor}\ \textsc{BT-Con} \qquad \frac{x : t \in G}{G \vdash_B x :: t}\ \textsc{BT-Var}$$

$$\frac{G \vdash_B e :: t_x \to t \qquad G \vdash_B e_x :: t_x}{G \vdash_B e\ e_x :: t}\ \textsc{BT-App} \qquad \frac{G, x : \lfloor \tau_x \rfloor \vdash_B e :: t}{G \vdash_B \lambda x{:}\tau_x .\ e :: \lfloor \tau_x \rfloor \to t}\ \textsc{BT-Lam}$$

$$\frac{\begin{array}{c} G \vdash_B e :: () \\ G \vdash_B e_1 :: b \qquad G \vdash_B e_2 :: b \end{array}}{G \vdash_B \mathsf{bEq}_b\ e_1\ e_2\ e :: \mathsf{PBEq}_b}\ \textsc{BT-Eq-Base} \qquad \frac{\begin{array}{c} G \vdash_B e :: () \\ G \vdash_B e_1 :: \lfloor \tau_x \to \tau \rfloor \qquad G \vdash_B e_2 :: \lfloor \tau_x \to \tau \rfloor \end{array}}{G \vdash_B \mathsf{xEq}_{x:\tau_x \to \tau}\ e_1\ e_2\ e :: \mathsf{PBEq}_{\lfloor \tau_x \to \tau \rfloor}}\ \textsc{BT-Eq-Fun}$$

Fig. 2. Syntax and Typing of λ^E.

B PROOFS AND DEFINITIONS FOR METATHEORY

In this section we provide proofs and definitions ommitted from § 3.

B.1 Base Type Checking

For completeness, we defined λ^E, the unrefined version of λ^{RE}, that ignores the refinements on basic types and the expression indexes from the typed equality.

The function $\lfloor \cdot \rfloor$ is defined to turn λ^{RE} types to their unrefined counterparts.

$$\begin{array}{rcl} \lfloor \mathsf{Bool} \rfloor & \doteq & \mathsf{Bool} \\ \lfloor () \rfloor & \doteq & () \\ \lfloor \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\} \rfloor & \doteq & \mathsf{PBEq}_{\lfloor \tau \rfloor} \\ \lfloor \{v{:}b \mid r\} \rfloor & \doteq & b \\ \lfloor x{:}\tau_x \to \tau \rfloor & \doteq & \lfloor \tau_x \rfloor \to \lfloor \tau \rfloor \end{array}$$

Figure 2 defines the syntax and typing of λ^E that we use to define type denotations of λ^{RE}.

B.2 Constant Property

Theorem B.1. *For the constants* $c = \mathtt{true}, \mathtt{false}, \mathtt{unit}$, *and* $==_b$, *constants are sound, i.e.,* $c \in \llbracket \mathsf{TyCons}(c) \rrbracket$.

Proof. Below are the proofs for each of the four constants.

- $e \equiv \mathtt{true}$ and $e \in \llbracket \{x{:}\mathsf{Bool} \mid x ==_{\mathsf{Bool}} \mathtt{true}\} \rrbracket$. We need to prove the below three requirements of membership in the interpretation of basic types:
  - $e \hookrightarrow^* v$, which holds because $\mathtt{true}$ is a value, thus $v = \mathtt{true}$;
  - $\vdash_B e :: \mathtt{Bool}$, which holds by the typing rule BT-Con; and
  - $(x ==_{\mathsf{Bool}} \mathtt{true})[e/x] \hookrightarrow^* \mathtt{true}$, which holds because

$$\begin{array}{rcl} (x ==_{\mathsf{Bool}} \mathtt{true})[e/x] & = & \mathtt{true} ==_{\mathsf{Bool}} \mathtt{true} \\ & \hookrightarrow & (==_{(\mathtt{true},\mathtt{Bool})})\ \mathtt{true} \\ & \hookrightarrow & \mathtt{true} = \mathtt{true} \\ & = & \mathtt{true} \end{array}$$

- $e \equiv \texttt{false}$ and $e \in [\![\, \{x{:}\texttt{Bool} \mid x ==_{\texttt{Bool}} \texttt{false}\} \,]\!]$. We need to prove the below three requirements of membership in the interpretation of basic types:
  - $e \hookrightarrow^* v$, which holds because $\texttt{false}$ is a value, thus $v = \texttt{false}$;
  - $\vdash_B e :: \texttt{Bool}$, which holds by the typing rule BT-Con; and
  - $(x ==_{\texttt{Bool}} \texttt{false})[e/x] \hookrightarrow^* \texttt{true}$, which holds because

$$\begin{array}{rcl} (x ==_{\texttt{Bool}} \texttt{false})[e/x] & = & \texttt{false} ==_{\texttt{Bool}} \texttt{false} \\ & \hookrightarrow & (==_{(\texttt{false},\texttt{Bool})})\ \texttt{false} \\ & \hookrightarrow & \texttt{false} = \texttt{false} \\ & = & \texttt{true} \end{array}$$

- $e \equiv \texttt{unit}$ and $e \in \left[\!\left|\, \{x{:}() \mid x ==_{()} \texttt{unit}\} \,\right|\!\right]$. We need to prove the below three requirements of membership in the interpretation of basic types:
  - $e \hookrightarrow^* v$, which holds because $\texttt{unit}$ is a value, thus $v = \texttt{unit}$;
  - $\vdash_B e :: ()$, which holds by the typing rule BT-Con; and
  - $(x ==_{()} \texttt{unit})[e/x] \hookrightarrow^* \texttt{true}$, which holds because

$$\begin{array}{rcl} (x ==_{()} \texttt{unit})[e/x] & = & \texttt{unit} ==_{()} \texttt{unit} \\ & \hookrightarrow & (==_{(\texttt{unit},\,())})\ \texttt{unit} \\ & \hookrightarrow & \texttt{unit} = \texttt{unit} \\ & = & \texttt{true} \end{array}$$

- $==_b \in\ [\![\, x{:}b \to y{:}b \to \{z{:}\texttt{Bool} \mid z ==_{\texttt{Bool}} (x ==_b y)\} \,]\!]$. By the definition of interpretation of function types, we fix $e_x, e_y \in [\![\, b \,]\!]$ and we need to prove that $e \equiv e_x ==_b e_y \in \left[\!\left|\, (\{z{:}\texttt{Bool} \mid z ==_{\texttt{Bool}} (x ==_b y)\})[e_x/x][e_y/y] \,\right|\!\right]$. We prove the below three requirements of membership in the interpretation of basic types:
  - $e \hookrightarrow^* v$, which holds because

$$\begin{array}{rcll} e & = & e_x ==_b e_y & \\ & \hookrightarrow^* & v_x ==_b e_y & \text{because } e_x \in [\![\, b \,]\!] \\ & \hookrightarrow^* & v_x ==_b v_y & \text{because } e_y \in [\![\, b \,]\!] \\ & \hookrightarrow & (==_{(v_x,b)})\ v_y & \\ & \hookrightarrow & v_x = v_y & \\ & = & v & \text{with } v = \texttt{true} \text{ or } v = \texttt{false} \end{array}$$

  - $\vdash_B e :: \texttt{Bool}$, which holds by the typing rule BT-Con and because $e_x, e_y \in [\![\, b \,]\!]$ thus $\vdash_B e_x :: b$ and $\vdash_B e_y :: b$; and

- $(z ==_{\texttt{Bool}} (x ==_b y))[e/z][e_x/x][e_y/y] \hookrightarrow^* \texttt{true}$. Since $e_x, e_y \in \llbracket b \rrbracket$ both expressions evaluate to values, say $e_x \hookrightarrow^* v_x$ and $e_y \hookrightarrow^* v_y$ which holds because

$$
\begin{array}{rcll}
(z ==_{\texttt{Bool}} (x ==_b y))[e/z][e_x/x][e_y/y] & = & e ==_{\texttt{Bool}} (e_x ==_b e_y) & \\
& = & (e_x ==_b e_y) ==_{\texttt{Bool}} (e_x ==_b e_y) & \\
& \hookrightarrow^* & (v_x ==_b e_y) ==_{\texttt{Bool}} (e_x ==_b e_y) & \text{since } e_x \hookrightarrow^* v_x \\
& \hookrightarrow^* & (v_x ==_b v_y) ==_{\texttt{Bool}} (e_x ==_b e_y) & \text{since } e_y \hookrightarrow^* v_y \\
& \hookrightarrow & ((==_{(v_x, b)})\ v_y) ==_{\texttt{Bool}} (e_x ==_b e_y) & \\
& \hookrightarrow & (v_x = v_y) ==_{\texttt{Bool}} (e_x ==_b e_y) & \\
& \hookrightarrow^* & (v_x = v_y) ==_{\texttt{Bool}} (v_x ==_b e_y) & \text{since } e_x \hookrightarrow^* v_x \\
& \hookrightarrow^* & (v_x = v_y) ==_{\texttt{Bool}} (v_x ==_b v_y) & \text{since } e_y \hookrightarrow^* v_y \\
& \hookrightarrow & (v_x = v_y) ==_{\texttt{Bool}} ((==_{(v_x, b)})\ v_y) & \\
& \hookrightarrow & (v_x = v_y) ==_{\texttt{Bool}} (v_x = v_y) & \\
& \hookrightarrow & (v_x = v_y) ==_{\texttt{Bool}} (v_x = v_y) & \\
& \hookrightarrow & ((==_{((v_x = v_y), \texttt{Bool})})\ (v_x = v_y) & \\
& \hookrightarrow & (v_x = v_y) = (v_x = v_y) & \\
& = & \texttt{true} &
\end{array}
$$

□

B.3 Type Soundness

Theorem B.2 (Semantic soundness). *If* $\Gamma \vdash e :: \tau$ *then* $\Gamma \models e \in \tau$.

Proof. By induction on the typing derivation.

T-Sub By inversion of the rule we have
(1) $\Gamma \vdash e :: \tau'$
(2) $\Gamma \vdash \tau' \preceq \tau$
By IH on (1) we have
(3) $\Gamma \models e \in \tau'$
By Theorem B.6 and (2) we have
(4) $\Gamma \vdash \tau' \subseteq \tau$
By (3), (4), and the definition of subsets we directly get $\Gamma \models e \in \tau$.

T-Self Assume $\Gamma \vdash e :: \{z{:}b \mid z ==_b e\}$. By inversion we have
(1) $\Gamma \vdash e :: \{z{:}b \mid r\}$
By IH we have
(2) $\Gamma \models e \in \{z{:}b \mid r\}$
We fix $\theta \in \llbracket \Gamma \rrbracket$. By the definition of semantic typing we get
(3) $\theta \cdot e \in \llbracket \theta \cdot \{z{:}b \mid r\} \rrbracket$
By the definition of denotations on basic types we have
(4) $\theta \cdot e \hookrightarrow^* v$
(5) $\vdash_B \theta \cdot e :: b$
(6) $\theta \cdot r[\theta \cdot e/z] \hookrightarrow^* \texttt{true}$
Since θ contains values, by the definition of $==_b$ we have
(7) $\theta \cdot e ==_b \theta \cdot e \hookrightarrow^* \texttt{true}$
Thus
(8) $\theta \cdot (z ==_b e)[\theta \cdot e/z] \hookrightarrow^* \texttt{true}$
By (4), (5), and (8) we have
(9) $\theta \cdot e \in \llbracket \theta \cdot \{z{:}b \mid z ==_b e\} \rrbracket$
Thus, $\Gamma \models e \in \{z{:}b \mid z ==_b e\}$.

T-Con This case holds exactly because of Property B.1.

T-Var This case holds by the definition of closing substitutions.

T-Lam Assume $\Gamma \vdash \lambda x{:}\tau_x.\ e :: x{:}\tau_x \rightarrow \tau$. By inversion of the rule we have $\Gamma, x : \tau_x \vdash e :: \tau$. By IH we get $\Gamma, x : \tau_x \models e \in \tau$.

We need to show that $\Gamma \models \lambda x{:}\tau_x.\ e \in x{:}\tau_x \rightarrow \tau$. Which, for some $\theta \in \llbracket \Gamma \rrbracket$ is equivalent to $\lambda x{:}\theta \cdot \tau_x.\ \theta \cdot e \in \llbracket x{:}\theta \cdot \tau_x \rightarrow \theta \cdot \tau \rrbracket$.

We pick a random $e_x \in \llbracket \theta \cdot \tau_x \rrbracket$ thus we need to show that $\theta \cdot e[e_x/x] \in \llbracket \theta \cdot \tau[e_x/x] \rrbracket$. By Lemma B.3, there exists v_x so that $e_x \hookrightarrow^* v_x$ and $v_x \in \llbracket \tau_x \rrbracket$. By the inductive hypothesis, $\theta \cdot e[v_x/x] \in \llbracket \theta \cdot \tau[v_x/x] \rrbracket$. By Lemma B.4, $\theta \cdot e[e_x/x] \in \llbracket \theta \cdot \tau[e_x/x] \rrbracket$, which concludes our proof.

T-App Assume $\Gamma \vdash e\ e_x :: \tau[e_x/x]$. By inversion we have

(1) $\Gamma \vdash e :: x{:}\tau_x \rightarrow \tau$

(2) $\Gamma \vdash e_x :: \tau_x$

By IH we get

(3) $\Gamma \models e \in x{:}\tau_x \rightarrow \tau$

(4) $\Gamma \models e_x \in \tau_x$

We fix $\theta \in \llbracket \Gamma \rrbracket$. By the definition of semantic types

(5) $\theta \cdot e \in \llbracket \theta \cdot x{:}\tau_x \rightarrow \tau \rrbracket$

(6) $\theta \cdot e_x \in \llbracket \theta \cdot \tau_x \rrbracket$

By (5), (6), and the definition of semantic typing on functions:

(7) $\theta \cdot e\ e_x \in \llbracket \theta \cdot \tau[e_x/x] \rrbracket$

Which directly leads to the required $\Gamma \models e\ e_x \in \tau[e_x/x]$

T-Eq-Base Assume $\Gamma \vdash \mathsf{bEq}_b\ e_l\ e_r\ e :: \mathsf{PEq}_b\ \{e_l\}\ \{e_r\}$. By inversion we get:

(1) $\Gamma \vdash e_l :: \tau_l$

(2) $\Gamma \vdash e_r :: \tau_r$

(3) $\Gamma \vdash \tau_l \preceq \{x{:}b \mid \mathtt{true}\}$

(4) $\Gamma \vdash \tau_r \preceq \{x{:}b \mid \mathtt{true}\}$

(5) $\Gamma, r : \tau_r, l : \tau_l \vdash e :: \{x{:}() \mid l ==_b r\}$

By IH we get

(4) $\Gamma \models e_l \in \tau_l$

(5) $\Gamma \models e_r \in \tau_r$

(6) $\Gamma, r : \tau_r, l : \tau_l \models e \in \{x{:}() \mid l ==_b r\}$

We fix $\theta \in \llbracket \Gamma \rrbracket$. Then (4) and (5) become

(7) $\theta \cdot e_l \in \llbracket \theta \cdot \tau_l \rrbracket$

(8) $\theta \cdot e_r \in \llbracket \theta \cdot \tau_r \rrbracket$

(9) $\Gamma \models e_r \in \tau_r$

(10) $\Gamma, r : \tau_r, l : \tau_l \models e \in \{x{:}() \mid l ==_b r\}$

Assume

(11) $\theta \cdot e_l \hookrightarrow^* v_l$

(12) $\theta \cdot e_r \hookrightarrow^* v_r$

By (7), (8), (11), (12), and Lemma B.3 we get

(13) $v_l \in \llbracket \theta \cdot \tau_l \rrbracket$

(14) $v_r \in \llbracket \theta \cdot \tau_r \rrbracket$

By (10), (11), and (12) we get

(15) $v_l ==_b v_r \hookrightarrow^* \mathtt{true}$

By (11), (12), (15), ane Lemma B.5 we have

(16) $\theta \cdot e_l ==_b \theta \cdot e_r \hookrightarrow^* \mathtt{true}$

By (1-5) we get:

(17) $\vdash_B \theta \cdot \mathsf{bEq}_b\ e_l\ e_r\ e :: \mathsf{PBEq}_b$
Trivially, with zero evaluation steps we have:
(18) $\theta \cdot \mathsf{bEq}_b\ e_l\ e_r\ e \hookrightarrow^* \mathsf{bEq}_b\ (\theta \cdot e_l)\ (\theta \cdot e_l)\ (\theta \cdot e)$
By (16), (17), (18) and the definition of semantic types on basic equality types we have
(19) $\theta \cdot \mathsf{bEq}_b\ e_l\ e_r\ e \in \llbracket \theta \cdot \mathsf{PEq}_b\ \{e_l\}\ \{e_r\} \rrbracket$
Which leads to the required $\Gamma \models \mathsf{bEq}_b\ e_l\ e_r\ e \in \mathsf{PEq}_b\ \{e_l\}\ \{e_r\}$.

T-Eq-Fun Assume $\Gamma \vdash \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e :: \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\}$. By inversion we have
(1) $\Gamma \vdash e_l :: \tau_l$
(2) $\Gamma \vdash e_r :: \tau_r$
(3) $\Gamma \vdash \tau_l\ \preceq\ x{:}\tau_x \to \tau$
(4) $\Gamma \vdash \tau_r\ \preceq\ x{:}\tau_x \to \tau$
(5) $\Gamma, r : \tau_r, l : \tau_l \vdash e :: (x{:}\tau_x \to \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\})$
(6) $\Gamma \vdash x{:}\tau_x \to \tau$
By IH and Theorem B.6 we get
(7) $\Gamma \models e_l \in \tau_l$
(8) $\Gamma \models e_r \in \tau_r$
(9) $\Gamma \vdash \tau_l\ \subseteq\ x{:}\tau_x \to \tau$
(10) $\Gamma \vdash \tau_r\ \subseteq\ x{:}\tau_x \to \tau$
(11) $\Gamma, r : \tau_r, l : \tau_l \models e \in (x{:}\tau_x \to \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\})$
By (1-5) we get
(12) $\vdash_B \theta \cdot \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e :: \mathsf{PBEq}_{\lfloor \theta \cdot (x:\tau_x \to \tau) \rfloor}$
Trivially, by zero evaluation steps, we get
(13) $\theta \cdot \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \hookrightarrow^* \mathsf{xEq}_{x:\theta \cdot \tau_x \to \theta \cdot \tau}\ (\theta \cdot e_l)\ (\theta \cdot e_r)\ (\theta \cdot e)$
By (7-10) we get
(14) $\theta \cdot e_l, \theta \cdot e_r \in \llbracket \theta \cdot x{:}\tau_x \to \tau \rrbracket$
By (7), (8), (11), the definition of semantic types on functions, and Lemmata B.3 and B.4 (similar to the previous case) we have
– $\forall e_x \in \llbracket \tau_x \rrbracket . e\ e_x \in \left\llbracket \mathsf{PEq}_{\tau[e_x/x]}\ \{e_l\ e_x\}\ \{e_r\ e_x\} \right\rrbracket$
By (12), (13), (14), and (15) we get
(19) $\theta \cdot \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \in \left\llbracket \theta \cdot \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\} \right\rrbracket$
Which leads to the required $\Gamma \models \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \in \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\}$.

□

Lemma B.3. *If $e \in \llbracket \tau \rrbracket$, then $e \hookrightarrow^* v$ and $v \in \llbracket \tau \rrbracket$.*

Proof. By structural induction of the type τ. □

Lemma B.4. *If $e_x \hookrightarrow^* v_x$ and $e[v_x/x] \in \llbracket \tau[v_x/x] \rrbracket$, then $e[e_x/x] \in \llbracket \tau[e_x/x] \rrbracket$.*

Proof. We can use parallel reductions (of §C) to prove that if $e_1 \rightrightarrows e_2$, then (1) $\llbracket \tau[e_1/x] \rrbracket = \llbracket \tau[e_2/x] \rrbracket$ and (2) $e_1 \in \llbracket \tau \rrbracket$ *iff* $e_2 \in \llbracket \tau \rrbracket$. The proof directly follows by these two properties. □

Lemma B.5. *If $e_x \hookrightarrow^* e'_x$ and $e[e'_x/x] \hookrightarrow^* c$, then $e[e_x/x] \hookrightarrow^* c$.*

Proof. As an instance of Corollary C.17. □

We define semantic subtyping as follows: $\Gamma \vdash \tau\ \subseteq\ \tau'$ iff $\forall \theta \in \llbracket \Gamma \rrbracket . \llbracket \theta \cdot \tau \rrbracket \subseteq \llbracket \theta \cdot \tau' \rrbracket$.

Theorem B.6 (Subtyping semantic soundness). *If $\Gamma \vdash \tau\ \preceq\ \tau'$ then $\Gamma \vdash \tau\ \subseteq\ \tau'$.*

Proof. By induction on the derivation tree:

S-Base Assume $\Gamma \vdash \{x{:}b \mid r\} \;\preceq\; \{x'{:}b \mid r'\}$. By inversion $\forall \theta \in \llbracket \Gamma \rrbracket,\ \llbracket \theta \cdot \{x{:}b \mid r\} \rrbracket \subseteq \llbracket \theta \cdot \{x'{:}b \mid r'\} \rrbracket$, which exactly leads to the required.

S-Fun Assume $\Gamma \vdash x{:}\tau_x \to \tau \;\preceq\; x{:}\tau'_x \to \tau'$. By inversion

(1) $\Gamma \vdash \tau'_x \;\preceq\; \tau_x$

(2) $\Gamma, x : \tau'_x \vdash \tau \;\preceq\; \tau'$

By IH

(3) $\Gamma \vdash \tau'_x \;\subseteq\; \tau_x$

(4) $\Gamma, x : \tau'_x \vdash \tau \;\subseteq\; \tau'$

We fix $\theta \in \Gamma$. We pick e. We assume $e \in \llbracket \theta \cdot x{:}\tau_x \to \tau \rrbracket$ and we will show that $e \in \llbracket \theta \cdot x{:}\tau'_x \to \tau' \rrbracket$. By assumption

(5) $\forall e_x \in \llbracket \theta \cdot \tau_x \rrbracket.\ e\ e_x \in \llbracket \theta \cdot \tau[e_x/x] \rrbracket$

We need to show $\forall e_x \in \llbracket \theta \cdot \tau'_x \rrbracket.\ e\ e_x \in \llbracket \theta \cdot \tau'[e_x/x] \rrbracket$. We fix e_x. By (3), if $e_x \in \llbracket \theta \cdot \tau'_x \rrbracket$, then $e_x \in \llbracket \theta \cdot \tau_x \rrbracket$ and (5) applies, so $e\ e_x \in \llbracket \theta \cdot \tau[e_x/x] \rrbracket$, which by (4) gives $e\ e_x \in \llbracket \theta \cdot \tau'[e_x/x] \rrbracket$. Thus, $e \in \llbracket \theta \cdot x{:}\tau'_x \to \tau' \rrbracket$. This leads to $\llbracket \theta \cdot x{:}\tau_x \to \tau \rrbracket \subseteq \llbracket \theta \cdot x{:}\tau'_x \to \tau' \rrbracket$, which by definition gives semantic subtyping: $\Gamma \vdash x{:}\tau_x \to \tau \;\subseteq\; x{:}\tau'_x \to \tau'$.

S-Eq Assume $\Gamma \vdash \mathsf{PEq}_{\tau_i}\ \{e_l\}\ \{e_r\} \;\preceq\; \mathsf{PEq}_{\tau'_i}\ \{e_l\}\ \{e_r\}$. We split cases on the structure of τ_i.

- *If* τ_i is a basic type, then τ_i is trivially refined to true. Thus, $\tau_i = \tau'_i = b$ and for each $\theta \in \Gamma$, $\llbracket \theta \cdot \mathsf{PEq}_{\tau}\ \{e_l\}\ \{e_r\} \rrbracket = \llbracket \theta \cdot \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\} \rrbracket$, thus set inclusion reduces to equal sets.
- *If* τ_i is a function type, thus $\Gamma \vdash \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\} \;\preceq\; \mathsf{PEq}_{x:\tau'_x \to \tau'}\ \{e_l\}\ \{e_r\}$

By inversion

(1) $\Gamma \vdash x{:}\tau_x \to \tau \;\preceq\; x{:}\tau'_x \to \tau'$

(2) $\Gamma \vdash x{:}\tau'_x \to \tau' \;\preceq\; x{:}\tau_x \to \tau$

By inversion on (1) and (2) we get

(3) $\Gamma \vdash \tau'_x \;\preceq\; \tau_x$

(4) $\Gamma, x : \tau'_x \vdash \tau \;\preceq\; \tau'$

(5) $\Gamma, x : \tau_x \vdash \tau' \;\preceq\; \tau$

By IH on (1) and (3) we get

(6) $\Gamma \vdash x{:}\tau_x \to \tau \;\subseteq\; x{:}\tau'_x \to \tau'$

(7) $\Gamma \vdash \tau'_x \;\subseteq\; \tau_x$

We fix $\theta \in \Gamma$ and some e. If $e \in \llbracket \theta \cdot \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\} \rrbracket$ we need to show that $e \in \llbracket \theta \cdot \mathsf{PEq}_{x:\tau'_x \to \tau'}\ \{e_l\}\ \{e_r\} \rrbracket$. By the assumption we have

(8) $\vdash_B e :: \mathsf{PBEq}_{\lfloor \theta \cdot (x:\tau_x \to \tau) \rfloor}$

(9) $e \hookrightarrow^* \mathsf{xEq\_}\ (\theta \cdot e_l)\ (\theta \cdot e_r)\ e_{pf}$

(10) $(\theta \cdot e_l), (\theta \cdot e_r) \in \llbracket \theta \cdot (x{:}\tau_x \to \tau) \rrbracket$

(11) $\forall e_x \in \llbracket \theta \cdot \tau_x \rrbracket .e_{pf}\ e_x \in \llbracket \mathsf{PEq}_{\theta \cdot (\tau[e_x/x])}\ \{(\theta \cdot e_l)\ e_x\}\ \{(\theta \cdot e_r)\ e_x\} \rrbracket$

Since (8) only depends on the structure of the type index, we get

(12) $\vdash_B e :: \mathsf{PBEq}_{\lfloor \theta \cdot (x:\tau'_x \to \tau') \rfloor}$

By (6) and (10) we get

(13) $(\theta \cdot e_l), (\theta \cdot e_r) \in \llbracket \theta \cdot (x{:}\tau'_x \to \tau') \rrbracket$

By (4), (5), Lemma B.7, the rule S-Eq and the IH, we get that $\llbracket \mathsf{PEq}_{\theta \cdot (\tau[e_x/x])}\ \{(\theta \cdot e_l)\ e_x\}\ \{(\theta \cdot e_r)\ e_x\} \rrbracket \subseteq \llbracket \mathsf{PEq}_{\theta \cdot (\tau'[e_x/x])}\ \{(\theta \cdot e_l)\ e_x\}\ \{(\theta \cdot e_r)\ e_x\} \rrbracket$. By which, (11), (7), and reasoning similar to the S-Fun case, we get

(14) $\forall e_x \in \llbracket \theta \cdot \tau'_x \rrbracket .e_{pf}\ e_x \in \llbracket \mathsf{PEq}_{\theta \cdot (\tau'[e_x/x])}\ \{(\theta \cdot e_l)\ e_x\}\ \{(\theta \cdot e_r)\ e_x\} \rrbracket$

By (12), (9), (13), and (14) we conclude that $e \in \llbracket \theta \cdot \mathsf{PEq}_{x:\tau'_x \to \tau'}\ \{e_l\}\ \{e_r\} \rrbracket$, thus $\Gamma \vdash \mathsf{PEq}_{x:\tau_x \to \tau}\ \{e_l\}\ \{e_r\} \subseteq \mathsf{PEq}_{x:\tau'_x \to \tau'}\ \{e_l\}\ \{e_r\}$.

□

Lemma B.7 (Strengthening). *If* $\Gamma_1 \vdash \tau_1 \preceq \tau_2$, *then:*

(1) If $\Gamma_1, x : \tau_2, \Gamma_2 \vdash e :: \tau$ *then* $\Gamma_1, x : \tau_1, \Gamma_2 \vdash e :: \tau$.
(2) If $\Gamma_1, x : \tau_2, \Gamma_2 \vdash \tau \preceq \tau'$ *then* $\Gamma_1, x : \tau_1, \Gamma_2 \vdash \tau \preceq \tau'$.
(3) If $\Gamma_1, x : \tau_2, \Gamma_2 \vdash \tau$ *then* $\Gamma_1, x : \tau_1, \Gamma_2 \vdash \tau$.
(4) If $\vdash \Gamma_1, x : \tau_2, \Gamma_2$ *then* $\vdash \Gamma_1, x : \tau_1, \Gamma_2$.

Proof. The proofs go by induction. Only the T-Var case is insteresting; we use T-Sub and our assumption. □

Lemma B.8 (Semantic typing is closed under parallel reduction in expressions). *If* $e_1 \rightrightarrows^* e_2$, *then* $e_1 \in [\![\tau]\!]$ *iff* $e_2 \in [\![\tau]\!]$.

Proof. By induction on τ, using parallel reduction as a bisimulation (Lemma C.5 and Corollary C.15). □

Lemma B.9 (Semantic typing is closed under parallel reduction in types). *If* $\tau_1 \rightrightarrows^* \tau_2$ *then* $[\![\tau_1]\!] = [\![\tau_2]\!]$.

Proof. By induction on τ_1 (which necessarily has the same shape as τ_2). We use parallel reduction as a bisimulation (Lemma C.5 and Corollary C.15). □

Lemma B.10 (Parallel reducing types are equal). *If* $\Gamma \vdash \tau_1$ *and* $\Gamma \vdash \tau_2$ *and* $\tau_1 \rightrightarrows^* \tau_2$ *then* $\Gamma \vdash \tau_1 \preceq \tau_2$ *and* $\Gamma \vdash \tau_1 \preceq \tau_2$.

Proof. By induction on the parallel reduction sequence; for a single step, by induction on τ_1 (which must have the same structure as τ_2). We use parallel reduction as a bisimulation (Lemma C.5 and Corollary C.15). □

Lemma B.11 (Regularity). *(1) If* $\Gamma \vdash e :: \tau$ *then* $\vdash \Gamma$ *and* $\Gamma \vdash \tau$.
(2) If $\Gamma \vdash \tau$ *then* $\vdash \Gamma$.
(3) If $\Gamma \vdash \tau_1 \preceq \tau_2$ *then* $\vdash \Gamma$ *and* $\Gamma \vdash \tau_1$ *and* $\Gamma \vdash \tau_2$.

Proof. By a big ol' induction. □

Lemma B.12 (Canonical forms). *If* $\Gamma \vdash v :: \tau$, *then:*

- *If* $\tau = \{x{:}b \mid e\}$, *then* $v = c$ *such that* $\mathsf{TyCons}(c) = b$ *and* $\Gamma \vdash \mathsf{TyCons}(c) \preceq \{x{:}b \mid e\}$.
- *If* $\tau = x{:}\tau_x \rightarrow \tau'$, *then* $v = \textit{T-Lam} x \tau'_x e$ *such that* $\Gamma \vdash \tau_x \preceq \tau'_x$ *and* $\Gamma, x : \tau'_x \vdash e :: \tau''$ *such that* $\tau'' \vdash \tau' \preceq$.
- *If* $\tau = \mathsf{PEq}_b\ \{e_l\}\ \{e_r\}$ *then* $v = \mathsf{bEq}_b\ e_l\ e_r\ v_p$ *such that* $\Gamma \vdash e_l :: \tau_l$ *and* $\Gamma \vdash e_r :: \tau_r$ *(for some* τ_l *and* τ_r *that are refinements of* b*) and* $\Gamma, r : \tau_r, l : \tau_l \vdash v_p :: \{x{:}() \mid l ==_b r\}$.
- *If* $\tau = \mathsf{PEq}_{x:\tau_x \rightarrow \tau'}\ \{e_l\}\ \{e_r\}$ *then* $v = \mathsf{xEq}_{x:\tau'_x \rightarrow \tau''}\ e_l\ e_r\ v_p$ *such that* $\Gamma \vdash \tau_x \preceq \tau'_x$ *and* $\Gamma, x : \tau_x \vdash \tau'' \preceq \tau'$ *and* $\Gamma \vdash e_l :: \tau_l$ *and* $\Gamma \vdash e_r :: \tau_r$ *(for some* τ_l *and* τ_r *that are subtypes of* $x{:}\tau'_x \rightarrow \tau''$*) and* $\Gamma, r : \tau_r, l : \tau_l \vdash v_p :: x{:}\tau'_x \rightarrow \mathsf{PEq}_{\tau''}\ \{e_l\ x\}\ \{e_r\ x\}$.

B.4 The Binary Logical Relation

Theorem B.13 (EqRT soundness). *If* $\Gamma \vdash e :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$, *then* $\Gamma \vdash e_1 \sim e_2 :: \tau$.

Proof. By $\Gamma \vdash e :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$ and the Fundamental Property B.22 we have $\Gamma \vdash e \sim e :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$. Thus, for a fixed $\delta \in \Gamma$, $\delta_1 \cdot e \sim \delta_2 \cdot e :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}; \delta$. By the definition of the logical relation for EqRT, we have $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \tau; \delta$. So, $\Gamma \vdash e_1 \sim e_2 :: \tau$. □

Lemma B.14 (LR respects subtyping). *If* $\Gamma \vdash e_1 \sim e_2 :: \tau$ *and* $\Gamma \vdash \tau \preceq \tau'$, *then* $\Gamma \vdash e_1 \sim e_2 :: \tau'$.

Proof. By induction on the derivation of the subtyping tree.

S-Base By assumption we have

(1) $\Gamma \vdash e_1 \sim e_2 :: \{x{:}b \mid r\}$

(2) $\Gamma \vdash \{x{:}b \mid r\} \;\preceq\; \{x'{:}b \mid r'\}$

By inversion on (2) we get

(3) $\forall \theta \in \llbracket \Gamma \rrbracket,\; \llbracket \theta \cdot \{x{:}b \mid r\} \rrbracket \subseteq \llbracket \theta \cdot \{x'{:}b \mid r'\} \rrbracket$

We fix $\delta \in \Gamma$. By (1) we get

(4) $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \{x{:}b \mid r\};\, \delta$

By the definition of logical relations:

(5) $\delta_1 \cdot e_1 \hookrightarrow^* v_1$

(6) $\delta_2 \cdot e_2 \hookrightarrow^* v_2$

(7) $v_1 \sim v_2 :: \{x{:}b \mid r\};\, \delta$

By (7) and the definition of the logical relation on basic types we have

(8) $v_1 = v_2 = c$

(9) $\vdash_B c :: b$

(10) $\delta_1 \cdot r[c/x] \hookrightarrow^* \texttt{true}$

(11) $\delta_2 \cdot r[c/x] \hookrightarrow^* \texttt{true}$

By (3), (10) and (11) become

(12) $\delta_1 \cdot r'[c/x'] \hookrightarrow^* \texttt{true}$

(13) $\delta_2 \cdot r'[c/x'] \hookrightarrow^* \texttt{true}$

By (8), (9), (12), and (13) we get

(14) $v_1 \sim v_2 :: \{x'{:}b \mid r'\};\, \delta$

By (5), (6), and (14) we have

(15) $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \{x'{:}b \mid r'\};\, \delta$

Thus, $\Gamma \vdash e_1 \sim e_2 :: \{x'{:}b \mid r'\}$.

S-Fun By assumption:

(1) $\Gamma \vdash e_1 \sim e_2 :: x{:}\tau_x \to \tau$

(2) $\Gamma \vdash x{:}\tau_x \to \tau \;\preceq\; x{:}\tau'_x \to \tau'$

By inversion of the rule (2)

(3) $\Gamma \vdash \tau'_x \;\preceq\; \tau_x$

(4) $\Gamma, x : \tau'_x \vdash \tau \;\preceq\; \tau'$

We fix $\delta \in \Gamma$. By (1) and the definition of logical relation

(5) $\delta_1 \cdot e_1 \hookrightarrow^* v_1$

(6) $\delta_2 \cdot e_2 \hookrightarrow^* v_2$

(7) $v_1 \sim v_2 :: x{:}\tau_x \to \tau;\, \delta$

We fix v'_1 and v'_2 so that

(8) $v'_1 \sim v'_2 :: \tau'_x;\, \delta$

By (8) and the definition of logical relations, since the values are idempotent under substitution, we have

(9) $\Gamma \vdash v'_1 \sim v'_2 :: \tau'_x$

By (9) and inductive hypothesis on (3) we have

(10) $\Gamma \vdash v'_1 \sim v'_2 :: \tau_x$

By (10), idempotence of values under substitution, and the definition of logical relations, we have

(11) $v'_1 \sim v'_2 :: \tau_x;\, \delta$

By (7), (11), and the definition of logical relations on function values:

(12) $v_1\ v'_1 \sim v_2\ v'_2 :: \tau;\, \delta, (v'_1, v'_2)/x$

By (9), (12), and the definition of logical relations we have

(12) $\Gamma, x : \tau'_x \vdash v_1\ v'_1 \sim v_2\ v'_2 :: \tau$

By (12) and inductive hypothesis on (4) we have

(13) $\Gamma, x : \tau'_x \vdash v_1\ v'_1 \sim v_2\ v'_2 :: \tau'$

By (8), (13), and the definition of logical relations, we have

(14) $v_1\ v'_1 \sim v_2\ v'_2 :: \tau';\ \delta, (v'_1, v'_2)/x$

By (8), (14), and the definition of logical relations, we have

(15) $v_1 \sim v_2 :: x{:}\tau'_x \rightarrow \tau';\ \delta$

By (5), (6), and (15), we get

(16) $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: x{:}\tau'_x \rightarrow \tau';\ \delta$

So, $\Gamma \vdash e_1 \sim e_2 :: x{:}\tau'_x \rightarrow \tau'$.

S-Eq By hypothesis:

(1) $\Gamma \vdash e_1 \sim e_2 :: \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\}$

(2) $\Gamma \vdash \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\}\ \preceq\ \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$

We fix $\delta \in \Gamma$. By (1)

(3) $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\};\ \delta$

By (3) and the definition of logical relations.

(4) $\delta_1 \cdot e_1 \hookrightarrow^* v_1$

(5) $\delta_2 \cdot e_2 \hookrightarrow^* v_2$

(6) $v_1 \sim v_2 :: \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\};\ \delta$

By (6) and the definition of logical relations

(7) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: \tau;\ \delta$

By (7) and the definition of logical relations.

(8) $\Gamma \vdash e_l \sim e_r :: \tau$

By inversion on (2)

(9) $\Gamma \vdash \tau\ \preceq\ \tau'$

(10) $\Gamma \vdash \tau'\ \preceq\ \tau$

By (8) and inductive hypothesis on (9)

(11) $\Gamma \vdash e_l \sim e_r :: \tau'$

Thus,

(12) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: \tau';\ \delta$

By (12), (4), (5), and determinism of operational semantics:

(12) $v_1 \sim v_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\};\ \delta$

By (4), (5), and (13)

(14) $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\};\ \delta$

So, by definition of logical relations, $\Gamma \vdash e_1 \sim e_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$.

□

Lemma B.15 (Constant soundness). $\Gamma \vdash c \sim c :: \mathsf{TyCons}(c)$

Proof. The proof follows the same steps as Theorem B.1. □

Lemma B.16 (Selfification of constants). *If* $\Gamma \vdash e \sim e :: \{z{:}b \mid r\}$ *then* $\Gamma \vdash x \sim x :: \{z{:}b \mid z ==_b x\}$.

Proof. We fix $\delta \in \Gamma$. By hypothesis $(v_1, v_2)/x \in \delta$ with $v_1 \sim v_2 :: \{z{:}b \mid r\};\ \delta$. We need to show that $\delta_1 \cdot x \sim \delta_2 \cdot x :: \{z{:}b \mid z ==_b x\};\ \delta$. Which reduces to $v_1 \sim v_2 :: \{z{:}b \mid z ==_b x\};\ \delta$. By the definition on the logical relation on basic values, we know $v_1 = v_2 = c$ and $\vdash_B c :: b$. Thus, we are left to prove that $\delta_1 \cdot ((z ==_b x)[c/z]) \hookrightarrow^* \mathtt{true}$ and $\delta_2 \cdot ((z ==_b x)[c/z]) \hookrightarrow^* \mathtt{true}$ which, both, trivially hold by the definition of $==_b$. □

Lemma B.17 (Variable soundness). *If* $x : \tau \in \Gamma$, *then* $\Gamma \vdash x \sim x :: \tau$.

Proof. By the definition of the logical relation it suffices to show that $\forall \delta \in \Gamma . \delta_1(x) \sim \delta_2(x) :: \tau;\ \delta$; which is trivially true by the definition of $\delta \in \Gamma$. □

Lemma B.18 (Transitivity of Evaluation). *If $e \hookrightarrow^* e'$, then $e \hookrightarrow^* v$ iff $e' \hookrightarrow^* v$.*

Proof. Assume $e \hookrightarrow^* v$. Since the $\hookrightarrow$ is by definition deterministic, there exists a unique sequence $e \hookrightarrow e_1 \hookrightarrow \ldots \hookrightarrow e_i \hookrightarrow \ldots \hookrightarrow v$. By assumption, $e \hookrightarrow^* e'$, so there exists a j, so $e' \equiv e_j$, and $e' \hookrightarrow^* v$ following the same sequence.

Assume $e' \hookrightarrow^* v$. Then $e \hookrightarrow^* e' \hookrightarrow^* v$ uniquely evaluates e to v. □

Lemma B.19 (LR closed under evaluation). *If $e_1 \hookrightarrow^* e_1'$, $e_2 \hookrightarrow^* e_2'$, then $e_1' \sim e_2' :: \tau;\ \delta$ iff $e_1 \sim e_2 :: \tau;\ \delta$.*

Proof. Assume $e_1' \sim e_2' :: \tau;\ \delta$, by the definition of the logical relation on closed terms we have $e_1' \hookrightarrow^* v_1, e_2' \hookrightarrow^* v_2$, and $v_1 \sim v_2 :: \tau;\ \delta$. By Lemma B.18 and by assumption, $e_1 \hookrightarrow^* e_1'$ and $e_2 \hookrightarrow^* e_2'$, we have $e_1 \hookrightarrow^* v_1$ and $e_2 \hookrightarrow^* v_2$. By which and $v_1 \sim v_2 :: \tau;\ \delta$ we get that $e_1 \sim e_2 :: \tau;\ \delta$. The other direction is identical. □

Lemma B.20 (LR closed under parallel reduction). *If $e_1 \rightrightarrows^* e_1'$, $e_2 \rightrightarrows^* e_2'$, and $e_1' \sim e_2' :: \tau;\ \delta$, then $e_1 \sim e_2 :: \tau;\ \delta$.*

Proof. By induction on τ, using parallel reduction as a backward simulation (Corollary C.15). □

Lemma B.21 (LR Compositionality). *If $\delta_1 \cdot e_x \hookrightarrow^* v_{x_1}, \delta_2 \cdot e_x \hookrightarrow^* v_{x_2}, e_1 \sim e_2 :: \tau;\ \delta, (v_{x_1}, v_{x_2})/x$, then $e_1 \sim e_2 :: \tau[e_x/x];\ \delta$.*

Proof. By the assumption we have that

(1) $\delta_1 \cdot e_x \hookrightarrow^* v_{x_1}$
(2) $\delta_2 \cdot e_x \hookrightarrow^* v_{x_2}$
(3) $e_1 \hookrightarrow^* v_1$
(4) $e_2 \hookrightarrow^* v_2$
(5) $v_1 \sim v_2 :: \tau;\ \delta, (v_{x1}, v_{x_2})/x$

and we need to prove that $v_1 \sim v_2 :: \tau[e_x/x];\ \delta$. The proof goes by structural induction on the type τ.

- $\tau \doteq \{z{:}b \mid r\}$. For $i = 1, 2$ we need to show that if $\delta_i, [v_{x_i}/x] \cdot r[v_i/z] \hookrightarrow^*$ `true` then $\delta_i \cdot r[v_i/z][e_i/x] \hookrightarrow^*$ `true`. We have $\delta_i, [v_{x_i}/x] \cdot r[v_i/z] \rightrightarrows^* \delta_i \cdot r[v_i/z][e_i/x]$ because substituting parallel reducing terms parallel reduces (Corollary C.3) and parallel reduction subsumes reduction (Lemma C.4). By cotermination at constants (Corollary C.17), we have $\delta_i \cdot r[v_i/z][e_i/x] \hookrightarrow^*$ `true`.
- $\tau \doteq y{:}\tau_y' \to \tau'$. We need to show that if $v_1 \sim v_2 :: y{:}\tau_y' \to \tau';\ \delta, (v_{x_1}, v_{x_2})/x$, then $v_1 \sim v_2 :: y{:}\tau_y' \to \tau'[e_x/x];\ \delta$.
  We fix v_{y_1} and v_{y_2} so that $v_{y_1} \sim v_{y_2} :: \tau_y';\ \delta, (v_{x_1}, v_{x_2})/x$.
  Then, we have that $v_1\ v_{y_1} \sim v_2\ v_{y_2} :: \tau';\ \delta, (v_{x_1}, v_{x_2})/x, (v_{y_1}, v_{y_2})/y$.
  By inductive hypothesis, we have that $v_1\ v_{y_1} \sim v_2\ v_{y_2} :: \tau'[e_x/x];\ \delta, (v_{y_1}, v_{y_2})/y$.
  By inductive hypothesis on the fixed arguments, we also get $v_{y_1} \sim v_{y_2} :: \tau_y'[e_x/x];\ \delta$.
  Combined, we get $v_1 \sim v_2 :: y{:}\tau_y' \to \tau'[e_x/x];\ \delta$.
- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$. We need to show that if $v_1 \sim v_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\};\ \delta, (v_{x_1}, v_{x_2})/x$, then $v_1 \sim v_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}[e_x/x];\ \delta$.
  This reduces to showing that if $\delta_1, [v_{x_1}/x] \cdot e_l \sim \delta_2, [v_{x_2}/x] \cdot e_r :: \tau';\ \delta$, then $\delta_1 \cdot e_l[e_x/x] \sim \delta_2 \cdot e_r[e_x/x] :: \tau';\ \delta$; we find $\delta_1 \cdot e_l[e_x/x] \rightrightarrows^* \delta_1, [v_{x_1}/x] \cdot e_l$ and $\delta_2 \cdot e_r[e_x/x] \rightrightarrows^* \delta_2, [v_{x_2}/x] \cdot e_r$

because substituting multiple parallel reduction is parallel reduction (Corollary C.3). The logical relation is closed under parallel reduction (Lemma B.20), and so $\delta_1 \cdot e_l[e_x/x] \sim \delta_2 \cdot e_r[e_x/x] :: \tau'; \delta$.

□

Theorem B.22 (LR Fundamental Property). *If* $\Gamma \vdash e :: \tau$, *then* $\Gamma \vdash e \sim e :: \tau$.

Proof. The proof goes by induction on the derivation tree:

T-Sub By inversion of the rule we have

(1) $\Gamma \vdash e :: \tau'$

(2) $\Gamma \vdash \tau' \preceq \tau$

By IH on (1) we have

(3) $\Gamma \vdash e \sim e :: \tau'$

By (3), (4), and Lemma B.14 we have $\Gamma \vdash e \sim e :: \tau$.

T-Con By Lemma B.15.

T-Self By inversion of the rule, we have:

(1) $\Gamma \vdash e :: \{z{:}b \mid r\}$.

(2) By the IH on (1), we have:
$\Gamma \vdash e \sim e :: \{z{:}b \mid r\}$.

(3) We fix a δ such that:
$\delta \in \Gamma$ and
$\delta_1 \cdot e \sim \delta_2 \cdot e :: \{z{:}b \mid r\}; \delta$

(4) There must exist v_1 and v_2 such that:
$\delta_1 \cdot e \hookrightarrow^* v_1$
$\delta_2 \cdot e \hookrightarrow^* v_2$
$v_1 \sim v_2 :: \{z{:}b \mid r\}; \delta$

(5) By definition, $v_1 = v_2 = c$ such that:
$\vdash_B c :: b$
$\delta_1 \cdot r[c/x] \hookrightarrow^* \texttt{true}$
$\delta_2 \cdot r[c/x] \hookrightarrow^* \texttt{true}$

(6) We find $v_1 \sim v_2 :: \{z{:}b \mid z ==_b e\}; \delta$, because:
$\vdash_B c :: b$ by (5)
$\delta_1 \cdot (z ==_b e)[c/z] \hookrightarrow^* \texttt{true}$ because $\delta_1 \cdot e \hookrightarrow^* v_1 = c$ by (4)
$\delta_2 \cdot (z ==_b e)[c/z] \hookrightarrow^* \texttt{true}$ because $\delta_2 \cdot e \hookrightarrow^* v_2 = c$ by (4)

T-Var By inversion of the rule and Lemma B.17.

T-Lam By hypothesis:

(1) $\Gamma \vdash \lambda x{:}\tau_x.\, e :: x{:}\tau_x \to \tau$

By inversion of the rule we have

(2) $\Gamma, x : \tau_x \vdash e :: \tau$

(3) $\Gamma \vdash \tau_x$

By inductive hypothesis on (2) we have

(4) $\Gamma, x : \tau_x \vdash e \sim e :: \tau$

We fix a δ, v_{x_1}, and v_{x_2} so that

(5) $\delta \in \Gamma$

(6) $v_{x_1} \sim v_{x_2} :: \tau_x; \delta$

Let $\delta' \doteq \delta, (v_{x_1}, v_{x_2})/x$.

By the definition of the logical relation on open terms, (4), (5), and (6) we have

(7) $\delta'_1 \cdot e \sim \delta'_2 \cdot e :: \tau; \delta'$

By the definition of substitution

(8) $\delta_1 \cdot e[v_{x_1}/x] \sim \delta_2 \cdot e[v_{x_2}/x] :: \tau;\ \delta'$

By the definition of the logical relation on closed expressions

(9) $\delta_1 \cdot e[v_{x_1}/x] \hookrightarrow^* v_1$, $\delta_2 \cdot e[v_{x_2}/x] \hookrightarrow^* v_2$, and $v_1 \sim v_2 :: \tau;\ \delta'$

By the definition and determinism of operational semantics

(10) $\delta_1 \cdot (\lambda x{:}\tau_x.\ e)\ v_{x_1} \hookrightarrow^* v_1$, $\delta_2 \cdot (\lambda x{:}\tau_x.\ e)\ v_{x_2} \hookrightarrow^* v_2$, and $v_1 \sim v_2 :: \tau;\ \delta'$

By (6) and the definition of logical relation on function values,

(11) $\delta_1 \cdot \lambda x{:}\tau_x.\ e \sim \delta_2 \cdot \lambda x{:}\tau_x.\ e :: x{:}\tau_x \to \tau;\ \delta$

Thus, by the definition of the logical relation, $\Gamma \vdash \lambda x{:}\tau_x.\ e \sim \lambda x{:}\tau_x.\ e :: x{:}\tau_x \to \tau$

T-App By hypothesis:

(1) $\Gamma \vdash e\ e_x :: \tau[e_x/x]$

By inversion we get

(2) $\Gamma \vdash e :: x{:}\tau_x \to \tau$

(3) $\Gamma \vdash e_x :: \tau_x$

By inductive hypothesis

(3) $\Gamma \vdash e \sim e :: x{:}\tau_x \to \tau$

(4) $\Gamma \vdash e_x \sim e_x :: \tau_x$

We fix a $\delta \in \Gamma$. Then, by the definition of the logical relation on open terms

(5) $\delta_1 \cdot e \sim \delta_2 \cdot e :: (x{:}\tau_x \to \tau);\ \delta$

(6) $\delta_1 \cdot e_x \sim \delta_2 \cdot e_x :: \tau_x;\ \delta$

By the definition of the logical relation on open terms:

(7) $\delta_1 \cdot e \hookrightarrow^* v_1$

(8) $\delta_2 \cdot e \hookrightarrow^* v_2$

(9) $v_1 \sim v_2 :: x{:}\tau_x \to \tau;\ \delta$

(10) $\delta_1 \cdot e_x \hookrightarrow^* v_{x_1}$

(11) $\delta_2 \cdot e_x \hookrightarrow^* v_{x_2}$

(12) $v_{x_1} \sim v_{x_2} :: \tau_x;\ \delta$

By (7) and (10)

(13) $\delta_1 \cdot e\ e_x \hookrightarrow^* v_1\ v_{x_1}$

By (8) and (11)

(14) $\delta_2 \cdot e\ e_x \hookrightarrow^* v_2\ v_{x_2}$

By (9), (12), and the definition of logical relation on functions:

(15) $v_1\ v_{x_1} \sim v_2\ v_{x_2} :: \tau;\ \delta, (v_{x_1}, v_{x_2})/x$

By (13), (14), (15), and Lemma B.19

(16) $\delta_1 \cdot e\ e_x \sim \delta_2 \cdot e\ e_x :: \tau;\ \delta, (v_{x_1}, v_{x_2})/x$

By (10), (11), (16), and Lemma B.21

(17) $\delta_1 \cdot e\ e_x \sim \delta_2 \cdot e\ e_x :: \tau[e_x/x];\ \delta$

So from the definition of logical relations, $\Gamma \vdash e\ e_x \sim e\ e_x :: \tau[e_x/x]$.

T-Eq-Base By hypothesis:

(1) $\Gamma \vdash \mathsf{bEq}_b\ e_l\ e_r\ e :: \mathsf{PEq}_b\ \{e_l\}\ \{e_r\}$

By inversion of the rule:

(2) $\Gamma \vdash e_l :: \tau_r$

(3) $\Gamma \vdash e_r :: \tau_l$

(4) $\Gamma \vdash \tau_r\ \preceq\ b$

(5) $\Gamma \vdash \tau_l\ \preceq\ b$

(6) $\Gamma, r : \tau_r, l : \tau_l \vdash e :: \{x{:}()\mid l ==_b r\}$

By inductive hypothesis on (2), (3), and (6) we have

(7) $\Gamma \vdash e_l \sim e_l :: \tau_r$

(8) $\Gamma \vdash e_r \sim e_r :: \tau_l$

(9) $\Gamma, r : \tau_r, l : \tau_l \vdash e \sim e :: \{x{:}() \mid l ==_b r\}$

We fix $\delta \in \Gamma$. Then (7) and (8) become

(10) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_l :: \tau_r;\ \delta$

(11) $\delta_1 \cdot e_r \sim \delta_2 \cdot e_r :: \tau_l;\ \delta$

By the definition of the logical relation on closed terms:

(12) $\delta_1 \cdot e_l \hookrightarrow^* v_{l_1}$

(13) $\delta_2 \cdot e_l \hookrightarrow^* v_{l_2}$

(14) $v_{l_1} \sim v_{l_2} :: \tau_l;\ \delta$

(15) $\delta_1 \cdot e_r \hookrightarrow^* v_{r_1}$

(16) $\delta_2 \cdot e_r \hookrightarrow^* v_{r_2}$

(17) $v_{r_1} \sim v_{r_2} :: \tau_r;\ \delta$

We define $\delta' \doteq \delta, (v_{r_1}, v_{r_2})/r, (v_{l_1}, v_{l_2})/l$.

By (9), (14), and (17) we have

(18) $\delta'{}_1 \cdot e \sim \delta'{}_2 \cdot e :: \{x{:}() \mid l ==_b r\};\ \delta'$

By the definition of the logical relation on closed terms:

(19) $\delta' \cdot e \hookrightarrow^* v_1$

(20) $\delta' \cdot e \hookrightarrow^* v_2$

(21) $v_1 \sim v_2 :: \{x{:}() \mid l ==_b r\};\ \delta'$

By (21) and the definition of logical relation on basic values:

(19) $\delta'_1 \cdot (l ==_b r) \hookrightarrow^* \texttt{true}$

(20) $\delta'_2 \cdot (l ==_b r) \hookrightarrow^* \texttt{true}$

By the definition of $==_b$

(21) $v_{l_1} = v_{r_1}$

(22) $v_{l_2} = v_{r_2}$

By (14) and (17) and since τ_l and τ_r are basic types

(23) $v_{l_1} = v_{l_2}$

(24) $v_{r_1} = v_{r_2}$

By (21) and (24)

(25) $v_{l_1} = v_{r_2}$

By the definition of the logical relation on basic types

(26) $v_{l_1} \sim v_{r_2} :: b;\ \delta$

By which, (12), (16), and Lemma B.19

(27) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: b;\ \delta$

By (12), (15), and (19)

(28) $\delta_1 \cdot \mathsf{bEq}_b\ e_l\ e_r\ e \hookrightarrow^* \mathsf{bEq}_b\ v_{l_1}\ v_{r_1}\ v_1$

By (13), (16), and (20)

(29) $\delta_2 \cdot \mathsf{bEq}_b\ e_l\ e_r\ e \hookrightarrow^* \mathsf{bEq}_b\ v_{l_2}\ v_{r_2}\ v_2$

By (27) and the definition of the logical relation on EqRT

(30) $\mathsf{bEq}_b\ v_{l_1}\ v_{r_1}\ v_1 \sim \mathsf{bEq}_b\ v_{l_2}\ v_{r_2}\ v_2 :: \mathsf{PEq}_b\ \{e_l\}\ \{e_r\};\ \delta.$

By (28), (29), and (30)

(31) $\delta_1 \cdot \mathsf{bEq}_b\ e_l\ e_r\ e \sim \delta_2 \cdot \mathsf{bEq}_b\ e_l\ e_r\ e :: \mathsf{PEq}_b\ \{e_l\}\ \{e_r\};\ \delta.$

So, by the definition on the logical relation, $\Gamma \vdash \mathsf{bEq}_b\ e_l\ e_r\ e \sim \mathsf{bEq}_b\ e_l\ e_r\ e :: \mathsf{PEq}_b\ \{e_l\}\ \{e_r\}$.

T-Eq-Fun By hypothesis

(1) $\Gamma \vdash \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ e_l\ e_r\ e :: \mathsf{PEq}_{x:\tau_x\rightarrow\tau}\ \{e_l\}\ \{e_r\}$

By inversion of the rule

(2) $\Gamma \vdash e_l :: \tau_r$

(3) $\Gamma \vdash e_r :: \tau_l$

(4) $\Gamma \vdash \tau_r \;\preceq\; x{:}\tau_x \rightarrow \tau$

(5) $\Gamma \vdash \tau_l \;\preceq\; x{:}\tau_x \rightarrow \tau$

(6) $\Gamma, r : \tau_r, l : \tau_l \vdash e :: (x{:}\tau_x \rightarrow \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\})$

(7) $\Gamma \vdash x{:}\tau_x \rightarrow \tau$

By inductive hypothesis on (2), (3), and (6) we have

(8) $\Gamma \vdash e_l \sim e_l :: \tau_r$

(9) $\Gamma \vdash e_r \sim e_r :: \tau_l$

(10) $\Gamma, r : \tau_r, l : \tau_l \vdash e \sim e :: (x{:}\tau_x \rightarrow \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\})$

By (8), (9), and Lemma B.14

(11) $\Gamma \vdash e_l \sim e_l :: x{:}\tau_x \rightarrow \tau$

(12) $\Gamma \vdash e_r \sim e_r :: x{:}\tau_x \rightarrow \tau$

We fix $\delta \in \Gamma$. Then (11), and (12) become

(13) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_l :: x{:}\tau_x \rightarrow \tau;\ \delta$

(14) $\delta_1 \cdot e_r \sim \delta_2 \cdot e_r :: x{:}\tau_x \rightarrow \tau;\ \delta$

By the definition of the logical relation on closed terms:

(15) $\delta_1 \cdot e_l \hookrightarrow^* v_{l_1}$

(16) $\delta_2 \cdot e_l \hookrightarrow^* v_{l_2}$

(17) $v_{l_1} \sim v_{l_2} :: x{:}\tau_x \rightarrow \tau;\ \delta$

(18) $v_{l_1} \sim v_{l_2} :: \tau_l;\ \delta$

(19) $\delta_1 \cdot e_r \hookrightarrow^* v_{r_1}$

(20) $\delta_2 \cdot e_r \hookrightarrow^* v_{r_2}$

(21) $v_{r_1} \sim v_{r_2} :: x{:}\tau_x \rightarrow \tau;\ \delta$

(22) $v_{r_1} \sim v_{r_2} :: \tau_r;\ \delta$

We fix v_{x_1} and v_{x_2} so that $v_{x_1} \sim v_{x_2} :: \tau_x;\ \delta$. Let $\delta_x \doteq \delta, (v_{x_1}, v_{x_2})/x$.

By the definition on the logical relation on function values, (17) and (21) become

(23) $v_{l_1}\ v_{x_1} \sim v_{l_2}\ v_{x_2} :: \tau;\ \delta_x$

(24) $v_{r_1}\ v_{x_1} \sim v_{r_2}\ v_{x_2} :: \tau;\ \delta_x$

Let $\delta_{lr} \doteq \delta, (v_{r_1}, v_{r_2})/r, (v_{l_1}, v_{l_2})/l$.

By the definition of the logical relation on closed terms, (10) becomes:

(25) $\delta_{lr} \cdot e \hookrightarrow^* v_1$

(26) $\delta_{lr} \cdot e \hookrightarrow^* v_2$

(27) $v_1 \sim v_2 :: x{:}\tau_x \rightarrow \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\};\ \delta_{lr}$

By (27) and the definition of logical relation on function values:

(28) $v_1\ v_{x_1} \sim v_2\ v_{x_2} :: \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\};\ \delta_{lr}, (v_{x_1}, v_{x_2})/x$

By the definition of the logical relation on EqRT

(29) $v_{l_1}\ v_{x_1} \sim v_{r_2}\ v_{x_2} :: \tau;\ \delta_{lr}, (v_{x_1}, v_{x_2})/x$

By the definition of logical relations on function values

(30) $v_{l_1} \sim v_{r_2} :: x{:}\tau_x \rightarrow \tau;\ \delta_{lr}$

By (7), l and r do not appear free in the relation, so

(31) $v_{l_1} \sim v_{r_2} :: x{:}\tau_x \rightarrow \tau;\ \delta$

By which, (15), (20), and Lemma B.19

(32) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: x{:}\tau_x \rightarrow \tau;\ \delta$

By (15), (19), and (25)

(33) $\delta_1 \cdot \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ e_l\ e_r\ e \hookrightarrow^* \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ v_{l_1}\ v_{r_1}\ v_1$

By (16), (20), and (26)

(34) $\delta_2 \cdot \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ e_l\ e_r\ e \hookrightarrow^* \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ v_{l_2}\ v_{r_2}\ v_2$

By (32) and the definition of the logical relation on EqRT

(35) $\mathsf{xEq}_{\tau_x:\tau\rightarrow}\ v_{l_1}\ v_{r_1}\ v_1 \sim \mathsf{xEq}_{\tau_x:\tau\rightarrow}\ v_{l_2}\ v_{r_2}\ v_2 :: \mathsf{PEq}_{x:\tau_x\rightarrow\tau}\ \{e_l\}\ \{e_r\};\ \delta.$

By (33), (34), and (35)

(36) $\delta_1 \cdot \mathsf{xEq}_{\tau_x:\tau\to}\ e_l\ e_r\ e \sim \delta_2 \cdot \mathsf{xEq}_{\tau_x:\tau\to}\ e_l\ e_r\ e :: \mathsf{PEq}_{x:\tau_x\to\tau}\ \{e_l\}\ \{e_r\};\ \delta.$

So, by the definition on the logical relation, $\Gamma \vdash \mathsf{xEq}_{\tau_x:\tau\to}\ e_l\ e_r\ e \sim \mathsf{xEq}_{\tau_x:\tau\to}\ e_l\ e_r\ e :: \mathsf{PEq}_{x:\tau_x\to\tau}\ \{e_l\}\ \{e_r\}$.

□

B.5 The Logical Relation and the EqRT Type are Equivalence Relations

Theorem B.23 (The logical relation is an equivalence relation). *$\Gamma \vdash e_1 \sim e_2 :: \tau$ is reflexive, symmetric, and transivite.*

- *Reflexivity: If $\Gamma \vdash e :: \tau$, then $\Gamma \vdash e \sim e :: \tau$.*
- *Symmetry: If $\Gamma \vdash e_1 \sim e_2 :: \tau$, then $\Gamma \vdash e_2 \sim e_1 :: \tau$.*
- *Transitivity: If $\Gamma \vdash e_2 :: \tau$ and $\Gamma \vdash e_1 \sim e_2 :: \tau$ and $\Gamma \vdash e_2 \sim e_3 :: \tau$, then $\Gamma \vdash e_1 \sim e_3 :: \tau$.*

Proof. **Reflexivity:** This is exactly the Fundamental Property B.22.

Symmetry: Let $\bar{\delta}$ be defined such that $\bar{\delta}_1(x) = \delta_2(x)$ and $\bar{\delta}_2(x) = \delta_1(x)$. First, we prove that $v_1 \sim v_2 :: \tau;\ \delta$ implies $v_2 \sim v_1 :: \tau;\ \bar{\delta}$, by structural induction on τ.

- $\tau \doteq \{z{:}b \mid r\}$. This case is immediate: we have to show that $c \sim c :: \{z{:}b \mid r\};\ \bar{\delta}$ given $c \sim c :: \{z{:}b \mid r\};\ \delta$. But the definition in this case is itself symmetric: the predicate goes to `true` under both substitutions.
- $\tau \doteq x{:}\tau'_x \to \tau'$. We fix v_{x_1} and v_{x_2} so that
  (1) $v_{x_1} \sim v_{x_2} :: \tau'_x;\ \delta$
  By the definition of logical relations on open terms and inductive hypothesis
  (2) $v_{x_2} \sim v_{x_1} :: \tau'_x;\ \bar{\delta}$
  By the definition on logical relations on functions
  (3) $v_1\ v_{x_1} \sim v_2\ v_{x_2} :: \tau';\ \delta, (v_{x_1}, v_{x_2})/x$
  By the definition of logical relations on open terms and since the expressions $v_1\ v_{x_1}$ and $v_2\ v_{x_2}$ are closed, By the inductive hypothesis on τ':
  (4) $v_2\ v_{x_2} \sim v_1\ v_{x_1} :: \tau';\ \bar{\delta}, x : \tau'_x$
  By (2) and the definition of logical relations on open terms
  (5) $v_2\ v_{x_2} \sim v_1\ v_{x_1} :: \tau';\ \bar{\delta}, (v_{x_2}, v_{x_1})/x$
  By the definition of the logical relation on functions, we conclude that $v_2 \sim v_1 :: x{:}\tau'_x \to \tau';\ \bar{\delta}$
- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$. By assumption,
  (1) $v_1 \sim v_2 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\};\ \delta$
  By the definition of the logical relation on EqRT types
  (2) $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: \tau';\ \delta$
  i.e., $\delta_1 \cdot (e_l) \hookrightarrow^* v_l$ and similarly for v_r such that $v_l \sim v_r :: \tau';\ \delta$.
  By the IH on τ', we have:
  (3) $v_r \sim v_l :: \tau';\ \bar{\delta}$
  And so, by the definition of the LR on equality proofs:
  (4) $v_2 \sim v_1 :: \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\};\ \bar{\delta}$

Next, we show that $\delta \in \Gamma$ implies $\bar{\delta} \in \Gamma$. We go by structural induction on Γ.

- $\Gamma = \cdot$. This case is trivial.
- $\Gamma = \Gamma', x : \tau$. For $x : \tau$, we know that $\delta_1(x) \sim \delta_2(x) :: \tau;\ \delta$. By the IH on τ, we find $\delta_2(x) \sim \delta_1(x) :: \tau;\ \bar{\delta}$, which is just the same as $\bar{\delta}_1(x) \sim \bar{\delta}_2(x) :: \tau;\ \bar{\delta}$. By the IH on Γ', we can use similar reasoning to find $\bar{\delta}_1(y) \sim \bar{\delta}_2(y) :: \tau';\ \bar{\delta}$ for all $y : \tau' \in \Gamma'$.

Now, suppose $\Gamma \vdash e_1 \sim e_2 :: \tau$; we must show $\Gamma \vdash e_2 \sim e_1 :: \tau$. We fix $\delta \in \Gamma$; we must show $\delta_1 \cdot e_2 \sim \delta_2 \cdot e_1 :: \tau;\ \delta$, i.e., there must exist v_1 and v_2 such that $\delta_1 \cdot e_2 \hookrightarrow^* v_2$ and $\delta_2 \cdot e_1 \hookrightarrow^* v_1$ and $v_2 \sim v_1 :: \tau;\ \delta$. We have $\delta \in \Gamma$, and so $\bar{\delta} \in \Gamma$ by our second lemma. But then, by assumption, we have v_1 and v_2 such that $\bar{\delta_1} \cdot e_1 \hookrightarrow^* v_1$ and $\bar{\delta_2} \cdot e_2 \hookrightarrow^* v_2$ and $v_1 \sim v_2 :: \tau;\ \bar{\delta}$. Our first lemma then yields $v_2 \sim v_1 :: \tau;\ \delta$ as desired.

Transitivity: First, we prove an inner property: if $\delta \in \Gamma$ and $v_1 \sim v_2 :: \tau;\ \delta$ and $v_2 \sim v_3 :: \tau;\ \delta$, then $v_1 \sim v_3 :: \tau;\ \delta$. We go by structural induction on the type index τ.

- $\tau \doteq \{z{:}b \mid r\}$. Here all of the values must be the fixed constant c. Furthermore, we must have $\delta_1 \cdot r[c/x] \hookrightarrow^* \texttt{true}$ and $\delta_2 \cdot r[c/x] \hookrightarrow^* \texttt{true}$, so we can immediately find $v_1 \sim v_3 :: \tau;\ \delta$.
- $\tau \doteq x{:}\tau_x' \rightarrow \tau'$.
  Let $v_l \sim v_r :: \tau_x';\ \delta$ be given. We must show that $v_1 \sim v_3 :: \tau;\ \delta, (v_l, v_r)/x$. We know by assumption that: $v_1\ v_l \sim v_2\ v_r :: \tau';\ \delta, (v_l, v_r)/x$ and $v_2\ v_l \sim v_3\ v_r :: \tau';\ \delta, (v_l, v_r)/x$. By the IH on τ', we find $v_1\ v_l \sim v_3\ v_r :: \tau';\ \delta, (v_l, v_r)/x$; which gives $v_1 \sim v_3 :: \tau;\ \delta, (v_l, v_r)/x$.
- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$.
  To find $v_1 \sim v_3 :: \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\};\ \delta$, we merely need to find that $\delta_1 \cdot e_l \sim \delta_2 \cdot e_r :: \tau;\ \delta$, which we have by inversion on $v_1 \sim v_2 :: \mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\};\ \delta$.

With our proof that the value relation is transitive in hand, we turn our attention to the open relation. Suppose $\Gamma \vdash e_1 \sim e_2 :: \tau$ and $\Gamma \vdash e_2 \sim e_3 :: \tau$; we want to see $\Gamma \vdash e_1 \sim e_3 :: \tau$. Let $\delta \in \Gamma$ be given. We have $\delta_1 \cdot e_1 \sim \delta_2 \cdot e_2 :: \tau;\ \delta$ and $\delta_1 \cdot e_2 \sim \delta_2 \cdot e_3 :: \tau;\ \delta$. By the definition of the logical relations, we have $\delta_1 \cdot e_1 \hookrightarrow^* v_1$, $\delta_2 \cdot e_2 \hookrightarrow^* v_2$, $\delta_1 \cdot e_2 \hookrightarrow^* v_2'$, $\delta_2 \cdot e_3 \hookrightarrow^* v_3$, $v_1 \sim v_2 :: \tau;\ \delta$, and $v_2' \sim v_3 :: \tau;\ \delta$.

Moreover, we know that e_2 is well typed, so by the fundamental theorem (Theorem B.22), we know that $\Gamma \vdash e_2 \sim e_2 :: \tau$, and so $v_2 \sim v_2' :: \tau;\ \delta$.

By our transitivity lemma on the value relation, we can find that v_1 is equivalent to v_2 is equivalent to v_2' is equivalent to v_3, and so $v_1 \sim v_3 :: \tau;\ \delta$.

□

$$\begin{array}{rcl}
\mathsf{pf} & : & e \rightarrow e \rightarrow \tau \\
\mathsf{pf}(l, r, b) & = & \{x{:}() \mid l ==_b r\} \\
\mathsf{pf}(l, r, x{:}\tau_x \rightarrow \tau) & = & x{:}\tau_x \rightarrow \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\}
\end{array}$$

Our propositional equality $\mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\}$ is a reflection of the logical relation, so it is unsurprising that it is also an equivalence relation. We can prove that our propositional equality is treated as an equivalence relation by the syntactic type system. There are some tiny wrinkles in the syntactic system: symmetry and transitivity produce normalized proofs, but reflexivity produces unnormalized ones in order to generate the correct invariant types τ_l and τ_r in the base case.

Theorem B.24 (EqRT is an equivalence relation). $\mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$ *is reflexive, symmetric, and transitive on equable types. That is, for all τ that contain only refinements and functions:*

- *Reflexivity: If* $\Gamma \vdash e :: \tau$*, then there exists* e_p *such that* $\Gamma \vdash e_p :: \mathsf{PEq}_\tau\ \{e\}\ \{e\}$.
- *Symmetry:* $\forall \Gamma, \tau, e_1, e_2, v_{12}$*. if* $\Gamma \vdash v_{12} :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$*, then there exists* v_{21} *such that* $\Gamma \vdash v_{21} :: \mathsf{PEq}_\tau\ \{e_2\}\ \{e_1\}$.
- *Transitivity:* $\forall \Gamma, \tau, e_1, e_2, e_3, v_{12}, v_{23}$*. if* $\Gamma \vdash v_{12} :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$ *and* $\Gamma \vdash v_{23} :: \mathsf{PEq}_\tau\ \{e_2\}\ \{e_3\}$*, then there exists* v_{13} *such that* $\Gamma \vdash v_{13} :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_3\}$.

Proof. **Reflexivity**: We strengthen the IH, simultaneously proving that there exist e_p, e_{pf} and $\Gamma \vdash \tau_l \preceq \tau$ and $\Gamma \vdash \tau_r \preceq \tau$ such that $\Gamma, l : \tau_l, r : \tau_r \vdash e_{\mathsf{pf}} :: \mathsf{pf}(e, e, \tau)$ and $\Gamma \vdash e_p :: \mathsf{PEq}_\tau\ \{e\}\ \{e\}$ by induction on τ, leaving e general.

- $\tau \doteq \{x{:}b \mid e'\}$.

(1) Let $e_{\mathsf{pf}} = ()$.
(2) Let $e_p = \mathsf{bEq}_b\ e\ e\ e_{\mathsf{pf}}$.
(3) Let $\tau_l = \tau_r = \{x{:}b \mid x ==_b e\}$.

(4) We have $\Gamma \vdash x ==_b e \ \preceq \ \tau$ by S-Base and semantic typing.
(5) We find $\Gamma \vdash e_p :: \mathsf{PEq}_b\ \{e\}\ \{e\}$ by T-Eq-Base, with $e_l = e_r = e$. We must show:
  (a) $\Gamma \vdash e_l :: \tau_l$ and $\Gamma \vdash e_r :: \tau_r$, i.e., $\Gamma \vdash e :: \{x{:}b \mid x ==_b e\}$;
  (b) $\Gamma \vdash \tau_r \ \preceq \ \{x{:}b \mid \mathtt{true}\}$ and $\Gamma \vdash \tau_l \ \preceq \ \{x{:}b \mid \mathtt{true}\}$; and
  (c) $\Gamma, r : \tau_r, l : \tau_l \vdash e_{\mathsf{pf}} :: \{x{:}() \mid l ==_b r\}$.
(6) We find (5a) by T-Self.
(7) We find (5b) immediately by S-Base.
(8) We find (5c) by T-Var, using T-Sub to see that if $l, r : \{x{:}b \mid x ==_b e\}$ then $\mathtt{unit}$ will be typeable at the refinement where both l and r are equal to e.

- $\tau \doteq x{:}\tau_x \to \tau'$.

(1) $\Gamma, x : \tau_x \vdash e\ x :: \tau[x/x]$ by T-App and T-Var, noting that $\tau[x/x] = \tau$.
(2) By the IH on $\Gamma, x : \tau_x \vdash e\ x :: \tau'[x/x] = \tau'$, there exist $e'_p, e'_{\mathsf{pf}}, \tau'_l$, and τ'_r such that:
  (a) $x : \tau_x \vdash \tau'_l \ \preceq \ \tau$ and $x : \tau_x \vdash \tau'_r \ \preceq \ \tau$;
  (b) $\Gamma, x : \tau_x, l : \tau'_l, r : \tau'_r \vdash e'_{\mathsf{pf}} :: \mathsf{pf}(e\ x, e\ x, \tau')$; and
  (c) $\Gamma, x : \tau_x \vdash e'_p :: \mathsf{PEq}_{\tau'}\ \{e\ x\}\ \{e\ x\}$.
(3) If $\tau' = \{x{:}() \mid \tau'\}e\ xe\ x$, then $\mathsf{pf}(e\,x, ex, b) = \{x{:}() \mid ex ==_b ex\}$; otherwise, $\mathsf{pf}(l, r, x{:}\tau_x \to \tau) = x{:}\tau_x \to \mathsf{PEq}_\tau\ \{e\ x\}\ \{e\ x\}$.
In the former case, let $e''_{\mathsf{pf}} = \mathsf{bEq}_b\ (e\ x)(e\ x)e'_{\mathsf{pf}}$. In the latter case, let $e''_{\mathsf{pf}} = e'_{\mathsf{pf}}$.
Either way, we have $\Gamma, x : \tau_x, l : \tau'_l, r : \tau'_r \vdash e''_{\mathsf{pf}} :: \mathsf{PEq}_{\tau'}\ \{e\ x\}\ \{e\ x\}$ by T-Eq-Base or T-Eq-Fun, respectively.
(4) Let $e_{\mathsf{pf}} = x{:}\tau_x \to e''_{\mathsf{pf}}$.
(5) Let $e_p = \mathsf{xEq}_{x:\tau_x \to \tau}\ e\ e\ e_{\mathsf{pf}}$.
(6) Let $e_l = e_r = e$ and $\tau_l = x{:}\tau_x \to \tau'_l$ and $\tau_r = x{:}\tau_x \to \tau'_r$.
(7) We find subtyping by S-Fun and (2a).
(8) By T-Eq-Fun. We must show:
  (a) $\Gamma \vdash e_l :: \tau_l$ and $\Gamma \vdash e_r :: \tau_r$;
  (b) $\Gamma \vdash \tau_l \ \preceq \ x{:}\tau_x \to \tau$ and $\Gamma \vdash \tau_r \ \preceq \ x{:}\tau_x \to \tau$;
  (c) $\Gamma, r : \tau_r, l : \tau_l \vdash e_{\mathsf{pf}} :: (x{:}\tau_x \to \mathsf{PEq}_\tau\ \{l\ x\}\ \{r\ x\})$
  (d) $\Gamma \vdash x{:}\tau_x \to \tau$
(9) We find (8a) by assumption, T-Sub, and (7).
(10) We find (8b) by (7).
(11) We find (8c) by T-Lam and (2b).

- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_1\}\ \{e_2\}$. These types are not equable, so we ignore them.

Symmetry: By induction on τ.

- $\tau \doteq \{x{:}b \mid e\}$.

(1) We have $\Gamma \vdash v_{12} :: \mathsf{PEq}_b\ \{e_1\}\ \{e_2\}$.
(2) By canonical forms, $v_{12} = \mathsf{bEq}_b\ e_l\ e_r\ v_p$ such that $\Gamma \vdash e_l :: \tau_l$ and $\Gamma \vdash e_r :: \tau_r$ (for some τ_l and τ_r that are refinements of b) and $\Gamma, r : \tau_r, l : \tau_l \vdash v_p :: \{x{:}() \mid l ==_b r\}$ (Lemma B.12).
(3) Let $v_{21} = \mathsf{bEq}_b\ e_r\ e_l\ v_p$.
(4) By T-Eq-Base, swapping τ_l and τ_r from (2). We already have appropriate typing and subtyping derivations; we only need to see $\Gamma, l : \tau_l, r : \tau_r \vdash v_p :: \{x{:}() \mid r ==_b l\}$.
(5) We have $\Gamma, l : \tau_l, r : \tau_r \vdash \{x{:}() \mid r ==_b l\} \ \preceq \ \{x{:}() \mid l ==_b r\}$ by S-Base and symmetry of $(==_b)$.

- $\tau \doteq x{:}\tau_x \to \tau'$.

(1) We have $\Gamma \vdash v_{12} :: \mathsf{PEq}_{x:\tau_x \to \tau'}\ \{e_1\}\ \{e_2\}$.

(2) By canonical forms, $v_{12} = \mathsf{xEq}_{x{:}\tau'_x \to \tau''}\ e_l\ e_r\ v_p$ such that $\tau_x \vdash \tau'_x \ \preceq\ and\ \tau'' \vdash \tau' \ \preceq\ and$ $\Gamma \vdash e_l :: \tau_l$ and $\Gamma \vdash e_r :: \tau_r$ (for some τ_l and τ_r that are subtypes of $x{:}\tau'_x \to \tau''$) and $\Gamma, r : \tau_r, l : \tau_l \vdash v_p :: x{:}\tau'_x \to \mathsf{PEq}_{\tau''}\ \{l\ x\}\ \{r\ x\}$.

(3) By canonical forms, this time on v_p from (2), $v_p = \text{T-L}{\small\text{AM}} x \tau'_x e_p$ such that $\Gamma \vdash \tau_x \ \preceq\ \tau'_x$ and $\Gamma, r : \tau_r, l : \tau_l, x : \tau'_x \vdash e :: \tau'''$ such that $\Gamma, r : \tau_r, l : \tau_l, x : \tau'_x \vdash \tau''' \ \preceq\ \mathsf{PEq}_{\tau''}\ \{l\ x\}\ \{r\ x\}$.

(4) By T-Sub, (3), and the IH on $\mathsf{PEq}_{\tau''}\ \{l\ x\}\ \{r\ x\}$, we know there exists some e'_p such that $\Gamma, l : \tau_l, r : \tau_r, x : \tau'_x \vdash e'_p :: \mathsf{PEq}_{\tau''}\ \{r\ x\}\ \{l\ x\}$.

(5) Let $v'_p = x{:}\tau'_x \to e'_p$.

(6) By (4) and T-Lam, and T-Sub (using subtyping from (3) and (2)), $\Gamma, l : \tau_l, r : \tau_r \vdash v'_p :: \mathsf{PEq}_{x{:}\tau_x \to \tau'}\ \{e_r\ x\}\ \{e_l\ x\}$.

(7) Let $v_{21} = \mathsf{xEq}_{x{:}\tau_x \to \tau'}\ e_r\ e_l\ v'_p$.

(8) By T-Eq-Base, with (6) for the proof and (3) and (2) for the rest.

- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_1\}\ \{e_2\}$. These types are not equable, so we ignore them.

Transitivity: By induction on τ.

- $\tau \doteq \{x{:}b \mid e\}$.

(1) We have $\Gamma \vdash v_{12} :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$ and $\Gamma \vdash v_{23} :: \mathsf{PEq}_\tau\ \{e_2\}\ \{e_3\}$.

(2) By canonical forms, $v_{12} = \mathsf{bEq}_b\ e_1\ e_2\ v'_{12}$ such that $\Gamma \vdash e_1 :: \tau_1$ and $\Gamma \vdash e_2 :: \tau_2$ (for some τ_1 and τ_2 that are refinements of b) and $\Gamma, r : \tau_2, l : \tau_1 \vdash v'_{12} :: \{x{:}() \mid l ==_b r\}$. and, similarly, $v_{23} = \mathsf{bEq}_b\ e_1\ e_2\ v'_{23}$ such that $\Gamma \vdash e_2 :: \tau'_2$ and $\Gamma \vdash e_3 :: \tau_3$ (for some τ'_2 and τ_3 that are refinements of b) and $\Gamma, r : \tau_3, l : \tau'_2 \vdash v'_{23} :: \{x{:}() \mid l ==_b r\}$.

(3) By canonical forms again, we know that $v'_{12} = v'_{23} = \texttt{unit}$ and we have:

$$\begin{array}{l} \Gamma, r : \tau_2, l : \tau_1 \vdash \{x{:}() \mid x ==_{()} \texttt{unit}\} \ \preceq\ \{x{:}b \mid \{x{:}() \mid l ==_b r\}\}, \text{and} \\ \Gamma, r : \tau_3, l : \tau'_2 \vdash \{x{:}() \mid x ==_{()} \texttt{unit}\} \ \preceq\ \{x{:}b \mid \{x{:}() \mid l ==_b r\}\}. \end{array}$$

(4) Elaborating on (3), we know that $\forall \theta \in \llbracket \Gamma, r : \tau_2, l : \tau_1 \rrbracket$, we have:

$$\left\llbracket \theta \cdot \{x{:}() \mid x ==_{()} \texttt{unit}\} \right\rrbracket \subseteq \llbracket \theta \cdot \{x{:}() \mid l ==_b r\} \rrbracket$$

and $\forall \theta \in \left\llbracket \Gamma, r : \tau_3, l : \tau'_2 \right\rrbracket$, we have:

$$\left\llbracket \theta \cdot \{x{:}() \mid x ==_{()} \texttt{unit}\} \right\rrbracket \subseteq \llbracket \theta \cdot \{x{:}() \mid l ==_b r\} \rrbracket .$$

(5) Since $\{x{:}() \mid x ==_{()} \texttt{unit}\}$ contains all computations that terminate with $\texttt{unit}$ in all models (Theorem B.1), the right-hand sides of the equations must also hold all unit computations. That is, all choices for l and r_2 (resp. l and r) that are semantically well typed are necessarily equal.

(6) By (5), we can infer that in any given model, τ_1, τ_2, τ'_2, and τ_3 identify just one b-constant. Why must τ_2 and τ'_2 agree? In particular, e_2 has *both* of those types, but by semantic soundness (Theorem B.2), we know that it will go to a value in the appropriate type interpretation. By determinism of evaluation, we know it must be the *same* value. We can therefore conclude that $\forall \theta \in \llbracket \Gamma, r : \tau_3, l : \tau_1 \rrbracket$, $\left\llbracket \theta \cdot \{x{:}() \mid x ==_{()} \texttt{unit}\} \right\rrbracket \subseteq \llbracket \theta \cdot \{x{:}() \mid l ==_b r\} \rrbracket$.

(7) By T-Eq-Base, using τ_1 and τ_3 and $\texttt{unit}$ as the proof. We need to show $\Gamma, r : \tau_3, l : \tau_1 \vdash \texttt{unit} :: \{x{:}() \mid l ==_b r\}$; all other premises follow from (2).

(8) By T-Sub and S-Base, using (6) for the subtyping.

- $\tau \doteq x{:}\tau_x \to \tau'$.

(1) We have $\Gamma \vdash v_{12} :: \mathsf{PEq}_\tau\ \{e_1\}\ \{e_2\}$ and $\Gamma \vdash v_{23} :: \mathsf{PEq}_\tau\ \{e_2\}\ \{e_3\}$.

(2) By canonical forms, we have

$$\begin{array}{rcl} v_{12} & = & \mathsf{xEq}_{x{:}\tau_x \to \tau'}\ e_1\ e_2\ v'_{12} \\ v_{23} & = & \mathsf{xEq}_{x{:}\tau_x \to \tau'}\ e_2\ e_3\ v'_{23} \end{array}$$

where there exist types τ_1, τ_2, τ_2', and τ_3 subtypes of $x{:}\tau_x \to \tau'$ such that

$$\begin{array}{ll} \Gamma \vdash e_1 :: \tau_1 & \Gamma \vdash e_2 :: \tau_2 \\ \Gamma \vdash e_2 :: \tau_2' & \Gamma \vdash e_3 :: \tau_3 \end{array}$$

and there exist types $\tau_{x_{12}}$, $\tau_{x_{23}}$, τ_{12}', and τ_{23}' such that

$$\begin{array}{l} \Gamma, r : \tau_2, l : \tau_1 \vdash v_{p_{12}} :: x{:}\tau_{x_{12}} \to \mathsf{PEq}_{\tau_{12}'}\ \{l\ x\}\ \{r\ x\}, \\ \Gamma, r : \tau_2, l : \tau_1 \vdash \tau_x \ \preceq\ \tau_{x_{12}}, \\ \Gamma, r : \tau_2, l : \tau_1, x : \tau_x \vdash \tau_{12}' \ \preceq\ \tau', \\ \Gamma, r : \tau_3, l : \tau_2' \vdash v_{p_{23}} :: x{:}\tau_x' \to \mathsf{PEq}_{\tau_{23}'}\ \{l\ x\}\ \{r\ x\}, \\ \Gamma, r : \tau_3, l : \tau_2' \vdash \tau_x \ \preceq\ \tau_{x_{23}}, \text{ and} \\ \Gamma, r : \tau_3, l : \tau_2', x : \tau_x \vdash \tau_{23}' \ \preceq\ \tau'. \end{array}$$

(3) By canonical forms on $v_{p_{12}}$ and $v_{p_{23}}$ from (2), we know that:

$$v_{p_{12}} = \lambda x{:}\tau_{x_{12}}.\ e_{12}' \qquad v_{p_{23}} = \lambda x{:}\tau_{x_{23}}.\ e_{23}'$$

such that:

$$\begin{array}{l} \Gamma, r : \tau_2, l : \tau_1, x : \tau_{x_{12}} \vdash e_{12}' :: \tau_{12}'', \\ \Gamma, r : \tau_2, l : \tau_1, x : \tau_{x_{12}} \vdash \tau_{12}'' \ \preceq\ \tau_{12}', \\ \\ \Gamma, r : \tau_3, l : \tau_2', x : \tau_{x_{23}} \vdash e_{23}' :: \tau_{23}'', \\ \Gamma, r : \tau_3, l : \tau_2', x : \tau_{x_{23}} \vdash \tau_{23}'' \ \preceq\ \tau_{23}', \text{ and} \end{array}$$

(4) By strengthening (Lemma B.7) using (2), we can replace x's type with τ_x in both proofs, to find:

$$\begin{array}{l} \Gamma, r : \tau_2, l : \tau_1, x : \tau_x \vdash e_{12}' :: \tau_{12}', \text{and} \\ \Gamma, r : \tau_3, l : \tau_2', x : \tau_x \vdash e_{23}' :: \tau_{23}'. \end{array}$$

Then, by T-Sub, we can relax the type of the proof bodies:

$$\begin{array}{l} \Gamma, r : \tau_2, l : \tau_1, x : \tau_x \vdash e_{12}' :: \tau', \text{and} \\ \Gamma, r : \tau_3, l : \tau_2', x : \tau_x \vdash e_{23}' :: \tau'. \end{array}$$

(5) By (4, (3), and the IH on $\mathsf{PEq}_{\tau'}\ \{l\ x\}\ \{r\ x\}$, we know there exists some proof body e_{13}' such that $\Gamma, r : \tau_3, l : \tau_1 \vdash e_{13}' :: \mathsf{PEq}_{\tau'}\ \{l\ x\}\ \{r\ x\}$.

(6) Let $v_p = x{:}\tau_x \to e_{13}'$.

(7) By (5), and T-Lam.

(8) Let $v_{13} = \mathsf{xEq}_{x:\tau_x \to \tau'}\ e_1\ e_3\ v_p$.

(9) By T-Eq-Base, with (7) for the proof and (2) for the rest.

- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_1\}\ \{e_2\}$. These types are not equable, so we ignore them. □

C PARALLEL REDUCTION AND COTERMINATION

The conventional application rule for dependent types substitutes a term into a type, finding $e_1\ e_2 : \tau[e_2/x]$ when $e_1 : x{:}\tau_x \to \tau$. We define two logical relations: a unary interpretation of types and a binary logical relation characterizing equivalence. Both of these logical relations are defined as fixpoints on types. The type index poses a problem: the function case of these logical relations quantify over values in the relation, but we sometimes need to reason about expressions, not values. If $e \hookrightarrow^* v$, are $\tau[e/x]$ and $\tau[v/x]$ treated the same by our logical relations? We encounter this problem in particular in proof of logical relation compositionality, which is precisely about exchanging expressions in types with the values the expressions reduce to in closing substitutions: for the unary logical relation and binary logical relation (Lemma B.21).

The key technical device to prove these compositionality lemmas is *parallel reduction* (Figure 3). Parallel reduction generalizes our call-by-value relation to allow multiple steps at once, throughout a term—even under a lambda. Parallel reduction is a bisimulation (Lemma C.5 for forward simulation; Corollary C.15 for backward simulation). That is, expressions that parallel reduce to each other go to identical constants or expressions that themselves parallel reduce, and the logical relations put terms that parallel reduce in the same equivalence class.

To prove the compositionality lemmas, we first show that (a) the logical relations are closed under parallel reduction (for the unary relation and Lemma B.20 for the binary relation) and (b) use the backward simulation to change values in the closing substitution to a substituted expression in the type.

Our proof comes in three steps. First, we establish some basic properties of parallel reduction (§C.1). Next, proving the forward simulation is straightforward (§C.2): if $e_1 \rightrightarrows e_2$ and $e_1 \hookrightarrow e_1'$, then either parallel reduction contracted the redex for us and $e_1' \rightrightarrows e_2$ immediately, or the redex is preserved and $e_2 \hookrightarrow e_2'$ such that $e_1' \rightrightarrows e_2'$. Proving the backward simulation is more challenging (§C.3). If $e_1 \rightrightarrows e_2$ and $e_2 \hookrightarrow e_2'$, the redex contracted in e_2 may not yet be exposed. The trick is to show a tighter bisimulation, where the outermost constructors are always the same, with the subparts parallel reducing. We call this relation *congruence* (Figure 4); it's a straightforward restriction of parallel reduction, eliminating β, eq1, and eq2 as outermost constructors (but allowing them deeper inside). The key lemma shows that if $e_1 \rightrightarrows e_2$, then there exists e_1' $e_1 \hookrightarrow^* e_1'$ such that $e_1' \mathrel{\tilde{\rightrightarrows}} e_2$ (Lemma C.11). Once we know that parallel reduction implies reduction to congruent terms, proving that congruence is a backward simulation allows us to reason "up to congruence". In particular, congruence is a sub-relation of parallel reduction, so we find that parallel reduction is a backward simulation. Finally, we can show that $e_1 \rightrightarrows e_2$ implies observational equivalence (§C.4); for our purposes, it suffices to find cotermination at constants (Corollary C.17).

One might think, in light of Takahashi's explanation of parallel reduction [Takahashi 1989], that the simulation techniques we use are too powerful for our needs: why not simply rely on the Church-Rosser property and confluence, which she proves quite simply? Her approach works well when relating parallel reduction to full β-reduction (and/or η-reduction): the transitive closure of her parallel reduction relation is equal to the transitive closure of plain β-reduction (resp. η- and $\beta\eta$-reduction). But we're interested in programming languages, so our underlying reduction relation isn't full β: we use call-by-value, and we will never reduce under lambdas. But even if we were call-by-name, we would have the same issue. Parallel reduction implies reduction, but not to the *same* value, as in her setting. Parallel reduction yields values that are equivalent, *up to parallel reduction and congruence* (see, e.g., Corollary C.13).

C.1 Basic Properties

Lemma C.1 (Parallel reduction is reflexive). *For all e and τ, $e \rightrightarrows e$ and $\tau \rightrightarrows \tau$.*

Proof. By mutual induction on e and τ.

Expressions.

- $e \doteq x$. By var.
- $e \doteq c$. By const.
- $e \doteq \lambda x{:}\tau.\ e'$. By the IHs on τ and e' and lam.
- $e \doteq e_1\ e_2$. By the IH on e_1 and e_2 and app.
- $e \doteq \mathsf{bEq}_b\ e_l\ e_r\ e'$. By the IHs on e_l, e_r, and e' and beq.
- $e \doteq \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e'$. By the IHs on τ_x, τ, e_l, e_r, and e' and xeq.

Types.

- $\tau \doteq \{x{:}b \mid r\}$. By the IH on r (an expression) and ref.
- $\tau \doteq x{:}\tau_x \to \tau'$. By the IHs on τ_x and τ' and fun.

$$\boxed{e \rightrightarrows e}$$

$$\dfrac{}{x \rightrightarrows x}\;\text{var} \qquad \dfrac{}{c \rightrightarrows c}\;\text{const} \qquad \dfrac{\tau \rightrightarrows \tau' \quad e \rightrightarrows e'}{\lambda x{:}\tau.\ e \rightrightarrows \lambda x{:}\tau'.\ e'}\;\text{lam} \qquad \dfrac{e_1 \rightrightarrows e_1' \quad e_2 \rightrightarrows e_2'}{e_1\ e_2 \rightrightarrows e_1'\ e_2'}\;\text{app}$$

$$\dfrac{e \rightrightarrows e' \quad v \rightrightarrows v'}{(\lambda x{:}\tau.\ e)\ v \rightrightarrows e'[v'/x]}\;\beta \qquad \dfrac{}{(==_b)\ c_1 \rightrightarrows (==_{(c_1,b)})}\;\text{eq1} \qquad \dfrac{}{(==_{(c_1,b)})\ c_2 \rightrightarrows c_1 = c_2}\;\text{eq2}$$

$$\dfrac{e_l \rightrightarrows e_l' \quad e_r \rightrightarrows e_r' \quad e \rightrightarrows e'}{\mathsf{bEq}_b\ e_l\ e_r\ e \rightrightarrows \mathsf{bEq}_b\ e_l'\ e_r'\ e'}\;\text{beq} \qquad \dfrac{\tau_x \rightrightarrows \tau_x' \quad \tau \rightrightarrows \tau' \quad e_l \rightrightarrows e_l' \quad e_r \rightrightarrows e_r' \quad e \rightrightarrows e'}{\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \rightrightarrows \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ e'}\;\text{xeq}$$

$$\boxed{\tau \rightrightarrows \tau}$$

$$\dfrac{r \rightrightarrows r'}{\{x{:}b \mid r\} \rightrightarrows \{x{:}b \mid r'\}}\;\text{ref} \qquad \dfrac{\tau_x \rightrightarrows \tau_x' \quad \tau \rightrightarrows \tau'}{x{:}\tau_x \to \tau \rightrightarrows x{:}\tau_x' \to \tau'}\;\text{fun}$$

$$\dfrac{\tau \rightrightarrows \tau' \quad e_l \rightrightarrows e_l' \quad e_r \rightrightarrows e_r'}{\mathsf{PEq}_\tau\ \{e_l\}\ \{e_r\} \rightrightarrows \mathsf{PEq}_{\tau'}\ \{e_l'\}\ \{e_r'\}}\;\text{eq}$$

Fig. 3. Parallel reduction in terms and types.

- $\tau \doteq \mathsf{PEq}_{\tau'}\ \{e_l\}\ \{e_r\}$. By the IHs on τ', e_l, and e_r and eq. □

Lemma C.2 (Parallel reduction is substitutive). *If $e \rightrightarrows e'$, then:*

(1) If $e_1 \rightrightarrows e_2$, then $e_1[e/x] \rightrightarrows e_2[e'/x]$.
(2) If $\tau_1 \rightrightarrows \tau_2$, then $\tau_1[e/x] \rightrightarrows \tau_2[e'/x]$.

Proof. By mutual induction on e_1 and τ_1.

Expressions.

var $y \rightrightarrows y$. If $y \neq x$, then the substitution has no effect and the case is trivial. If $y = x$, then $x[e/x] = e$ and we have $e \rightrightarrows e'$ by assumption. We have $e \rightrightarrows e$ by reflexivity (Lemma C.1).

const $c \rightrightarrows c$. This case is trivial: the substitution has no effect.

lam $\lambda y{:}\tau.\ e' \rightrightarrows \lambda y{:}\tau.\ e''$. If $y \neq x$, then by the IH on e' and lam. If $y = x$, then the substitution has no effect and the case is trivial.

app $e_{11}\ e_{12} \rightrightarrows e_{21}\ e_{22}$, where $e1i \rightrightarrows e2i$ for $i = 1, 2$. By the IHs on e_{1i} and app.

beta $(\lambda y{:}\tau.\ e')\ v \rightrightarrows e'[v'/y]$, where $e' \rightrightarrows e''$ and $v \rightrightarrows v'$. If $y \neq x$, then $(\lambda y{:}\tau.\ e'[e/x])\ v[e/x] \rightrightarrows e''[e/x][v'[e/x]/y]$ by β. Since $y \neq x$, $e''[e/x][v'[e/x]/y] = e''[v'/y][e/x]$ as desired.
If $y = x$, then the substitution in the lambda has no effect, and we find $(\lambda x{:}\tau.\ e')\ v[e/x] \rightrightarrows e''[v'[e/x]/x]$ by β. We have $e''[v'[e/x]/x] = e''[v'/x][e/x]$ as desired.

eq1 $(==_b)\ c_1 \rightrightarrows (==_{(c_1,b)})$. This case is trivial by eq1, as the substitution has no effect.

eq2 $(==_{(c_1,b)})\ c_2 \rightrightarrows c_1 = c_2$. This case is trivial by eq2, as the substitution has no effect.

beq $\mathsf{bEq}_b\ e_l\ e_r\ e_p \rightrightarrows \mathsf{bEq}_b\ e_l'\ e_r'\ e_p'$, where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e_p \rightrightarrows e_p'$. By the IHs on e_l, e_r, and e_p and beq.

xeq $\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e_p \rightrightarrows \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l'\ e_r'\ e_p'$, where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e_p \rightrightarrows e_p'$. By the IHs on e_l, e_r, and e_p and xeq.

Types.

ref $\{y{:}b \mid r\} \rightrightarrows \{y{:}b \mid r'\}$ where $r \rightrightarrows r'$. If $y \neq x$, then $r[e/x] \rightrightarrows r'[e'/x]$ by the IH on r; we are done by ref.
If $y = x$, then the substitution has no effect, and the case is immediate by reflexivity (Lemma C.1).

fun $y{:}\tau_y \to \tau \rightrightarrows y{:}\tau_y' \to \tau'$ where $\tau_y \rightrightarrows \tau_y'$ and $\tau \rightrightarrows \tau'$. If $y \neq x$, then by the IH on τ_y and τ and fun.
If $y = x$, then the substitution only has effect in the domain. The IH on τ_y finds $\tau_y[e/x] \rightrightarrows \tau_y'[e'/x]$ in the domain; reflexivity covers the codomain (Lemma C.1), and we are done by fun.

eq $\mathsf{PEq}_{\tau}\ \{e_l\}\ \{e_r\} \rightrightarrows \mathsf{PEq}_{\tau'}\ \{e_l'\}\ \{e_r'\}$. By the IHs and eq. □

Corollary C.3 (Substituting multiple parallel reduction is parallel reduction). *If $e_1 \rightrightarrows^* e_2$, then $e[e_1/x] \rightrightarrows^* e[e2/x]$.*

Proof. First, notice that $e \rightrightarrows e$ by reflexivity (Lemma C.1). By induction on $e_1 \rightrightarrows^* e_2$, using reflexivity in the base case (Lemma C.1); the inductive step uses substituting parallel reduction (Lemma C.2) and the IH. □

Lemma C.4 (Parallel reduction subsumes reduction). *If $e_1 \hookrightarrow e_2$ then $e_1 \rightrightarrows e_2$.*

Proof. By induction on the evaluation derivation, using reflexivity of parallel reduction to cover expressions and types that didn't step (Lemma C.1).

ctx $\mathcal{E}[e] \hookrightarrow \mathcal{E}[e']$, where $e \hookrightarrow e'$. By the IH, $e \rightrightarrows e'$. By structural induction on $\mathcal{E}$.
- $\mathcal{E} \doteq \bullet$. By the outer IH.
- $\mathcal{E} \doteq \mathcal{E}_1\ e_2$. By the inner IH on $\mathcal{E}_1$, reflexivity on e_2, and app.
- $\mathcal{E} \doteq v_1\ \mathcal{E}_2$. By reflexivity on v_1, the inner IH on $\mathcal{E}_2$, and app.
- $\mathcal{E} \doteq \mathsf{bEq}_b\ e_l\ e_r\ \mathcal{E}'$. By reflexivity on e_l and e_r, the inner IH on and $\mathcal{E}'$, and beq.
- $\mathcal{E} \doteq \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ \mathcal{E}'$. By reflexivity on τ_x, τ, e_l and e_r, the inner IH on and $\mathcal{E}'$, and xeq.

β $(\lambda x{:}\tau.\ e)\ v \hookrightarrow e[v/x]$. By reflexivity (Lemma C.1, $e \rightrightarrows e$ and $v \rightrightarrows v$. By beta, $(\lambda x{:}\tau.\ e)\ v \rightrightarrows e[v/x]$.

eq1 By eq1.

eq2 By eq2. □

C.2 Forward Simulation

Lemma C.5 (Parallel reduction is a forward simulation). *If $e_1 \rightrightarrows e_2$ and $e_1 \hookrightarrow e_1'$, then there exists e_2' such that $e_2 \hookrightarrow^* e_2'$ and $e_1' \rightrightarrows e_2'$.*

Proof. By induction on the derivation of $e_1 \hookrightarrow e_1'$, leaving e_2 general.

ctx By structural induction on $\mathcal{E}$, using reflexivity (Lemma C.1) on parts where the IH doesn't apply.
- $\mathcal{E} \doteq \bullet$. By the outer IH on the actual step.
- $\mathcal{E} \doteq \mathcal{E}_1\ e_2$. By the IH on $\mathcal{E}_1$, reflexivity on e_2, and app.
- $\mathcal{E} \doteq v_1\ \mathcal{E}_2$. By reflexivity on v_1, the IH on $\mathcal{E}_2$, and app.
- $\mathcal{E} \doteq \mathsf{bEq}_b\ e_l\ e_r\ \mathcal{E}'$. By reflexivity on e_l and e_r, the IH on $\mathcal{E}'$, and beq.
- $\mathcal{E} \doteq \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ \mathcal{E}'$. By reflexivity on τ_x, τ, e_l and e_r, the IH on $\mathcal{E}'$, and xeq.

β $(\lambda x{:}\tau.\ e)\ v \hookrightarrow e[v/x]$. One of two rules could have applied to find $e_1 \rightrightarrows e_2$: app or β.

In the app case, we have $e_2 = (\lambda x{:}\tau'.\ e')\ v'$ where $\tau \Rrightarrow \tau'$ and $e \Rrightarrow e'$ and $v \Rrightarrow v'$. Let $e_2' = e'[v'/x]$. We find $e_2 \hookrightarrow^* e_2'$ in one step by β. We find $e[v/x] \Rrightarrow e'[v'/x]$ by substitutivity of parallel reduction (Lemma C.2).

In the β case, we have $e_2 = e'[v'/x]$ such that $e \Rrightarrow e'$ and $v \Rrightarrow v'$. Let $e_2' = e_2$. We find $e_2 \hookrightarrow^* e_2'$ in no steps at all; we find $e_1' \Rrightarrow e_2'$ by substitutivity of parallel reduction (Lemma C.2).

eq1 $(==_b)\ c_1 \hookrightarrow (==_{(c_1,b)})$. One of two rules could have applied to find $(==_b)\ c_1 \Rrightarrow e_2$: app or eq1.

In the app case, we must have $e_2 = e_1 = (==_b)\ c_1$, because there are no reductions available in these constants. Let $e_2' = (==_{(c_1,b)})$. We find $e_2 \hookrightarrow^* e_2'$ in a single step by our assumption (or eq1). We find parallel reduction by reflexivity (Lemma C.1).

In the eq2 case, we have $e_2 = e_1' = (==_{(c_1,b)})$. Let $e_2' = e_2$. We find $e_2 \hookrightarrow^* e_2'$ in no steps at all. We find parallel reduction by reflexivity (Lemma C.1).

eq2 $(==_{(c_1,b)})\ c_2 \hookrightarrow c_1 = c_2$. One of two rules could have applied to find $(==_{(c_1,b)})\ c_2 \Rrightarrow e_2$: app or eq2.

In the app case, we have $e_2 = e_1 = (==_{(c_1,b)})\ c_2$, because there are no reductions available in these constants. Let $e_2' \doteq c_1 = c_2$, i.e. `true` when $c_1 = c_2$ and `false` otherwise. We find $e_2 \hookrightarrow^* e_2'$ in a single step by our assumption (or eq2). We find parallel reduction by reflexivity (Lemma C.1).

In the eq2 case, we have $e_2 = e_1' \doteq c_1 = c_2$, i.e. `true` when $c_1 = c_2$ and `false` otherwise. Let $e_2' = e_2$. We find $e_2 \hookrightarrow^* e_2'$ in no steps at all. We find parallel reduction by reflexivity (Lemma C.1). □

C.3 Backward Simulation

Lemma C.6 (Reduction is substitutive). *If $e_1 \hookrightarrow e_2$, then $e_1[e/x] \hookrightarrow e_2[e/x]$.*

Proof. By induction on the derivation of $e_1 \hookrightarrow e_2$.

ctx By structural induction on $\mathcal{E}$.

- $\mathcal{E} \doteq \bullet$. By the outer IH.
- $\mathcal{E} \doteq \mathcal{E}_1\ e_2$. By the IH on $\mathcal{E}_1$ and ctx.
- $\mathcal{E} \doteq v_1\ \mathcal{E}_2$. By the IH on $\mathcal{E}_2$ and ctx.
- $\mathcal{E} \doteq \mathsf{bEq}_b\ e_l\ e_r\ \mathcal{E}'$. By the IH on $\mathcal{E}'$ and ctx.
- $\mathcal{E} \doteq \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ \mathcal{E}'$. By the IH on $\mathcal{E}'$ and ctx.

β $(\lambda y{:}\tau.\ e')\ v \hookrightarrow e'[v/y]$. We must show $(\lambda y{:}\tau.\ e')[e/x]\ v[e/x] \hookrightarrow e'[v/y][e/x]$.

The exact result depends on whether $y = x$. If $y \neq x$, the substitution goes through, and we have $(\lambda y{:}\tau.\ e')[e/x] = \lambda y{:}\tau[e/x].\ e'[e/x]$. By β, $(\lambda y{:}\tau[e/x].\ e'[e/x])\ v[e/x] \hookrightarrow e'[e/x][v[e/x]/y]$. But $e'[e/x][v[e/x]/y] = e'[v/y][e/x]$, and we are done.

If, on the other hand, $y = x$, then the substitution has no effect in the body of the lambda, and $(\lambda y{:}\tau.\ e')[e/x] = \lambda y{:}\tau[e/x].\ e'$. By β again, we find $(\lambda y{:}\tau[e/x].\ e')\ v[e/x] \hookrightarrow e'[v[e/x]/y]$. Since $y = x$, we really have $e'[v[e/x]/x]$ which is the same as $e'[v/x][e/x] = e'[v/y][e/x]$, as desired.

eq1 The substitution has no effect; immediate, by eq1.

eq2 The substitution has no effect; immediate, by eq2. □

Corollary C.7 (Multi-step reduction is substitutive). *If $e_1 \hookrightarrow^* e_2$, then $e_1[e/x] \hookrightarrow^* e_2[e/x]$.*

Proof. By induction on the derivation of $e_1 \hookrightarrow^* e_2$. The base case is immediate ($e_1 = e_2$, and we take no steps). The inductive case follows by the IH and single-step substitutivity (Lemma C.6). □

$$\frac{}{x \rightsquigarrow x}\ \text{var} \qquad \frac{}{c \rightsquigarrow c}\ \text{const} \qquad \frac{\tau \rightrightarrows \tau' \quad e \rightrightarrows e'}{\lambda x{:}\tau.\ e \rightsquigarrow \lambda x{:}\tau'.\ e'}\ \text{lam} \qquad \frac{e_1 \rightrightarrows e_1' \quad e_2 \rightrightarrows e_2'}{e_1\ e_2 \rightsquigarrow e_1'\ e_2'}\ \text{app}$$

$$\frac{e_l \rightrightarrows e_l' \quad e_r \rightrightarrows e_r' \quad e \rightrightarrows e'}{\mathsf{bEq}_b\ e_l\ e_r\ e \rightsquigarrow \mathsf{bEq}_b\ e_l'\ e_r'\ e'}\ \text{beq} \qquad \frac{\tau_x \rightrightarrows \tau_x' \quad \tau \rightrightarrows \tau' \quad e_l \rightrightarrows e_l' \quad e_r \rightrightarrows e_r' \quad e \rightrightarrows e'}{\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \rightsquigarrow \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ e'}\ \text{xeq}$$

Fig. 4. Term congruence.

We say terms are *congruent* when they (a) have the same outermost constructor and (b) their subparts parallel reduce to each other.[1] That is, $\rightsquigarrow \subseteq \rightrightarrows$, where the outermost rule must be one of var, const, lam, app, beq, or xeq and cannot be a *real* reduction like β, eq1, or eq2.

Congruence is a key tool in proving that parallel reduction is a backward simulation. Parallel reductions under a lambda prevent us from having an "on-the-nose" relation, but reduction can keep up enough with parallel reduction to maintain congruence.

Lemma C.8 (Congruence implies parallel reduction). *If* $e_1 \rightsquigarrow e_2$ *then* $e_1 \rightrightarrows e_2$.

Proof. By induction on the derivation of $e_1 \rightsquigarrow e_2$.

var $x \rightsquigarrow x$. By var.
const $c \rightsquigarrow c$. By const.
lam $\lambda x{:}\tau.\ e \rightsquigarrow \lambda x{:}\tau'.\ e'$, with $\tau \rightrightarrows \tau'$ and $e \rightrightarrows e'$. By lam.
app $e_1\ e_2 \rightsquigarrow e_1'\ e_2'$, with $e_1 \rightrightarrows e_1'$ and $e_2 \rightrightarrows e_2'$. By app.
beq $\mathsf{bEq}_b\ e_l\ e_r\ e \rightsquigarrow \mathsf{bEq}_b\ e_l'\ e_r'\ e$, with $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e \rightrightarrows e'$. By beq.
xeq By xeq. $\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \rightsquigarrow \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l'\ e_r'\ e$, with $\tau_x \rightrightarrows \tau_x'$ and $\tau \rightrightarrows \tau'$ and $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e \rightrightarrows e'$. By xeq. □

We need to strengthen substitutivity (Lemma C.2) to show that it preserves congruence.

Corollary C.9 (Congruence is substitutive). *If* $e_1 \rightsquigarrow e_1'$ *and* $e_2 \rightsquigarrow e_2'$, *then* $e_1[e_2/x] \rightsquigarrow e_2[e_2'/x]$.

Proof. By cases on e_1.

- $e_1 = y$. It must be that $e_2 = y$ as well, since only var could have applied. If $y \neq x$, then the substitution has no effect and we have $y \rightsquigarrow y$ by assumption (or var). If $x = y$, then $e_1[e_2/x] = e_2$ and we have $e_2 \rightsquigarrow e_2'$ by assumption.
- $e_1 = c$. It must be that $e_2 = c$ as well. The substitution has no effect; immediate by var.
- $e_1 = \lambda y{:}\tau.\ e$. It must be that $e_2 = \lambda y{:}\tau'.\ e'$ such that $\tau \rightrightarrows \tau'$ and $e \rightrightarrows e'$. If $y \neq x$, then we must show $\lambda y{:}\tau[e_2/x].\ e[e_2/x] \rightsquigarrow \lambda y{:}\tau'[e_2'/x].\ e'[e_2'/x]$, which we have immediately by lam and Lemma C.2 on our two subparts. If $y = x$, then we must show $\lambda y{:}\tau[e_2/x].\ e \rightsquigarrow \lambda y{:}\tau'[e_2'/x].\ e'$, which we have immediately by lam, Lemma C.2 on our $\tau \rightrightarrows \tau'$, and the fact that $e \rightrightarrows e'$.
- $e_1 = e_{11}\ e_{12}$. It must be that $e_2 = e_{21}\ e_{22}$, such that $e_{11} \rightrightarrows e_{21}$ and $e_{12} \rightrightarrows e_{22}$. By app and Lemma C.2 on the subparts.
- $e_1 = \mathsf{bEq}_b\ e_l\ e_r\ e$. It must be the case that $e_2 = \mathsf{bEq}_b\ e_l'\ e_r'\ e'$ where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$. By beq and Lemma C.2 on the subparts.
- $e_1 = \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e$. It must be the case that $e_2 = \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ e'$ where $e_l \rightrightarrows e_l'$ (and similarly for τ_x, τ, e_r, and e). By xeq and Lemma C.2 on the subparts. □

[1] Congruent terms are related to Takahashi's $\tilde{M}$ operator: in that they characterize parallel reductions that preserve structure. They are not the same, though: Takahashi's $\tilde{M}$ will do $\beta\eta$-reductions on outermost redexes.

Lemma C.10 (Parallel reduction of values implies congruence). *If $v_1 \rightrightarrows v_2$ then $v_1 \overset{\rightsquigarrow}{\rightsquigarrow} v_2$.*

Proof. By induction on the derivation of $v_1 \rightrightarrows v_2$.

var Contradictory: variables aren't values.

const Immediate, by const.

lam Immediate, by lam.

app Contradictory: applications aren't values.

beq Immediate, by beq.

xeq Immediate, by xeq.

β Contradictory: applications aren't values.

eq1 Contradictory: applications aren't values.

eq2 Contradictory: applications aren't values. □

Lemma C.11 (Parallel reduction implies reduction to congruent forms). *If $e_1 \rightrightarrows e_2$, then there exists e_1' $e_1 \hookrightarrow^* e_1'$ such that $e_1' \overset{\rightsquigarrow}{\rightsquigarrow} e_2$.*

Proof. By induction on $e_1 \rightrightarrows e_2$.

Structural rules.

var $x \rightrightarrows x$. We have $e_1 = e_2 = x$ by var.

const $c \rightrightarrows c$. We have $e_1 = e_2 = c$ by const.

lam $\lambda x{:}\tau.\ e \rightrightarrows \lambda x{:}\tau'.\ e'$, where $\tau \rightrightarrows \tau'$ and $e \rightrightarrows e'$. Immediate, by lam.

app $e_{11}\ e_{12} \rightrightarrows e_{21}\ e_{22}$, where $e_{11} \rightrightarrows e_{21}$ and $e_{12} \rightrightarrows e_{22}$. Immediate, by app.

beq $\mathsf{bEq}_b\ e_l\ e_r\ e \rightrightarrows \mathsf{bEq}_b\ e_l'\ e_r'\ e'$ where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e \rightrightarrows e'$. Immediate, by beq.

xeq $\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e \rightrightarrows \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ e'$ where $\tau_x \rightrightarrows \tau_x'$ and $\tau \rightrightarrows \tau'$ and $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e \rightrightarrows e'$. Immediate, by xeq.

Reduction rules. These are the more interesting cases, where the parallel reduction does a reduction step—ordinary reduction has to do more work to catch up.

β $(\lambda x{:}\tau.\ e)\ v \rightrightarrows e'[v'/x]$, where $e \rightrightarrows e''$ and $v \rightrightarrows v''$.
We have $(\lambda x{:}\tau.\ e)\ v \hookrightarrow e[v/x]$ by β. By the IH on $e \rightrightarrows e''$, there exists e' such that $e \hookrightarrow^* e'$ such that $e' \overset{\rightsquigarrow}{\rightsquigarrow} e''$. We *ignore* the IH on $v \rightrightarrows v''$, noticing instead that parallel reducing values are congruent (Lemma C.10) and so $v \overset{\rightsquigarrow}{\rightsquigarrow} v''$. Since reduction is substitutive (Corollary C.7), we can find that $e[v/x] \hookrightarrow^* e'[v/x]$. Since congruence is substitutive (Lemma C.9), we have $e'[v/x] \overset{\rightsquigarrow}{\rightsquigarrow} e''[v''/x]$, as desired.

eq1 $(==_b)\ c_1 \rightrightarrows (==_{(c_1,b)})$. We have $(==_b)\ c_1 \hookrightarrow (==_{(c_1,b)})$ in a single step; we find congruence by const.

eq2 $(==_{(c_1,b)})\ c_2 \rightrightarrows c_1 = c_2$. We have $(==_{(c_1,b)})\ c_2 \hookrightarrow c_1 = c_2$ in a single step; we find congruence by const. □

Lemma C.12 (Congruence to a value implies reduction to a value). *If $e \overset{\rightsquigarrow}{\rightsquigarrow} v'$ then $e \hookrightarrow^* v$ such that $v \overset{\rightsquigarrow}{\rightsquigarrow} v'$.*

Proof. By induction on v'.

- $v' \doteq c$. It must be the case that $e = c$. Let $v = c$. By const.
- $v' \doteq \lambda x{:}\tau'.\ e''$. It must be the case that $e = \lambda x{:}\tau.\ e'$ such that $\tau \rightrightarrows \tau'$ and $e \rightrightarrows e''$. By lam.
- $v \doteq \mathsf{bEq}_b\ e_l'\ e_r'\ v_p'$. It must be the case that $e = \mathsf{bEq}_b\ e_l\ e_r\ e_p$ where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e_p \rightrightarrows v_p'$. Since parallel reduction implies reduction to congruent forms (Lemma C.11), we have $e_p \hookrightarrow^* e_p'$ and $e_p' \overset{\rightsquigarrow}{\rightsquigarrow} v_p'$. By the IH on v_p', we know that $e_p' \hookrightarrow^* v_p$ such that $v_p \overset{\rightsquigarrow}{\rightsquigarrow} v_p'$.

By repeated use of ctx, we find $\mathsf{bEq}_b\ e_l\ e_r\ e_p \hookrightarrow^* \mathsf{bEq}_b\ e_l\ e_r\ v_p$. Since its proof part is a value, this term is a value. We find $\mathsf{bEq}_b\ e_l\ e_r\ v_p \stackrel{\rightsquigarrow}{\rightsquigarrow} \mathsf{bEq}_b\ e_l'\ e_r'\ v_p'$ by ebeq.

- $v \doteq \mathsf{xEq}_{x:\tau_x' \to \tau}\ e_l'\ e_r'\ v_p'$. It must be the case that $e = \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e_p$ where $\tau_x \rightrightarrows \tau_x'$ and $\tau \rightrightarrows \tau'$ and $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$ and $e_p \rightrightarrows v_p'$. Since parallel reduction implies reduction to congruent forms (Lemma C.11), we have $e_p \hookrightarrow^* e_p'$ and $e_p' \stackrel{\rightsquigarrow}{\rightsquigarrow} v_p'$. By the IH on v_p', we know that $e_p' \hookrightarrow^* v_p$ such that $v_p \stackrel{\rightsquigarrow}{\rightsquigarrow} v_p'$. By repeated application of ctx, we find $\mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e_p \hookrightarrow^* \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ v_p$. Since its proof part is a value, this term is a value. We find $\mathsf{xEq}_{\tau_x:\tau \to}\ e_l\ e_r\ v_p \stackrel{\rightsquigarrow}{\rightsquigarrow} \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ v_p'$ by exeq. □

Corollary C.13 (Parallel reduction to a value implies reduction to a related value). *If $e \rightrightarrows v'$ then there exists v such that $e \hookrightarrow^* v$ and $v \stackrel{\rightsquigarrow}{\rightsquigarrow} v'$.*

Proof. Since parallel reduction implies reduction to congruent forms (Lemma C.11), we have $e \hookrightarrow^* e'$ such that $e' \stackrel{\rightsquigarrow}{\rightsquigarrow} v'$. But congruence to a value implies reduction to a value (Lemma C.12), so $e' \hookrightarrow^* v$ such that $v \stackrel{\rightsquigarrow}{\rightsquigarrow} v'$. By transitivity of reduction, $e \hookrightarrow^* v$. □

Lemma C.14 (Congruence is a backward simulation). *If $e_1 \stackrel{\rightsquigarrow}{\rightsquigarrow} e_2$ and $e_2 \hookrightarrow e_2'$ then there exists e_1' where $e_1 \hookrightarrow^* e_1'$ such that $e_1' \stackrel{\rightsquigarrow}{\rightsquigarrow} e_2'$.*

Proof. By induction on the derivation of $e_2 \hookrightarrow e_2'$.

ctx $\mathcal{E}[e] \hookrightarrow \mathcal{E}[e']$, where $e \hookrightarrow e'$.

- $\mathcal{E} \doteq \bullet$. By the outer IH.
- $\mathcal{E} \doteq \mathcal{E}_1\ e_2$. It must be that $e_1 = e_{11}\ e_{12}$, where $e_{11} \rightrightarrows \mathcal{E}_1[e]$ and $e_{12} \rightrightarrows e_2$. By the IH on $\mathcal{E}_1$, finding evaluation with ctx and congruence with app.
- $\mathcal{E} \doteq v_1'\ \mathcal{E}_2$. It must be that $e_1 = e_{11}\ e_{12}$, where $e_{11} \rightrightarrows v_1'$ and $e_{12} \rightrightarrows \mathcal{E}_2[e_2]$. We find that $e_{11} \hookrightarrow^* v_1$ such that $v_1 \stackrel{\rightsquigarrow}{\rightsquigarrow} v_1'$ by Corollary C.13. By the IH on $\mathcal{E}_2$ and evaluation with ctx and congruence with app.
- $\mathcal{E} \doteq \mathsf{bEq}_b\ e_l'\ e_r'\ \mathcal{E}'$. It must be the case that $e_1 = \mathsf{bEq}_b\ e_l\ e_r\ e_p$ where $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$. By the IH on $\mathcal{E}'$; we find the evaluation with ctx and congruence with beq.
- $\mathcal{E} \doteq \mathsf{xEq}_{x:\tau_x' \to \tau'}\ e_l'\ e_r'\ \mathcal{E}'$. It must be the case that $e_1 = \mathsf{xEq}_{x:\tau_x \to \tau}\ e_l\ e_r\ e_p$ such that $\tau_x \rightrightarrows \tau_x'$ and $\tau \rightrightarrows \tau'$ and $e_l \rightrightarrows e_l'$ and $e_r \rightrightarrows e_r'$. By the IH on $\mathcal{E}'$; we find the evaluation with ctx and congruence with xeq.

β $(\lambda x{:}\tau'.\ e')\ v' \hookrightarrow e'[v'/x]$. Congruence implies that $e_1 = e_{11}\ e_{12}$ such that $e_{11} \rightrightarrows \lambda x{:}\tau'.\ e'$ and $e_{12} \rightrightarrows v'$. Parallel reduction to a value implies reduction to a congruent value (Corollary C.13), $e_{11} \hookrightarrow^* v_{11}$ such that $v_{11}' \stackrel{\rightsquigarrow}{\rightsquigarrow} \lambda x{:}\tau'.\ e'$, i.e., $v_{11} = \lambda x{:}\tau.\ e$ such that $\tau \rightrightarrows \tau'$ and $e \rightrightarrows e'$. Similarly, $e_{12} \hookrightarrow^* v$ such that $v \stackrel{\rightsquigarrow}{\rightsquigarrow} v'$.

By β, we find $(\lambda x{:}\tau.\ e)\ v \hookrightarrow^* e'[v/x]$; by transitivity of reduction, we have $e_1 = e_{11}\ e_{12} \hookrightarrow^* e'[v/x]$. Since congruence is substitutive (Corollary C.9), we have $e[v/x] \stackrel{\rightsquigarrow}{\rightsquigarrow} e'[v'/x]$.

eq1 $(==_b)\ c_1 \hookrightarrow (==_{(c_1,b)})$. Congruence implies that $e_1 = e_{11}\ e_{12}$ such that $e_{11} \rightrightarrows (==_b)$ and $e_{12} \rightrightarrows c_1$. Parallel reduction to a value implies reduction to a related value (Corollary C.13), $e_{11} \hookrightarrow^* v_{11}$ such that $v_{11} \stackrel{\rightsquigarrow}{\rightsquigarrow} (==_b)$ (and similarly for e_{12} and c_1). But the each constant is congruent only to itself, so $v_{11} = (==_b)$ and $v_{12} = c_1$. We have $(==_b)\ c_1 \hookrightarrow (==_{(c_1,b)})$ by assumption. So $e_1 = e_{11}\ e_{12} \hookrightarrow^* (==_{(c_1,b)})$ by transitivity, and we have congruence by const.

eq2 $(==_{(c_1,b)})\ c_2 \hookrightarrow c_1 = c_2$. Congruence implies that $e_1 = e_{11}\ e_{12}$ such that $e_{11} \rightrightarrows (==_{(c_1,b)})\ c_2$ and $e_{12} \rightrightarrows c_2$. Parallel reduction to a value implies reduction to a related value (Corollary C.13), $e_{11} \hookrightarrow^* v_{11}$ such that $v_{11} \rightrightarrows (==_{(c_1,b)})\ c_2$ (and similarly for e_{12} and c_2). But the each constant is congruent only to itself, so $v_{11} = (==_{(c_1,b)})\ c_2$ and $v_{12} = c_2$. We have $(==_{(c_1,b)})\ c_2 \hookrightarrow c_1 = c_2$ already, by assumption. So $e_1 = e_{11}\ e_{12} \hookrightarrow^* c_1 = c_2$ by transitivity, and we have congruence by const. □

Corollary C.15 (Parallel reduction is a backward simulation). *If $e_1 \rightrightarrows e_2$ and $e_2 \hookrightarrow e_2'$, then there exists e_1' such that $e_1 \hookrightarrow^* e_1'$ and $e_1' \rightrightarrows e_2'$.*

Proof. Parallel reduction implies reduction to congruent forms, so $e_1 \hookrightarrow^* e_1'$ such that $e_1' \mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}} e_2$. But congruence is a backward simulation (Lemma C.14), so $e_1' \hookrightarrow^* e_1''$ such that $e_1'' \mathrel{\substack{\rightsquigarrow\\\rightsquigarrow}} e_2'$. By transitivity of evaluation, $e_1 \hookrightarrow^* e_1''$. Finally, congruence implies parallel reduction (Lemma C.8), so $e_1'' \rightrightarrows e_2'$, as desired. □

C.4 Cotermination

Theorem C.16 (Cotermination at constants). *If $e_1 \rightrightarrows e_2$ then $e_1 \hookrightarrow^* c$ iff $e_2 \hookrightarrow^* c$.*

Proof. By induction on the evaluation steps taken, using direct reduction in the base case (Corollary C.13) and using parallel reduction as a forward and backward simulation (Lemmas C.5 and Corollary C.15) in the inductive case. □

Corollary C.17 (Cotermination at constants (multiple parallel steps)). *If $e_1 \rightrightarrows^* e_2$ then $e_1 \hookrightarrow^* c$ iff $e_2 \hookrightarrow^* c$.*

Proof. By induction on the parallel reduction derivation. The base case is immediate ($e_1 = e_2$); the inductive case follows from cotermination at constants (Theorem C.16) and the IH. □

REFERENCES

Masako Takahashi. 1989. Parallel Reductions in lambda-Calculus. *J. Symb. Comput.* 7, 2 (1989), 113–123. https://doi.org/10.1016/S0747-7171(89)80045-8