Refinement Types For Haskell

In this document we provide the proofs for the Theorems and Lemmata of the paper:

Refer to § A for proofs on dynamic semantics of $\lambda_\downarrow$ (Section 3 of the paper); to § B for proofs on static semantics of $\lambda_\downarrow$ with the Termination Oracle (Section 4 of the paper); and to § C for proofs on Termination Analysis of $\lambda_\downarrow$ (Section 4 of the paper);

The technical report of the paper is available on http://goto.ucsd.edu/ nvazou/lazytechreport.pdf

A. Language

We prove Lemma 1(Optimistic Reduction) in (Lemma 1) and Theorem 1(Optimistic Equivalence) in (Theorem 1).

Lemma 1 (Optimistic Reduction)**.** *if* $e \hookrightarrow_n^j v$ *then* $e \hookrightarrow_o^* v$.

Proof. We define a new version of optimistic semantics $\hookrightarrow_{o'}$ in Figure 1. We define the semantics for core λ-calculus (ignoring constants, let and fix operators), but its extension to $\lambda_\downarrow$ is straightforward. Intuitively, when an argument to the application is trivial, its evaluation is fired and then ignored. Obviously

$$\textbf{\textit{Expressions}} \quad e ::= x \mid \lambda x.e \mid e\, e \mid < e, e >$$
$$\textbf{\textit{Values}} \quad e ::= \lambda x.e$$

$$\begin{array}{lrll}
(\lambda x.e) & e_x \hookrightarrow_{o'} & e\,[e_x/x] & \text{if } \neg(e_x \hookrightarrow_{o'}^* v) \\
(\lambda x.e) & e_x \hookrightarrow_{o'} & \lambda x.e\ < e_x, e_x > & \text{if } (e_x \hookrightarrow_{o'}^* v) \\
(\lambda x.e) & < e_x, v > \hookrightarrow_{o'} & e\,[e_x/x] & \\
(\lambda x.e) & < e_x, e_y > \hookrightarrow_{o'} & (\lambda x.e)\ < e_x, e_y' > & \text{if } e_y \hookrightarrow_{o'} e_y' \\
e_1 & e_2 \hookrightarrow_{o'} & e_1'\ e_2 & \text{if } e_1 \hookrightarrow_{o'} e_1'
\end{array}$$

Figure 1. Operational Semantics of OPT'

$$e \hookrightarrow_n^* v \Leftrightarrow e \hookrightarrow_{o'}^* v\ (a)$$

The $\Leftarrow$ direction follows from the fact that if we have the evaluation path of $e \hookrightarrow_{o'}^* v$ then we can ignore the sub-paths between the second and third rule and get its respective $e \hookrightarrow_n^* v$ path.

The $\Rightarrow$ direction follows from the fact that if we have the evaluation path of e at every application $(\lambda x.e)\ e_x$ and if $e_x \hookrightarrow_{o'}^* v$ we insert the respective path. This way, we get the path $e \hookrightarrow_{o'}^* v$.

Now, we can prove that

$$e \hookrightarrow_{o'}^l v \Rightarrow e \hookrightarrow_o^* v$$

by induction on l.

Consider the path $e \hookrightarrow_{o'}^l v$. The first place where the evaluations differ is the one with $e_1 \equiv (\lambda x.e'')\ e_x$ and $e_x \hookrightarrow_{o'}^* v$ (equivalently from (a), e_x is trivial). Both evaluations will evaluate e_x and for some $l' < l$ by IH

$$e_x \hookrightarrow_{o'}^{l'} v_x \Rightarrow e_x \hookrightarrow_o^* v_x$$

Finally, since by Church Rosser, (if $e_x \hookrightarrow^\star v_x$ and $e''\,[e_x/x] \hookrightarrow^\star v$ then $e''\,[v_x/x] \hookrightarrow^\star v$) we get $e \hookrightarrow_o^* v$. □

$$\begin{array}{rrcl}
\textbf{\textit{Value Substitution}} & \rho & ::= & [] \mid [v/x]\,; \rho \\
\textbf{\textit{Expression Substitution}} & \theta & ::= & [] \mid [e/x]\,; \theta
\end{array}$$

Figure 2. Substitutions

Theorem 1 (Optimistic Equivalence)**.** $e \hookrightarrow_n^* v \Leftrightarrow e \hookrightarrow_o^* v$.

Proof. The $\Leftarrow$ direction follows from the fact that if a term reduces to a value under some evaluation strategy, then it reduces to that value under CBN. The $\Rightarrow$ direction follows from Lemma 1. □

B. Soundness With a Termination Oracle

We *assume* the Termination Oracle Hypothesis to prove Lemma 2 (Value Substitution) (follows from Lemma 4 with $\rho := [v/x]$, $\Gamma := x : \tau_x$, and $\Gamma' := \Gamma$) Lemma 3 (Serious Substitution) (follows from Lemma 3 with $\Gamma_1 = \emptyset$) Preservation (Theorem 2), and Progress (Theorem 3) We start by defining constants:

Definition 1 (Constants)**.** *Each constant c has a type* $\mathsf{Ty}(c)$ *such that*

- $\emptyset \vdash c : \mathsf{Ty}(c)$
- *If* $\mathsf{Ty}(c) \equiv \{v{:}b^l \mid e\}$*, then* $e \equiv v = c$
- *If* $\mathsf{Ty}(c)$ *is* $x{:}\tau_1 \to \tau_2$ *then for all values v such that* $\emptyset \vdash v : \tau_1$, $[|c|](v)$ *is defined and* $\emptyset \vdash [|c|](v) : \tau_2\,[v/x]$ *(so, it is not equal to* `crash`*).*
- *If* $\mathsf{Ty}(c)$ *is* $\forall\alpha.\sigma$ *then for all types τ such that* $\emptyset \vdash \tau$ $[|c|](\tau)$ *is defined and* $\emptyset \vdash [|c|](\tau) : \sigma\,[\tau/\alpha]$.
- $\models c : \mathsf{Ty}(c)$

Substitutions

Next, we formally define substitutions:

$$\boxed{\Gamma \models \rho}$$

$$\frac{}{\emptyset \models \emptyset}\ \textsc{WS-Empty} \qquad \frac{\Gamma \models \rho \qquad \emptyset \vdash v : \rho\sigma}{\Gamma, x : \sigma \models \rho; [v/x]}\ \textsc{WS-Ext}$$

Lemma 2 (Serious Narrowing)**.**

$$\Gamma = \Gamma_1; \Gamma_2$$
$$\tau_x \textit{ is serious}$$
$$\Gamma' = \Gamma_1; x : \tau_x; \Gamma_2$$
$$x \notin \mathit{freeVars}(e)$$

1. *If* $\Gamma' \vdash \sigma$ *then* $\Gamma \vdash \sigma$.
2. *If* $\Gamma' \vdash \sigma \preceq \sigma'$ *then* $\Gamma \vdash \sigma \preceq \sigma'$.
3. *If* $\Gamma' \vdash e : \sigma$ *then* $\Gamma \vdash e : \sigma$.

Proof. 1 By induction on the derivation $\Gamma' \vdash \sigma$:

- Case WF-BASE-↓. Assume:

$$\Gamma' \vdash \{v{:}b^{\downarrow} \mid e\}$$

By inversion:

$$\Gamma_T = \mathsf{Trivial}(\Gamma', v : b^{\downarrow})\ (a), \quad \Gamma_T \vdash e : \texttt{bool}\ (b)$$

By the definition of Trivial, we have

$$\Gamma'_T = \mathsf{Trivial}(\Gamma, v : b)\ (c)$$

So, from (c) and (b):

$$\Gamma \vdash \{v{:}b^{\downarrow} \mid e\}$$

- Cases WF-BASE-↑ and WF-VAR are trivial, as they do not depend on the environment.
- Case WF-FUN. Assume:

$$\Gamma' \vdash x{:}\tau \to \tau'$$

By inversion:

$$\Gamma' \vdash \tau, \qquad \Gamma', x : \tau \vdash \tau'$$

By IH,

$$\Gamma \vdash \tau, \qquad \Gamma, x : \tau \vdash \tau'$$

Using rule WF-FUN:

$$\Gamma \vdash x{:}\tau \to \tau'$$

- Case WF-POLY. Assume:

$$\Gamma' \vdash \forall\alpha.\sigma$$

By inversion:

$$\Gamma' \vdash \sigma$$

By IH

$$\Gamma \vdash \sigma$$

By rule WF-POLY

$$\Gamma \vdash \forall\alpha.\sigma$$

2 By induction on the derivation of $\Gamma' \vdash \sigma \preceq \sigma'$:

- Case $\preceq$-BASE-↓. Assume:

$$\Gamma' \vdash \{v{:}b^{\downarrow} \mid e_1\} \preceq \{v{:}b^{\downarrow} \mid e_2\}$$

By inversion

$$\mathsf{SmtValid}([|\Gamma'|] \Rightarrow [|e_1|] \Rightarrow [|e_2|])$$

By the definition of $[| * |]$

$$[|\Gamma|] = [|\Gamma'|]$$

So,

$$\mathsf{SmtValid}([|\Gamma|] \Rightarrow [|e_1|] \Rightarrow [|e_2|])$$

By rule $\preceq$-BASE-↓

$$\Gamma \vdash \{v{:}b^{\downarrow} \mid e_1\} \preceq \{v{:}b^{\downarrow} \mid e_2\}$$

- Cases $\preceq$-BASE-↑ and $\preceq$-VAR are trivial, as they do not depend on the environment.
- Case $\preceq$-FUN. Assume:

$$\Gamma' \vdash x{:}\tau_1 \to \tau'_1 \preceq x{:}\tau_2 \to \tau'_2$$

By inversion:

$$\Gamma' \vdash \tau_2 \preceq \tau_1, \qquad \Gamma', x : \tau_2 \vdash \tau'_1 \preceq \tau'_2$$

By IH

$$\Gamma \vdash \tau_2 \preceq \tau_1, \qquad \Gamma, x : \tau_2 \vdash \tau'_1 \preceq \tau'_2$$

By rule $\preceq$-FUN:

$$\Gamma \vdash x{:}\tau_1 \to \tau'_1 \preceq x{:}\tau_2 \to \tau'_2$$

- Case $\preceq$-POLY. Assume:

$$\Gamma' \vdash \forall\alpha.\sigma_1 \preceq \forall\alpha.\sigma_2$$

By inversion:

$$\Gamma' \vdash \sigma_1 \preceq \sigma_2$$

By IH

$$\Gamma \vdash \sigma_1 \preceq \sigma_2$$

By rule $\preceq$-POLY:

$$\Gamma \vdash \forall\alpha.\sigma_1 \preceq \forall\alpha.\sigma_2$$

3 By induction on the derivation $\Gamma' \vdash e : \sigma$. We push the rule T-SUB down in the tree:

- Case T-SUB. Assume:

$$\Gamma' \vdash e : \sigma$$

By inversion:

$$\Gamma' \vdash e : \sigma', \quad \Gamma' \vdash \sigma' \preceq \sigma, \quad \Gamma' \vdash \sigma$$

for some σ'. By IH, 2 and 1

$$\Gamma \vdash e : \sigma', \quad \Gamma \vdash \sigma' \preceq \sigma, \quad \Gamma \vdash \sigma$$

By rule T-SUB:

$$\Gamma \vdash e : \sigma$$

- Case $e \equiv y$. Assume:

$$\Gamma' \vdash e : \sigma$$

Since $x \notin freeVars(e)$ $x \neq y$, we have $\Gamma(y) = \Gamma'(y)$. Hence, for either rule T-VAR-BASE or T-VAR we get

$$\Gamma \vdash e : \sigma$$

- Case $e \equiv c$. Trivial.
- Case $e \equiv \lambda y.e'$. Assume:

$$\Gamma' \vdash (\lambda y.e') : y{:}\tau_y \to \tau$$

For some τ_y, τ. By inversion

$$\Gamma', y : \tau_y \vdash e' : \tau, \quad \Gamma' \vdash y{:}\tau_y \to \tau$$

By IH and 1

$$\Gamma, y : \tau_y \vdash e' : \tau, \quad \Gamma \vdash y{:}\tau_y \to \tau$$

By rule T-FUN

$$\Gamma \vdash (\lambda y.e') : y{:}\tau_y \to \tau$$

- Case $e \equiv e_1\ e_2$. Assume:

$$\Gamma' \vdash e_1\ e_2 : \tau\,[e_2/x]$$

and by inversion

$$\Gamma' \vdash e_2 : \tau_y, \qquad \Gamma' \vdash e_1 : (y{:}\tau_y \to \tau)$$

By IH

$$\Gamma \vdash e_1 : (y{:}\tau_y \to \tau), \qquad \Gamma \vdash e_2 : \tau_y$$

By rule T-APP

$$\Gamma \vdash e_1\ e_2 : \tau[e_2/y]$$

- Case $e \equiv \texttt{let}\ y = e_y\ \texttt{in}\ e'$. Assume:

$$\Gamma' \vdash \texttt{let}\ y = e_x\ \texttt{in}\ e' : \tau$$

By inversion

$$\Gamma' \vdash e_y : \tau_y, \Gamma', y : \tau_y \vdash e' : \tau, \Gamma' \vdash \tau$$

By IH and 1

$$\Gamma \vdash e_y : \tau_y, \Gamma, y : \tau_y \vdash e' : \tau, \Gamma \vdash \tau$$

By rule T-LET

$$\Gamma \vdash \texttt{let}\ y = e_x\ \texttt{in}\ e' : \tau$$

- Case $e \equiv \mu f.\lambda y.e'$. Assume:

$$\Gamma' \vdash \mu f.\lambda y.e : y{:}\tau_y \rightarrow \tau$$

By inversion

$$\Gamma', y : \tau_y, f : \tau \vdash e : \tau, \qquad \Gamma \vdash y{:}\tau_y \rightarrow \tau$$

By IH and 1

$$\Gamma', y : \tau_y, f : \tau \vdash e : \tau, \qquad \Gamma' \vdash y{:}\tau_y \rightarrow \tau$$

By rule T-REC

$$\Gamma \vdash \mu f.\lambda y.e : \tau$$

- Case $e \equiv [\Lambda\alpha]\, e'$. Assume:

$$\Gamma' \vdash [\Lambda\alpha]\, e' : \forall\alpha.\sigma'$$

By inversion

$$\Gamma' \vdash e' : \sigma'$$

By IH

$$\Gamma \vdash e' : \sigma'$$

By rule T-GEN

$$\Gamma \vdash [\Lambda\alpha]\, e' : \forall\alpha.\sigma'$$

- Case $e \equiv e'\,[\tau]$. Assume:

$$\Gamma' \vdash e'\,[\tau] : \sigma\,[\tau/\alpha]$$

By inversion

$$\Gamma' \vdash e' : \forall\alpha.\sigma, \Gamma' \vdash \tau, \tau \text{ is } \mathit{trivial}$$

By IH and 1

$$\Gamma \vdash e' : \forall\alpha.\sigma, \Gamma \vdash \tau, \tau \text{ is } \mathit{trivial}$$

By rule T-INST

$$\Gamma \vdash e'\,[\tau] : \sigma\,[\tau/\alpha]$$

□

Lemma 3 (Serious Substitution)**.** *Let*

$$\tau_x \textit{ is serious}$$

$$\Gamma' = \Gamma_1; x : \tau_x; \Gamma_2$$

$$\Gamma = \Gamma_1; \Gamma_2$$

If $\Gamma' \vdash e_1 : \sigma$ *and* $\Gamma \vdash e_2 : \tau_x$ *then* $\Gamma \vdash e_1\,[e_2/x] : \sigma$.

Proof. We split cases on the rule used at the root of the derivation. At each case we assume that the rule T-SUB is pushed down.

- Case T-SUB. Assume

$$\Gamma' \vdash e_1 : \sigma$$

By inversion

$$\Gamma' \vdash e_1 : \sigma_1, \quad \Gamma' \vdash \sigma_1 \preceq \sigma, \quad \Gamma' \vdash \sigma$$

By IH and Lemma 2

$$\Gamma \vdash e_1\,[e_2/x] : \sigma_1, \quad \Gamma \vdash \sigma_1 \preceq \sigma, \quad \Gamma \vdash \sigma$$

By rule T-SUB

$$\Gamma \vdash e_1\,[e_2/x] : \sigma$$

- Case T-VAR-BASE. Assume

$$\Gamma' \vdash y : \{v{:}b^{\downarrow} \mid v = y\}\ (a)$$

where $e_1 \equiv y$ and $\sigma \equiv \{v{:}b^{\downarrow} \mid v = y\}$. By inversion

$$\Gamma'(y) = \{v{:}b^{\downarrow} \mid e_y\}$$

There are two cases. Either $x = y$ or $x \neq y$. Since y has a trivial type, $x \neq y$. So, $x \notin \mathit{freeVars}(e_1)$. Hence $e_y\,[e_1/x] = e_y$ and from Lemma 2 and (a)

$$\Gamma \vdash e_1 : \sigma$$

- Case T-VAR. Assume

$$\Gamma' \vdash y : \Gamma'(y)\ (a)$$

where $e_1 \equiv y$ and $\sigma \equiv \Gamma'(y)$. Say $x = y$, then $\sigma \equiv \tau_x$ and $e_1\,[e_2/x] = e_2$, so

$$\Gamma \vdash e_1\,[e_2/x] : \sigma$$

Otherwise, $x \neq y$, so $x \notin \mathit{freeVars}(e_1)$. Hence $e_1\,[e_1/x] = e_1$ and from Lemma 2 and (a)

$$\Gamma \vdash e_1 : \sigma$$

- Case T-CON. Assume

$$\Gamma' \vdash c : \mathsf{Ty}(c)\ (a)$$

Then $x \notin \mathit{freeVars}(e_1)$. Hence $e_1\,[e_2/x] = e_1$ and from Lemma 2 and (a)

$$\Gamma \vdash e_1 : \sigma$$

- Case T-FUN. Assume

$$\Gamma' \vdash (\lambda y.e) : y{:}\tau_y \rightarrow \tau$$

where $e_1 \equiv \lambda y.e$ and $\sigma \equiv y{:}\tau_y \rightarrow \tau$. By inversion

$$\Gamma', y : \tau_y \vdash e : \tau, \quad \Gamma' \vdash y{:}\tau_y \rightarrow \tau$$

By IH and Lemma 2

$$\Gamma, y : \tau_y \vdash e\,[e_2/x] : \tau, \quad \Gamma \vdash y{:}\tau_y \rightarrow \tau$$

By rule T-FUN

$$\Gamma \vdash (\lambda y.e\,[e_2/x]) : y{:}\tau_y \rightarrow \tau$$

But $(\lambda y.e)\,[e_2/x] = e_1\,[e_2/x]$, since by α-renaming x should be different than y.

- Case T-APP. Assume

$$\Gamma' \vdash e'_1\ e'_2 : \tau\,[e'_2/y]$$

By inversion

$$\Gamma' \vdash e'_1 : (y{:}\tau_y \rightarrow \tau)\ (a), \quad \Gamma' \vdash e'_2 : \tau_y\ (b)$$

By IH

$$\Gamma \vdash e'_1\,[e_2/x] : (y{:}\tau_y \rightarrow \tau), \quad \Gamma \vdash e'_2\,[e_2/x] : \tau_y$$

By which and rule T-APP and since $(e'_1\,[e_2/x])\ (e'_2\,[e_2/x]) = (e'_1\ e'_2)\,[e_2/x]$

$$\Gamma \vdash e\,[e_2/x] : \tau\,[e'_2/y]$$

- Case T-LET. Assume

$$\Gamma' \vdash \texttt{let}\ y = e_y\ \texttt{in}\ e : \tau$$

By inversion

$$\Gamma' \vdash e_y : \tau_y, \quad \Gamma', y : \tau_y \vdash e : \tau, \quad \Gamma' \vdash \tau$$

By IH and Lemma 2

$$\Gamma \vdash e_y\,[e_2/x] : \tau_y, \quad \Gamma, y : \tau_y \vdash e\,[e_2/x] : \tau, \quad \Gamma \vdash \tau$$

By rule T-LET and since $\texttt{let}\ y = e_y\,[e_2/x]\ \texttt{in}\ e\,[e_2/x] = (\texttt{let}\ y = e_y\ \texttt{in}\ e)\,[e_2/x]$

$$\Gamma \vdash (\texttt{let}\ y = e_y\ \texttt{in}\ e)\,[e_2/x] : \tau$$

- Case T-REC. Assume

$$\Gamma' \vdash \mu f.\lambda y.e : y{:}\tau_x \to \tau$$

By inversion

$$\Gamma', y : \tau_y, f : \tau \vdash e : \tau$$

By IH

$$\Gamma, y : \tau_y, f : \tau \vdash e\,[e_2/x] : \tau$$

By rule T-REC and since $\mu f.\lambda y.(e\,[e_2/x]) = (\mu f.\lambda y.e)\,[e_2/x]$

$$\Gamma \vdash (\mu f.\lambda y.e)\,[e_2/x] : y{:}\tau_y \to \tau$$

- Case T-GEN. Assume

$$\Gamma' \vdash [\Lambda\alpha]\,e : \forall\alpha.\sigma$$

By inversion

$$\Gamma' \vdash e : \sigma$$

By IH

$$\Gamma \vdash e\,[e_2/x] : \sigma$$

By rule T-GEN and since $[\Lambda\alpha]\,e\,[e_2/x] = ([\Lambda\alpha]\,e)\,[e_2/x]$

$$\Gamma \vdash ([\Lambda\alpha]\,e)\,[e_2/x] : \forall\alpha.\sigma$$

- Case T-INST. Assume

$$\Gamma' \vdash e\,[\tau] : \sigma\,[\tau/\alpha]$$

By inversion

$$\Gamma' \vdash e : \forall\alpha.\sigma, \quad \Gamma' \vdash \tau, \quad \tau \text{ is trivial}$$

By IH and Lemma 2

$$\Gamma \vdash e\,[e_2/x] : \forall\alpha.\sigma, \quad \Gamma \vdash \tau$$

By rule T-INST and since $e\,[e_2/x]\,[\tau] = (e\,[\tau])\,[e_2/x]$

$$\Gamma \vdash (e\,[\tau])\,[e_2/x] : \sigma\,[\tau/\alpha]$$

□

Lemma 4 (Value Substitution). *If $\Gamma \models \rho$ then*

1. *If $\Gamma; \Gamma' \vdash e : \sigma$ then $\rho\Gamma' \vdash \rho e : \rho\sigma$.*
2. *If $\Gamma; \Gamma' \vdash \sigma \preceq \sigma'$ then $\rho\Gamma' \vdash \rho\sigma \preceq \rho\sigma'$.*

Proof. We use the respective Lemma on standard refinement types of λ_L [1]. For the first case, consider the derivation trees T_e and T_{x_i} (for every $[v_i/x_i] \in \Gamma$) of $x : \tau_x; \Gamma \vdash e : \sigma$ and $\Gamma \vdash v_i : \tau_{x_i}$, respectively. We map T_e and T_{x_i} to T_e^L and $T_{x_i}^L$ using the following transformation: we delete the labels from types and replace the rule T-VAR on variables, with the rules T-VAR-BASE and T-SUB of λ_L. T_e^L and $T_{x_i}^L$ are valid derivation trees for λ_L. Using the Lemma on λ_L we get a derivation tree $T_{\rho e}^L$ whose structure is the same as T_e^L where leaves typing x_is have been replaced with $T_{x_i}^L$. So, we can invert the transformation on $T_{\rho e}^L$ to get $T_{\rho e}$, a derivation tree of $\rho\Gamma \vdash \rho e : \rho\sigma$ in $\lambda_\downarrow$. □

Theorem 2 (Preservation). *If $\emptyset \vdash e : \sigma$ and $e \hookrightarrow_o e'$, then $\emptyset \vdash e' : \sigma$.*

Proof. By induction on the typing derivation $\emptyset \vdash e : \sigma$. We split cases on the rule used at the root of the derivation; at each case we push the rule T-SUB down in the tree.

- Case T-SUB. Assume

$$\emptyset \vdash e : \sigma$$

By inversion

$$\emptyset \vdash e : \sigma_1\ (a), \quad \emptyset \vdash \sigma_1 \preceq \sigma\ (b), \quad \emptyset \vdash \sigma\ (c)$$

for some σ_1. By IH and (a)

$$\emptyset \vdash e' : \sigma_1\ (a')$$

By (a'), (b) and (c) if we apply the rule T-SUB

$$\emptyset \vdash e' : \sigma$$

- Cases T-VAR-BASE, T-VAR, T-CON, T-FUN, T-REC, T-GEN are trivial, since there is no e' such that $e \hookrightarrow_o e'$.
- Case T-APP. Assume

$$\emptyset \vdash e_1\ e_2 : \tau[e_2/x]$$

where $e \equiv e_1\ e_2$ and $\sigma \equiv \tau\,[e_2/x]$. By inversion we have

$$\emptyset \vdash e_1 : (x{:}\tau_x \to \tau)\ (a), \quad \emptyset \vdash e_2 : \tau_x\ (b)$$

We split cases on the structure of e
  - $e \equiv e_1\ e_2$ and e_1 is not a value. By (a) and IH there exists an e_1' so that $e_1 \hookrightarrow_o e_1'$ and

$$\emptyset \vdash e_1' : (x{:}\tau_x \to \tau)$$

  Also, $e' \equiv e_1'\ e_2$. By (b) and T-APP

$$\emptyset \vdash e' : \sigma$$

  - $e \equiv v\ e_2$ and e_2 is not a value and trivial. By (b) and IH there exists an e_2' such that $e_2 \hookrightarrow_o e_2'$ and

$$\emptyset \vdash e_2' : \tau_x$$

  Then $e' \equiv v\ e_2'$. So, by rule T-APP:

$$\emptyset \vdash e' : \tau\,[e_2/x]$$

  - $e \equiv c\ e_2$. If e_2 is not a value, then by IH on (b), there exists e_2 such that $e_2 \hookrightarrow_o e_2'$, and

$$\emptyset \vdash e_2' : \tau_x\ (a')$$

  Then $e' \equiv c\ e_2'$. By (a) and (b') via rule T-APP, we get $\emptyset \vdash e' : \tau\,[e_2/x]$.
  Otherwise, there exists a value v, such that $e_2 \equiv v$ and $e' \equiv [|c|](v)$. From (a), (b) and Definition 1

$$\emptyset \vdash e' : \tau\,[e_2/x]$$

  - $e \equiv \lambda x.e_{11}\ e_2$. from (a) and by inversion of rule T-FUN we have

$$x : \tau_x \vdash e_{11} : \tau\ (c)$$

  We split cases on whether τ_x is trivial or serious.
  If τ_x is trivial By the Termination Hypothesis, it must be that the argument e_2 is trivial. By the definition of $\hookrightarrow_o$ the reduction happens only if the trivial e_2 is a value (otherwise e_2 is optimistically evaluated to a value before β-reduction). Hence, there is a value v, such that $e_2 \doteq v$ and $e' \doteq e_1\,[v/x]$. By Lemma 4 we get $\emptyset \vdash e_1\,[v/x] : \tau\,[v/x]$.
  Case 2: τ_x is serious: Then $e' \doteq e_1\,[e_2/x]$ we need to show that $\emptyset \vdash e_1\,[e_2/x] : \tau\,[e_2/x]$. By (b) we have $x : \tau_x \models [e_2/x]$ By well-formedness we are guaranteed that x does not appear in τ, or $\tau\,[e_2/x] = \tau$. So, from Lemma 4 we have $\emptyset \vdash e_1\,[e_2/x] : \tau$.
  - $e \equiv (\mu f.\lambda x.e_{11})\ v$ where $e' \equiv e\,[\mu f.\lambda x.e_{11}/f]\,[e_2/x]$
  From (a) and by inversion of the rule T-REC we have

$$f : (x{:}\tau_x \to \tau), x : \tau_x \vdash e_{11} : \tau$$

Since

$$\emptyset \vdash \mu f.\lambda x.e_{11} : (x{:}\tau_x \to \tau)$$

We have

$$f : (x{:}\tau_x \to \tau) \models [\mu f.\lambda x.e_{11}/f]$$

So, by Lemma 4 and since $f \notin freeVars(\tau)$

$$x : \tau_x \vdash e\,[\mu f.\lambda x.e_{11}/f] : \tau$$

As before, we split cases on whether τ_x is serious or trivial and we use Lemma 3 or Lemma 4 respectively to get

$$\emptyset \vdash e\,[\mu f.\lambda x.e_{11}/f]\,[e_2/x] : \tau\,[e_2/x]$$

- $e \equiv [\Lambda\alpha]\,e_{11}\ v$ Since $\emptyset \vdash [\Lambda\alpha]\,e_{11} : \forall\alpha.\tau'$ which is not a function type, this case cannot appear.

- Case T-LET. Assume

$$\emptyset \vdash \texttt{let}\ x = e_x\ \texttt{in}\ e_1 : \tau$$

where $\sigma \equiv \tau$ and $e \equiv \texttt{let}\ x = e_x\ \texttt{in}\ e_1$. By inversion

$$\emptyset \vdash e_x : \tau_x\ (a), \quad x : \tau_x \vdash e_1 : \tau\ (b), \quad \emptyset \vdash \tau\ (c)$$

We split cases on the structure of τ_x

- Say that e_x is serious. By Lemma 6, (a) and (b)

$$\emptyset \vdash e_1\,[e_x/x] : \tau$$

But $e' \equiv e_1\,[e_x/x]$ So,

$$\emptyset \vdash e' : \tau$$

The rest cases assume that e_x is trivial.

- Say that e_x is trivial and there exists a value v such that $e_x \equiv v$. From (a) $x : \tau_x \models [e_x/x]$ so, from Lemma 4

$$\emptyset \vdash e_1\,[e_x/x] : \tau\,[e_x/x]$$

From (c) $x \notin freeVars(\tau)$, so $\tau\,[e_x/x] \equiv \tau$. Also $e' \equiv e_1\,[e_x/x]$, so

$$\emptyset \vdash e' : \tau$$

- Say that e_x is trivial and not a value. So, there exists an e'_x such that $e_x \hookrightarrow_o e'_x$. Then by HI on (a)

$$\emptyset \vdash e'_x : \tau_x$$

By (b), (c) and rule T-LET

$$\emptyset \vdash \texttt{let}\ x = e'_x\ \texttt{in}\ e_1 : \tau$$

But $e' \equiv \texttt{let}\ x = e'_x\ \texttt{in}\ e_1$, so

$$\emptyset \vdash e' : \sigma$$

- Case T-INST. We assume that type annotations have been removed at runtime, so this case cannot occur.

□

Theorem 3 (Progress). *If $\emptyset \vdash e : \sigma$ and e is not a value, then there exists an e' so that $e \hookrightarrow_o e'$.*

Proof. We split cases on the type derivation:

- Case T-SUB. Assume

$$\emptyset \vdash e : \sigma$$

By inversion

$$\emptyset \vdash e : \sigma_1\ (a), \quad \emptyset \vdash \sigma_1 \preceq \sigma\ (b), \quad \emptyset \vdash \sigma(c)$$

By IH on (a), if e is not a value, there exists an e' so that $e \hookrightarrow_o e'$.

- Cases T-VAR-BASE, T-VAR, trivial, as we cannot typecheck a variable in an empty environment.
- Cases T-CON, T-FUN, T-REC and T-GEN are trivial as e is a value
- Case T-APP

$$\emptyset \vdash e_1\ e_2 : \tau[e_2/x]$$

where $e \equiv e_1\ e_2$ and $\sigma \equiv \tau\,[e_2/x]$. By inversion:

$$\emptyset \vdash e_1 : (x{:}\tau_x \to \tau)\ (a), \quad \emptyset \vdash e_2 : \tau_x\ (b)$$

If e_1 is not a value, by (a) and IH there exists an e_1 such that $e_1 \hookrightarrow_o e'_1$. Then $e_1\ e_2 \hookrightarrow_o e'_1\ e_2$. Otherwise, we split cases on the structure of e_1:

- $e_1 \equiv c$. If e_2 is not a value, then by IH and (b) there exists an e'_2 such that $e_2 \hookrightarrow_o e'_2$ and $e' \equiv c\,e'_2$. Otherwise, $e_2 \equiv v$ for some value v and $e' \equiv [|c|](v)$ which is well defined and is not equal to `crash` by the Definition 1.
- $e_1 \equiv \lambda x.e_x$. If e_2 is serious or a value, then $e' \equiv e_x\,[e_2/x]$. Otherwise, by IH and (b) there exists an e'_2 such that $e_2 \hookrightarrow_o e'_2$ and $e' \equiv e_1\ e'_2$.
- $e_1 \equiv \mu f.\lambda x.e_x$. If e_2 is serious or a value, then $e' \equiv e_x\,[e_2/x]\,[e_1/f]$.
Otherwise, by IH and (b) there exists an e'_2 such that $e_2 \hookrightarrow_o e'_2$ and $e' \equiv e_1\ e'_2$.
- $e_1 \equiv [\Lambda\alpha]\,e_\alpha$. This case cannot occur, because then e_1 should be typed as a type abstraction and not a function.

- Case T-LET. Assume

$$\emptyset \vdash \texttt{let}\ x = e_x\ \texttt{in}\ e_1 : \tau$$

where $e \equiv \texttt{let}\ x = e_x\ \texttt{in}\ e_1$ and $\sigma \equiv \tau$. By inversion

$$\emptyset \vdash e_x : \tau_x\ (a), \quad x : \tau_x \vdash e : \tau\ (b), \quad \emptyset \vdash \tau(c)$$

We split cases on the structure of e_x. If e_x is serious or a value, then $e' \equiv e_1\,[e_x/x]$. Otherwise, by (b) and IH, these exists an e'_x such that $e_x \hookrightarrow_o e'_x$ and $e' \equiv \texttt{let}\ x = e'_x\ \texttt{in}\ e_1$.

- Case T-INST. We assume that types are erased during run-time, thus this case cannot occur.

□

C. Proof of Termination Theorem

We prove Substitution Lemma 8 (Lemma 4 in the paper), and Termination Lemma 9 (Lemma 5 in the paper). Termination Theorem 4 (Theorem 3 in the paper) is a direct application of Termination Lemma.

Since the Lemmata are mutually dependent imagine we simultaneously reprove preservation, and prove Lemmata 8 and 9.

Definition 2 (Well-formed Terms). *A term e is well-formed with type σ, writing $\models e : \sigma$ iff*

1. *if σ is trivial, then there exist i, v such that $e \hookrightarrow_o^i v$,*
2. *if $\sigma \equiv x_1{:}\tau_{x_1} \to \cdots \to x_n{:}\tau_{x_n} \to \tau$ and τ is trivial, then for any expressions e_{x_i} such that $\emptyset \vdash e_{x_i} : \tau_{x_i}$ and $\models e_{x_i} : \tau_{x_i}$, for $1 \le i \le n$, there exist j, v such that $e\ e_{x_1} \ldots e_{x_n} \hookrightarrow_o^j v$, and*
3. *if $\sigma \equiv \forall\alpha.\sigma'$, then for any trivial type τ such that $\Gamma \vdash \tau$ $\models e\,[\tau] : \sigma'\,[\tau/\alpha]$*

Lemma 5. *If $\sigma\,[\tau/\alpha]$ is trivial and τ is trivial then σ is trivial.*

Proof. By induction on the structure of types.

- Case $\sigma \equiv \alpha'$ If $\alpha' \neq \alpha$ then $\sigma\,[\tau/\alpha] \equiv \sigma$ which is trivial. Otherwise, $\alpha' = \alpha$, so $\sigma\,[\tau/\alpha] \equiv \tau$ which is trivial by hypothesis.
- Case $\sigma \equiv \{v{:}b^l \mid e\}$. Then $\sigma\,[\tau/\alpha] \equiv \sigma$ which is trivial.
- Case $\sigma \equiv x{:}\tau_x \to \tau$. Then σ is trivial.

$$\boxed{\Gamma \models \theta}$$

$$\frac{}{\emptyset \models \emptyset} \qquad \frac{e \text{ is value or serious} \quad \Gamma \models \theta \quad \emptyset \vdash \theta e : \theta\sigma \quad \models \theta e : \theta\sigma}{\Gamma, x : \sigma \models \theta; [e/x]}$$

Figure 3. Well-formed Expression Substitution

- Case $\sigma \equiv \forall\alpha'.\sigma'$ If $\alpha' = \alpha$ then $\sigma\,[\tau/\alpha] \equiv \sigma$ which is trivial. Otherwise, $\alpha' \neq \alpha$, so $\sigma\,[\tau/\alpha] \equiv \forall\alpha'.(\sigma'\,[\tau/\alpha])$. By induction $\sigma'\,[\tau/\alpha]$ is trivial, hence, so is σ.

□

Lemma 6. *If $\Gamma \vdash \sigma \preceq \sigma'$ and σ is trivial, then σ' is trivial.*

Proof. By induction on the derivation $\Gamma \vdash \sigma \preceq \sigma'$. Cases $\preceq$-BASE-$\downarrow$, $\preceq$-FUN and $\preceq$-VAR are trivial, as both sides are trivial. In case $\preceq$-BASE-$\uparrow$ then Lemma is satisfied, as σ is not trivial. Finally, case $\preceq$-POLY proceeds by inversion of the rule and applying the inductive hypothesis. □

Lemma 7. *If $\models e : \sigma$ and $\emptyset \vdash \sigma \preceq \sigma'$, then $\models e : \sigma'$*

Proof. By induction on the derivation $\emptyset \vdash \sigma \preceq \sigma'$.

- Case $\preceq$-BASE-$\downarrow$. Assume
$$\emptyset \vdash \{v{:}b^{\downarrow} \mid e_1\} \preceq \{v{:}b^{\downarrow} \mid e_2\}$$
where $\sigma \equiv \{v{:}b^{\downarrow} \mid e_1\}$, and $\sigma' \equiv \{v{:}b^{\downarrow} \mid e_2\}$. Only the first case of well-formed expressions apply and since σ is trivial, e should converge.
- Case $\preceq$-BASE-$\uparrow$. Assume
$$\emptyset \vdash \{v{:}b^{l} \mid e\} \preceq \{v{:}b^{\uparrow} \mid \texttt{true}\}$$
No case of well-formed expressions apply, thus the Lemma is trivially satisfied.
- Case $\preceq$-FUN. Assume
$$\emptyset \vdash x{:}\tau_1 \rightarrow \tau_1' \preceq x{:}\tau_2 \rightarrow \tau_2'$$
where $\sigma \equiv x{:}\tau_1 \rightarrow \tau_1'$ and $\sigma' \equiv x{:}\tau_2 \rightarrow \tau_2'$. By inversion
$$\emptyset \vdash \tau_2 \preceq \tau_1\ (a) \qquad x : \tau_2 \vdash \tau_1' \preceq \tau_2'\ (b)$$
The first case of well-formed expressions is satisfied, as σ is trivial, thus should e converge. For the second case, let τ_1' be trivial, then by (b) and Lemma 6, τ_2' is trivial. Assume an e_x such that $\emptyset \vdash e_x : \tau_2$ and $\models e_x : \tau_2$. By (a) and IH $\emptyset \vdash e_x : \tau_1$ and $\models e_x : \tau_1$. But since $\models e : \sigma'$, and τ_2' is trivial, $e\ e_x$ converges.
The third case does not apply.
- Case $\preceq$-VAR is trivial, as only the first case applies and e should converge.
- Case $\preceq$-POLY. Assume
$$\Gamma \vdash \forall\alpha.\sigma_1 \preceq \forall\alpha.\sigma_2$$
By inversion
$$\Gamma \vdash \sigma_1 \preceq \sigma_2$$
For the first case, if σ' is trivial, then σ is trivial and e converges. The second case does not apply. For the third case, for any trivial τ, such that $\emptyset \vdash \tau$, $\models e\,[\tau] : \sigma_1\,[\tau/\alpha]$, but $\emptyset \vdash \sigma_1\,[\tau/\alpha] \preceq \sigma_2\,[\tau/\alpha]$, so $\models e\,[\tau] : \sigma_2\,[\tau/\alpha]$.

□

Lemma 8 (Substitution Lemma)**.** *If $\Gamma \models \theta$, then*

- *If $\Gamma \vdash e : \sigma$, then $\emptyset \vdash \theta e : \theta\sigma$.*
- *If $\Gamma \vdash \sigma \preceq \sigma'$, then $\emptyset \vdash \theta\sigma \preceq \theta\sigma'$.*

Proof. Let $\Gamma = \Gamma', x : \sigma_x$ and $\theta = \theta'; [e_x/x]$. From the definition of well-formed substitutions, we have

$$e \text{ is value or serious } (a), \quad \emptyset \vdash \theta e : \theta\sigma\ (b), \quad \models \theta e : \theta\sigma\ (c)$$

By (a), e_x is either a value or serious, so either Lemma 4 or Lemma 6 applies. In either case we get $\Gamma' \vdash e\,[e_x/x] : \sigma\,[e_x/x]$. The Lemma follows from iteratively applying this reasoning.

Similarly, the second case follows from Lemma 4 or Lemma 2. □

Lemma 9 (Termination Lemma)**.** *If $\Gamma \vdash e : \sigma$ and $\Gamma \models \theta$, then $\models \theta e : \theta\sigma$.*

Proof. We assume a substitution θ such that $\Gamma \models \theta$. We prove the Lemma by induction on the typing derivation tree $\Gamma \vdash e : \sigma$ (each time we push the rule T-SUB down in the tree.)

- Case T-SUB. Assume
$$\Gamma \vdash e : \sigma$$
and σ is trivial. By inversion
$$\Gamma \vdash e : \sigma'\ (a), \quad \Gamma \vdash \sigma' \preceq \sigma\ (b), \quad \Gamma \vdash \sigma\ (c)$$
By IH on (a), $\models \theta e : \theta\sigma'(d)$. By (b) and Lemma 8 $\emptyset \vdash \theta\sigma' \preceq \theta\sigma$. By which, (d) and Lemma 7 we get $\models \theta e : \theta\sigma$
- Cases T-VAR-BASE. Assume
$$\Gamma \vdash x : \{v{:}b^{l} \mid v = x\}$$
By inversion
$$\Gamma(x) = \{v{:}b^{l} \mid e_x'\}$$
Since $\Gamma \models \theta$ there exists an e_x such that $[e_x/x] \in \theta$ and $\models \theta e_x : \theta\{v{:}b^{l} \mid e_x\}$. By which, if $l \equiv \downarrow$ then $\theta x = \theta e_x$ converges. Otherwise none of the cases apply.
- Cases T-VAR. Assume
$$\Gamma \vdash x : \sigma$$
By inversion
$$\Gamma(x) = \sigma$$
Since $\Gamma \models \theta$ there exists an e_x such that $[e_x/x] \in \theta$ and $\models \theta e_x : \theta\sigma$ But, $\theta x = \theta e_x$.
- Case T-CON. Assume
$$\Gamma \vdash c : ty(c)$$
The lemma trivially holds, as by Definition 1, $\models c : ty(c)$
- Case T-FUN. Assume
$$\Gamma \vdash (\lambda x.e') : x{:}\tau_x \rightarrow \tau\ (a)$$
where $e \equiv \lambda x.e'$ and $\sigma \equiv x{:}\tau_x \rightarrow \tau$ which is trivial. We will prove the three requirements of well-formed expressions.
  1. Trivial, since e is a value.
  2. Say that τ is trivial and assume some e_x such that $\emptyset \vdash e_x : \tau_x$ and $\models e_x : \theta\tau_x$. So, $\Gamma; x : \theta\tau_x \models \theta; [e_x/x]$.
  By inversion of the rule (a) we get
  $$\Gamma, x : \tau_x \vdash e' : \tau$$
  By IH and 1 we get that $(\theta; [e_x/x])e' \hookrightarrow_o^i v$. But $(\theta; [e_x/x])e' = \theta(\lambda x.e' e_x)$. So, $(\theta e)(e_x) \hookrightarrow_o^i v$.
  3. The third case cannot occur.

- Case T-App. Assume

$$e \equiv e_1\, e_2$$

By inversion we get

$$\Gamma \vdash e_1 : (x{:}\tau_x \to \tau)\ (a) \quad \Gamma \vdash e_2 : \tau_x\ (b)$$

where τ is trivial. By inductive hypothesis, we get that $\models \theta e_1 : \theta(x{:}\tau_x \to \tau)$ and $\models \theta e_2 : \theta\tau_x$. By Lemma 8 on (b) we get $\emptyset \vdash \theta e_2 : \theta\tau_x$. So, by 2 on e_1 we get that there exist i, v such that

$$\theta e = (\theta e_1)(\theta e_2) \hookrightarrow_o^i v$$

To prove 2 suppose that

$$\tau \equiv x_1{:}\tau_{x_1} \to \cdots \to x_n{:}\tau_{x_n} \to \tau'$$

Again by 2 on e_1, for $n := n+1$, we get for any expressions $e_{x_1}, \dots, e_{x_n}$ such that $\emptyset \vdash e_{x_i} : \theta\tau_{x_i}$ and $\models e_{x_i}; \theta\tau_{x_i}$ for $1 \le i \le n$, there exist j, v such that $(\theta e)\ e_{x_1} \dots e_{x_n} \equiv (\theta e_1)(\theta e_2)\ e_{x_1} \dots e_{x_n} \hookrightarrow_o^j v$

The third case does not apply.
- Case T-Let. Assume

$$\Gamma \vdash \mathtt{let}\ x = e_x\ \mathtt{in}\ e' : \tau$$

where $e \equiv \mathtt{let}\ x = e_x\ \mathtt{in}\ e'$ and $\sigma \equiv \tau$ which is trivial.
By inversion

$$\Gamma \vdash e_x : \tau_x(a), \qquad \Gamma, x : \tau_x \vdash e' : \tau(b)$$

We define a substitution θ' as follows: If τ_x is trivial, then $\theta e_x \hookrightarrow_o v_x i_x$ and $\theta' = \theta; [v_x/x]$. Otherwise, $\theta' = \theta; [e_x/x]$. In either case $\Gamma; x : \tau_x \models \theta'$

The lemma holds by IH on b using the substitution θ'.
- Case T-Rec-$\downarrow$. Assume

$$\Gamma \vdash \mu f.\lambda x.e' : x{:}\tau_x \to \tau\ (a)$$

where τ is trivial $e \equiv \mu f.\lambda x.e'$. The first case is trivial, as e is a value and the third does not apply. So, we need to prove that for any expression e_x such that

$$\emptyset \vdash e_x : \theta\tau_x\ (b) \qquad \emptyset \models e_x : \theta\tau_x\ (c)$$

there exist j, v such that $(\theta e)(e_x) \hookrightarrow_o^j v$.
By inverting the rule (a) we get

$$\tau_x = \{v{:}\mathtt{nat}^{\downarrow} \mid e_x\}\ (d) \quad \tau_y = \{v{:}\mathtt{nat}^{\downarrow} \mid e_x \wedge v < x\}\ (e)$$

$$\Gamma, x : \tau_x, f : y{:}\tau_y \to \tau\,[y/x] \vdash e : \tau\ (e)$$

Let N the set of *values* described by $\theta\tau_x$. By (d), $N \subseteq \mathbb{N}$ and $n \in N \Leftrightarrow \theta\,[n/v]\,e_x \hookrightarrow_o^* \mathtt{true}$. Since, $N \subseteq \mathbb{N}$, N is enumerable, thus consider its enumeration n_i, where $i \ge 0$ and $n_i < n_j \Leftrightarrow i < j$. From $((b))$ and $((c))$ $e_x \hookrightarrow_o^{i_x} n_x$ and from soundness of $\lambda_\downarrow$, $n_x \in N$. Also, $(\theta e)(e_x) \hookrightarrow_o^{i_x} (\theta e) n_x$. Hence, it suffices to prove that there exists v', j' such that $(\theta e) n_x \hookrightarrow_o^{j'} v'$. We will prove it by induction on N:
  - *Base case ($n_x = n_0$):* By (e) $(\theta\,[n_0/x])\tau_y = \{v{:}\mathtt{nat}^{\downarrow} \mid \theta e_x \wedge v < n_0\} = \{v{:}\mathtt{nat}^{\downarrow} \mid \mathtt{false}\}$. Hence, there is no natural value n such that $\emptyset \vdash n : (\theta\,[n_0/x])\tau_y$; which trivially gives us that $\models \theta\,[n_0/x]\,e : \theta\,[n_0/x]\,(y{:}\tau_y \to \tau\,[y/x])$. By Lemma 8 and (a) we get $\emptyset \vdash \theta\,[n_0/x]\,e : \theta\,[n_0/x]\,(y{:}\tau_y \to \tau\,[y/x])$. So, $\theta' = \theta\,[n_x/x]\,[e/f]$ is well-formed under $\Gamma' = \Gamma; x : \tau_x; f : y{:}\tau_y \to \tau\,[y/x]$ and by Inductive Hypothesis on (a), $\models \theta' e' : \theta'\tau$. Since τ is trivial, $\theta'\tau$ is trivial thus, there exist v_0, i_0 such that $\theta' e' \hookrightarrow_o^{i_0} v_0$. But, $(\theta e) n_x \hookrightarrow_o (\theta\,[n_x/x]\,[e/f]) e' = \theta' e'$. So $(\theta e) n_x \hookrightarrow_o^{i_0+1} v_0$.
  - *Inductive Step:* Suppose for all $i' < i$ there exist $v_{i'}, i_{i'}$ such that

$$(\theta e) n_{i'} \hookrightarrow_o^{i_{i'}} v_{i'}$$

By (e), $\theta\,[n_i/x]\,\tau_y = \{v{:}\mathtt{nat}^{\downarrow} \mid \theta e_x \wedge v < n_i\}$.
For any expression e_y such that $\emptyset \vdash e_y : (\theta\,[n_i/x])\tau_y$ and $\models e_y : (\theta\,[n_i/x])\tau_y$, there exist v_y, i_y such that $e_y \hookrightarrow_o^{i_y} v_y$ and from preservation $\vdash v_y : (\theta\,[n_i/x])\tau_y$. Hence, there exists an $l < i$ such that $v_y = n_l$. By Inductive Hypothesis on N, there exist v_l, i_l such that $(\theta e)\ (e_y) \hookrightarrow_o^{i_y} (\theta e)\ n_l \hookrightarrow_o^{i_l} v_l$. So,

$$\models (\theta\,[n_i/x]) e : (\theta\,[n_i/x])(y{:}\tau_y \to \tau\,[y/x])$$

By Lemma 8 and (a) we get

$$\emptyset \vdash (\theta\,[n_i/x]) e : (\theta\,[n_i/x])(y{:}\tau_y \to \tau\,[y/x])$$

So, $\theta' = \theta\,[n_i/x]\,[e/f]$ is well-formed under $\Gamma' = \Gamma, x : \tau_x, f : y{:}\tau_y \to \tau\,[y/x]$ and by Inductive Hypothesis on (a), $\models \theta' e' : \theta'\tau$. Since τ is trivial, $\theta'\tau$ is trivial; thus, there exist v_i, i_i such that $\theta' e' \hookrightarrow_o^{i_i} v_i$. But, $(\theta e) n_i \hookrightarrow_o (\theta\,[n_i/x]\,[e/f]) e' = \theta' e'$. So, $(\theta e) n_i \hookrightarrow_o^{i_i+1} v_i$.
- Case T-Rec-$\uparrow$. Assume

$$\Gamma \vdash \mu f.\lambda x.e' : x{:}\tau_x \to \tau$$

By inversion we get that τ is serious. We will prove the three requirements of well-formed expressions.
  1. Trivial, since, e is a value
  2. Trivial, since τ is serious.
  3. This case does not apply
- Case T-Gen. Assume

$$\Gamma \vdash [\Lambda\alpha]\, e' : \forall\alpha.\sigma'$$

where $e \equiv [\Lambda\alpha]\, e'$ and $\sigma \equiv \forall\alpha.\sigma'$ which is trivial. By inversion

$$\Gamma \vdash e' : \sigma'\ (a)$$

We will prove the three requirements of well-formed expressions.
  1. Trivial since e is a value.
  2. The second case cannot occur.
  3. Assume a trivial τ, such that $\emptyset \vdash \tau$. Then $\theta(e\,[\tau]) = \theta([\Lambda\alpha]\, e'\tau) \hookrightarrow_o \theta e'$. Moreover, σ' is trivial because σ is trivial. So, we can apply IH on (a) and get $\models \theta(e\,[\tau]) : \theta\sigma'$, for any α thus $\models \theta(e\,[\tau]) : \theta(\sigma'\,[\tau/\alpha])$.
- Case T-Inst. Assume

$$\Gamma \vdash e'\,[\tau] : \sigma'\,[\tau/\alpha]$$

where $e \equiv e'\,[\tau]$ and $\sigma \equiv \sigma'\,[\tau/\alpha]$ which is trivial. By inversion,

$$\Gamma \vdash e' : \forall\alpha.\sigma'\ (a), \qquad \Gamma \vdash \tau\ (b), \qquad \tau \text{ trivial } (c)$$

By Lemma 5 and since σ it trivial, $\forall\alpha.\sigma'$ is trivial.
So by IH on (a) using 3 we have that since (b) and (c) hold, $\models \theta e : \theta\sigma$.

□

Theorem 4 (Termination)**.** *If $\emptyset \vdash e : \sigma$ and σ is trivial, then e is trivial.*

Proof. Direct implication of Lemma 9 with $\Gamma = \emptyset$ and $\theta = \emptyset$. □

References

[1] P. Rondon, M. Kawaguchi, and R. Jhala. Liquid types. Technical Report.