Normal Form Bisimulation for Typed Calculi: Syntactic Minimal Invariance (Draft March 8, 2007)

Soren B. Lassen
Google, Inc.
soren@google.com

Paul Blain Levy
University of Birmingham, U.K.
pbl@cs.bham.ac.uk

This note uses the normal form bisimulation theory for recursively typed call-by-push-value (CBPV) [1] to prove a "syntactic minimal invariance" result.

- SOREN: NEED TO CHANGE FUNCTION APPLICATION SYNTAX TO OPERAND-FIRST EVERYWHERE

Given closed types A, $\underline{B}$, we define the function types:

$$A^\dagger \stackrel{\text{def}}{=} A \to FA, \quad \underline{B}^\dagger \stackrel{\text{def}}{=} U\underline{B} \to \underline{B}.$$

and closed terms $\vdash^c \gamma^v_A : A^\dagger$ and $\vdash^c \gamma^c_{\underline{B}} : \underline{B}^\dagger$. More generally, to deal with recursive types, we define in Figure , by structural induction on A and $\underline{B}$, open terms:

$$\Gamma \vdash^c \gamma^v_{\Gamma,A} : A[\Gamma]^\dagger, \quad \Gamma \vdash^c \gamma^c_{\Gamma,\underline{B}} : \underline{B}[\Gamma]^\dagger.$$

where $\Gamma = \vec{X} : \overrightarrow{UA^\dagger}, \vec{\underline{Y}} : \overrightarrow{U\underline{B}^\dagger}$ (we take the liberty to use $\vec{X}$ and $\vec{\underline{Y}}$ as term identifiers in Γ, $\gamma^v_{\Gamma,A}$, $\gamma^c_{\Gamma,\underline{B}}$ and as type identifiers in A and $\underline{B}$); $\vec{A}$ and $\vec{\underline{B}}$ are closed types; A and $\underline{B}$ are open types: $\vec{X}, \vec{\underline{Y}} \vdash^v A$ type, $\vec{X}, \vec{\underline{Y}} \vdash^c \underline{B}$ type; and $[\Gamma]$ denotes the type substitution $[\vec{A}/\vec{X}, \vec{\underline{B}}/\vec{\underline{Y}}]$.

Proposition 1. *$\gamma^v_{A[\Gamma]} = \gamma^v_{\Gamma,A}[\overrightarrow{\gamma^v_A/X}, \overrightarrow{\gamma^v_{\underline{B}}/\underline{Y}}]$ and $\gamma^c_{\underline{B}[\Gamma]} = \gamma^c_{\Gamma,\underline{B}}[\overrightarrow{\gamma^v_A/X}, \overrightarrow{\gamma^v_{\underline{B}}/\underline{Y}}]$.*

Proof. By structural induction on A and $\underline{B}$. □

So, for closed recursive types $\texttt{Rec}X.A$, $\texttt{Rec}\underline{Y}.\underline{B}$,

$$\gamma^v_{A[\texttt{Rec}X.A/X]} = \gamma^v_{X:(\texttt{Rec}X.A),A}[\gamma^v_{\texttt{Rec}X.A}/X],$$
$$\gamma^c_{\underline{B}[\texttt{Rec}\underline{Y}.\underline{B}/\underline{Y}]} = \gamma^v_{\underline{Y}:(\texttt{Rec}\underline{Y}.\underline{B}),\underline{B}}[\gamma^v_{\texttt{Rec}\underline{Y}.\underline{B}}/\underline{Y}].$$

Observe that

$$x : U\underline{B} \vdash \gamma^v_{U\underline{B}}(x), \texttt{nil} \rightsquigarrow^* \texttt{return}\, x^\ddagger, \texttt{nil}$$
$$x : UFA \vdash \gamma^c_{FA}(x), K \rightsquigarrow^* \texttt{force}\, x, K^\ddagger$$

$$\begin{aligned}
\gamma^v_{\Gamma,U\underline{B}} &= \lambda x.\, \texttt{return}\, \texttt{thunk}(\gamma^c_{\Gamma,\underline{B}}(x)) \\
\gamma^v_{\Gamma,1} &= \lambda\langle\,\rangle.\langle\,\rangle \\
\gamma^v_{\Gamma,A_1\times A_2} &= \lambda\langle x_1, x_2\rangle.\, \gamma^v_{\Gamma,A_1}(x_1) \\
&\qquad \texttt{to}\, y_1.\, \gamma^v_{\Gamma,A_2}(x_2) \\
&\qquad \texttt{to}\, y_2.\, \texttt{return}\, \langle y_1, y_2\rangle \\
\gamma^v_{\Gamma,\Sigma_{i\in I}A_i} &= \lambda\{\langle i, x\rangle.\, \gamma^v_{\Gamma,A_i}(x)\, \texttt{to}\, y.\, \texttt{return}\, \langle i, y\rangle\}_{i\in I} \\
\gamma^v_{\Gamma,X_i} &= \texttt{force}\, X_i \\
\gamma^v_{\Gamma,\texttt{Rec}X.A} &= \texttt{rec}X.\, \lambda\texttt{fold}\, x.\, \gamma^v_{\Gamma,X:U(\texttt{Rec}X.A)[\Gamma]^\dagger,A}(x) \\
&\qquad \texttt{to}\, y.\, \texttt{return}\, \texttt{fold}\, y \\
\gamma^c_{\Gamma,FA} &= \lambda x.\, \texttt{force}\, x\, \texttt{to}\, y.\, \gamma^v_{\Gamma,A}(y) \\
\gamma^c_{\Gamma,A\to\underline{B}} &= \lambda x.\, \lambda y.\, \gamma^v_{\Gamma,A}(y) \\
&\qquad \texttt{to}\, z.\, \gamma^c_{\Gamma,\underline{B}}(\texttt{thunk}(\texttt{force}\, x(z))) \\
\gamma^c_{\Gamma,\Pi_{i\in I}} &= \lambda x.\, \lambda\{i.\, \gamma^c_{\Gamma,A_i}(\texttt{thunk}(\pi_i(\texttt{force}\, x)))\}_{i\in I} \\
\gamma^v_{\Gamma,\underline{Y}_i} &= \texttt{force}\, \underline{Y}_i \\
\gamma^v_{\Gamma,\texttt{Rec}\underline{Y}.\underline{B}} &= \texttt{rec}\underline{Y}.\, \lambda x.\, \texttt{fold} \\
&\qquad \gamma^c_{\Gamma,\underline{Y}:U(\texttt{Rec}\underline{Y}.\underline{B})[\Gamma]^\dagger,\underline{B}} \\
&\qquad (\texttt{thunk}(\texttt{unfold}(\texttt{force}\, x)))
\end{aligned}$$

Figure 1. Definitions of $\gamma^v_{\Gamma,A}$ and $\gamma^c_{\Gamma,\underline{B}}$

where $x^\ddagger \stackrel{\text{def}}{=} \texttt{thunk}(\gamma^c_{\underline{B}}(x))$ and $K^\ddagger \stackrel{\text{def}}{=} \texttt{to}\ y.\,\gamma^v_A(y)::K$, provided x is of type $U\underline{B}$ and K is from type FA. More generally:

Lemma 2. *If* $\vec{x} : \overrightarrow{U\underline{B}} \vdash^v p(\vec{x}) : A$ *and* $\Gamma \mid FA \vdash^k K : FC$ *then*

$$\Gamma, \vec{x} : \overrightarrow{U\underline{B}} \vdash \gamma^v_A(p(\vec{x})), K \rightsquigarrow^* \texttt{return}\, p(\vec{x^\ddagger}), K$$

and, if $\Gamma \vdash^v V : U\underline{B}$, $\vec{x} : \overrightarrow{U\underline{A}} \mid \underline{B} \vdash^k q(\vec{x}; K) : \underline{D}$, *and* $\Gamma \mid FC \vdash^k K : \underline{D}$,

$$\Gamma, \vec{x} : \overrightarrow{U\underline{A}} \vdash \gamma^c_{\underline{B}}(V), q(\vec{x}; K) \rightsquigarrow^* \texttt{force}\, V, q(\vec{x^\ddagger}; K^\ddagger).$$

Proof. By structural induction on p and q. □

Theorem 3 (Syntactic minimal invariance)**.** *For all closed value types A and closed computation types $\underline{B}$,*

$$\vdash^c \gamma^v_A \approx \lambda z.\,\texttt{return}\, z : A^\dagger,$$
$$\vdash^c \gamma^c_{\underline{B}} \approx \lambda z.\,\texttt{force}\, z : \underline{B}^\dagger.$$

Proof. The equations follow from

$$z : A \vdash^c \gamma^v_A(z) \approx \texttt{return}\, z : FA,$$
$$z : U\underline{B} \vdash^c \texttt{force}\, z^\ddagger \approx \texttt{force}\, z : \underline{B},$$

which we prove by the exhibiting the following normal form bisimulation $\mathcal{R}$ which relates

$$\vec{x} : \overrightarrow{U\underline{A}} \vdash \gamma^v_A(p(\vec{x})), \texttt{nil}\ \mathcal{R}\ p(\vec{x}), \texttt{nil} : FA$$
$$\vec{y} : \overrightarrow{U\underline{B}}, z : U\underline{B} \vdash \texttt{force}\, z^\ddagger, q(\vec{y}; \texttt{nil})\ \mathcal{R}$$
$$\texttt{force}\, z, q(\vec{y}; \texttt{nil}) : FC$$

whenever $\vec{x} : \overrightarrow{U\underline{A}} \vdash^v p(\vec{x}) : A$ and $\vec{y} : \overrightarrow{U\underline{B}} \mid \underline{B} \vdash^k q(\vec{y}; \texttt{nil}) : FC$. By Lemma 2, $\mathcal{R}$ is a normal form bisimulation. □

References

[1] S. B. Lassen and P. B. Levy. Normal form bisimulation for typed calculi. Conference submission, 2007.

A Proofs

Here are some of the cases in the proof of Lemma 2.

Case $A = A_1 \times A_2$. Then there exist p_1, $\vec{x}_1$, $\underline{\vec{B}}_1$, p_2, $\vec{x}_2$, and $\underline{\vec{B}}_2$ such that

$$\begin{aligned}
\vec{x} &= \vec{x}_1, \vec{x}_2, \\
\underline{\vec{B}} &= \underline{\vec{B}}_1, \underline{\vec{B}}_2, \\
p(\vec{x}) &= \langle p_1(\vec{x}_1), p_2(\vec{x}_2) \rangle, \\
\vec{x}_i : \overrightarrow{U\underline{B}_i} &\vdash^v p_i(\vec{x}_i) : A_i,
\end{aligned}$$

for $i \in \{1, 2\}$. We use the abbreviations $M[V_1, V_2] \stackrel{\text{def}}{=} \gamma^v_{A_1}(V_1)\ \mathtt{to}\ y_1.\, N[y_1, V_2]$ and $N[V_1, V_2] \stackrel{\text{def}}{=} \gamma^v_{A_2}(V_2)\ \mathtt{to}\ y_2.\, \mathtt{return}\ \langle V_1, y_2 \rangle$ in the following calculation.

$$\begin{array}{lll}
\gamma^v_A(p(\vec{x})), & K & \rightsquigarrow \\
\gamma^v_A, & p(\vec{x})::K & = \\
\lambda\langle z_1, z_2 \rangle.\, M[z_1, z_2], & \langle p_1(\vec{x}_1), p_2(\vec{x}_2) \rangle::K & \rightsquigarrow^* \\
M[p_1(\vec{x}_1), p_2(\vec{x}_2)], & K & \rightsquigarrow^* \\
\gamma^v_{A_1}(p_1(\vec{x}_1)), & \mathtt{to}\ y_1.\, N[y_1, V_2]::K & \rightsquigarrow^* \text{ by the I.H.} \\
\mathtt{return}\ p_1(\overrightarrow{x_1^{\ddagger}}), & \mathtt{to}\ y_1.\, N[y_1, p_2(\vec{x}_2)]::K & \rightsquigarrow^* \\
\gamma^v_{A_2}(p_2(\vec{x}_2)), & \mathtt{to}\ y_2.\, \mathtt{return}\ \langle p_1(\overrightarrow{x_1^{\ddagger}}), y_2 \rangle::K & \rightsquigarrow^* \text{ by the I.H.} \\
\mathtt{return}\ p_2(\overrightarrow{x_2^{\ddagger}}), & \mathtt{to}\ y_2.\, \mathtt{return}\ \langle p_1(\overrightarrow{x_1^{\ddagger}}), y_2 \rangle::K & \rightsquigarrow \\
\mathtt{return}\ \langle p_1(\overrightarrow{x_1^{\ddagger}}), p_2(\overrightarrow{x_2^{\ddagger}}) \rangle, & K & = \\
\mathtt{return}\ \langle p(\overrightarrow{x^{\ddagger}}) \rangle & K &
\end{array}$$

Case $A = \Sigma_{i \in I} A_i$. Then there exist $i \in I$ and p' such that $p(\vec{x}) = \langle i, p'(\vec{x}) \rangle$ and $\vec{x} : \overrightarrow{U\underline{B}} \vdash^v p'(\vec{x}) : A_i$.

$$\begin{array}{lll}
\gamma^v_A(p(\vec{x})), & K & \rightsquigarrow \\
\gamma^v_A, & p(\vec{x})::K & = \\
\lambda\{\langle i, x \rangle.\, \gamma^v_{A_i}(x)\ \mathtt{to}\ y.\, \mathtt{return}\ \langle i, y \rangle\}_{i \in I}, & \langle i, p'(\vec{x}) \rangle::K & \rightsquigarrow^* \\
\gamma^v_{A_i}(p'(\vec{x}))\ \mathtt{to}\ y.\, \mathtt{return}\ \langle i, y \rangle, & K & \rightsquigarrow \\
\gamma^v_{A_i}(p'(\vec{x})), & \mathtt{to}\ y.\, \mathtt{return}\ \langle i, y \rangle::K & \rightsquigarrow^* \text{ by the I.H.} \\
\mathtt{return}\ p'(\overrightarrow{x^{\ddagger}}), & \mathtt{to}\ y.\, \mathtt{return}\ \langle i, y \rangle::K & \rightsquigarrow \\
\mathtt{return}\ \langle i, p'(\overrightarrow{x^{\ddagger}}) \rangle, & K & = \\
\mathtt{return}\ p(\overrightarrow{x^{\ddagger}}), & K &
\end{array}$$

Case $\underline{B} = \Pi_{i \in I} \underline{B_i}$. Then there exist $i \in I$ and q' such that $q(\vec{x}; K) = i::q'(\vec{x}; K)$ and $\vec{X} : \overrightarrow{U\underline{A}} \mid \underline{B_i} \vdash^k q'(\vec{x}; K) : FC$.

$$\begin{array}{lll}
\gamma^c_{\underline{B}}(V), & q(\vec{x}; K) & \rightsquigarrow^* \\
\lambda\{i.\, \gamma^c_{\Gamma, \underline{B_i}}(\mathtt{thunk}(\pi_i(\mathtt{force}\, V)))\}_{i \in I}, & q(\vec{x}; K) & = \\
\lambda\{i.\, \gamma^c_{\Gamma, \underline{B_i}}(\mathtt{thunk}(\pi_i(\mathtt{force}\, V)))\}_{i \in I}, & i::q'(\vec{x}; K) & \rightsquigarrow \\
\gamma^c_{\Gamma, \underline{B_i}}(\mathtt{thunk}(\pi_i(\mathtt{force}\, V))), & q'(\vec{x}; K) & \rightsquigarrow^* \text{ by the I.H.} \\
\mathtt{force}\ \mathtt{thunk}(\pi_i(\mathtt{force}\, V)), & q'(\overrightarrow{x^{\ddagger}}; K) & \rightsquigarrow^* \\
\pi_i(\mathtt{force}\, V), & q'(\overrightarrow{x^{\ddagger}}; K) & \rightsquigarrow \\
\mathtt{force}\, V, & i::q'(\overrightarrow{x^{\ddagger}}; K) & = \\
\mathtt{force}\, V, & q(\overrightarrow{x^{\ddagger}}; K) &
\end{array}$$

Case $\underline{B} = \mathtt{Rec}\underline{Y}.\underline{B_0}$. Then there exists q' such that $q(\vec{x}; K) = \mathtt{unfold}::q'(\vec{x}; K)$ and $\vec{X} : \overrightarrow{U\underline{A}} \mid \underline{B}' \vdash^k q'(\vec{x}; K) : FC$, where $\underline{B}' \stackrel{\text{def}}{=} \underline{B_0}[\underline{B}/\underline{Y}]$. Observe that

$$\gamma^c_{\underline{B}} = \mathtt{rec}\underline{Y}.\, \lambda x.\, \mathtt{fold}\, \gamma^c_{\underline{Y}:U\underline{B}^\dagger, \underline{B_0}}(\mathtt{thunk}(\mathtt{unfold}(\mathtt{force}\, x)))$$

and $\gamma^c_{\underline{Y}:U\underline{B}^\dagger,\underline{B_0}}[\texttt{thunk}(\gamma^c_{\underline{B}})/\underline{Y}] = \gamma^c_{\underline{B}'}$ and, if L is a stack from type $\underline{B}^\dagger$,

$$\gamma^c_{\underline{B}}, L \rightsquigarrow^* \lambda x.\, \texttt{fold}\, \gamma^c_{\underline{B}'}(\texttt{thunk}(\texttt{unfold}(\texttt{force}\, x))), L \tag{1}$$

$$\begin{array}{lll l}
\gamma^c_{\underline{B}}(V), & q(\vec{x}; K) & \rightsquigarrow & \\
\gamma^c_{\underline{B}}, & V{::}q(\vec{x}; K) & \rightsquigarrow^* & (1) \\
\lambda x.\, \texttt{fold}\, \gamma^c_{\underline{B}'}(\texttt{thunk}(\texttt{unfold}(\texttt{force}\, x))), & V{::}q(\vec{x}; K) & \rightsquigarrow & \\
\texttt{fold}\, \gamma^c_{\underline{B}'}(\texttt{thunk}(\texttt{unfold}(\texttt{force}\, V))), & q(\vec{x}; K) & = & \\
\texttt{fold}\, \gamma^c_{\underline{B}'}(\texttt{thunk}(\texttt{unfold}(\texttt{force}\, V))), & \texttt{unfold} :: q'(\vec{x}; K) & \rightsquigarrow & \\
\gamma^c_{\underline{B}'}(\texttt{thunk}(\texttt{unfold}(\texttt{force}\, V))), & q'(\vec{x}; K) & \rightsquigarrow^* & \text{by the I.H.} \\
\texttt{force thunk}(\texttt{unfold}(\texttt{force}\, V)), & q'(\vec{x^\ddagger}; K^\ddagger) & \rightsquigarrow^* & \\
\texttt{unfold}(\texttt{force}\, V), & q'(\vec{x^\ddagger}; K^\ddagger) & \rightsquigarrow & \\
\texttt{force}\, V, & \texttt{unfold}{::}q'(\vec{x^\ddagger}; K^\ddagger) & = & \\
\texttt{force}\, V, & q(\vec{x^\ddagger}; K^\ddagger) & &
\end{array}$$