A Framework for Equivalence of Cast Calculi

by Lu Kuang-Chen

1 Definitions

Definition 1.1. *$s \approx^* s'$ if and only if there exists a state t such that $s \longrightarrow^* t$ and $t \approx s'$. Intuitively, $s \approx^* s'$ means s will be related to s'.*

Notation 1.2. *$[\![c]\!]$ means $\lambda V.\mathrm{applyCast}(V, c)$.*

Lemma 1.3. *If $U \approx U' : A$ then $[\![A \Rightarrow^l B]\!](U) \approx [\![\lceil A \Rightarrow^l B \rceil]\!](U') : B$.*

Definition 1.4. $\mathrm{extCont}(c, [\square\langle d\rangle]k) = [\square\langle c;d\rangle]k$

Lemma 1.5. $\mathrm{extCont}(c;d, \kappa) = \mathrm{extCont}(c, \mathrm{extCont}(d, \kappa))$

Lemma 1.6. *If $\kappa \approx \kappa'$ then $[\square\langle A \Rightarrow^l B\rangle]\kappa \approx \mathrm{extCont}(\lceil A \Rightarrow^l B \rceil, \kappa')$*

Lemma 1.7. *For all $U \approx U' : S \to T$ and $V \approx V' : S$ and $k \approx k' : T$*

$$\mathrm{doApp}(U, V, \kappa) \approx^\star \mathrm{doApp}'(U', V', \kappa')$$

Proof. By induction on $U \approx U'$. There are two cases.

Case $\langle \lambda(x : S) : T.\, e, E\rangle \approx \langle \lambda(x : \mathrm{id}(S)) : \mathrm{id}(T).\, e, E'\rangle$

In this case

$$\begin{aligned}
& \mathrm{doApp}(U, V, k) \\
= \; & \mathrm{doApp}(\langle \lambda(x : A) : B.\, e, E\rangle, V, \kappa) \\
= \; & \langle e, E[x \leftarrow V], \kappa\rangle
\end{aligned}$$

$$\begin{aligned}
& & \mathrm{doApp}'(U', V', \kappa') \\
& = & \mathrm{doApp}'(\langle \lambda(x : \mathrm{id}(A)) : \mathrm{id}(B).\, e, E'\rangle, V', \kappa') \\
& = & [\![\mathrm{id}(A)]\!](V') \gtrsim \lambda V'.\langle e, E'[x \leftarrow V'], \kappa'\rangle \\
(\star) & = & \mathrm{Just}\, V' \gtrsim \lambda V'.\langle e, E'[x \leftarrow V'], \kappa'\rangle \\
& = & \langle e, E'[x \leftarrow V'], \kappa'\rangle
\end{aligned}$$

The step marked with $(\star)$ follows from the basic property of $\mathrm{id}(\cdot)$

$$[\![\mathrm{id}(T)]\!] = \mathrm{Just}$$

By definition, $\langle e, E[x \leftarrow V], \kappa\rangle \approx \langle e, E'[x \leftarrow V'], \kappa'\rangle$.

Thus $\langle e, E[x \leftarrow V], \kappa\rangle \approx^* \langle e, E'[x \leftarrow V'], \kappa'\rangle$.

Case $\langle U_1\langle S_1 \to T_1 \Rightarrow^l S \to T\rangle, E\rangle \approx \langle \lambda(x : \lceil S \Rightarrow^l S_1 \rceil; c) : (d; \lceil T_1 \Rightarrow^l T \rceil).\, e, E'\rangle$

Let $U_1' = \langle \lambda x^{c,d}.e, E'\rangle$, we also have $U_1 \approx U_1' : S_1 \to T_1$.

In this case,

$$\begin{array}{rl}
& \mathrm{doApp}(U, V, \kappa) \\
= & \mathrm{doApp}(\langle U_1\langle S_1 \to T_1 \Rightarrow^l S \to T\rangle, E\rangle, V, \kappa) \\
\longrightarrow & [\![S \Rightarrow^l S_1]\!](V) \gtrsim \lambda V_1.\langle V_1, [(U_1\,\square)][\square\langle T_1 \Rightarrow^l T\rangle]\kappa\rangle
\end{array}$$

$$\begin{array}{rrl}
& & \mathrm{doApp}'(U', V', k') \\
& = & \mathrm{doApp}'(\langle \lambda(x : \lceil C \Rightarrow^l A\rceil ; c) : (d ; \lceil B \Rightarrow^l D\rceil).\, e, E'\rangle, V', \kappa') \\
& = & [\![\lceil S \Rightarrow^l S_1\rceil ; c]\!](V') \gtrsim \lambda V_1'.\langle e, E'[x \leftarrow V_1'], \mathrm{extCont}(d ; \lceil T_1 \Rightarrow^l T\rceil, \kappa')\rangle \\
(\star) & = & ([\![d]\!] \circ [\![\lceil S \Rightarrow^l S_1\rceil]\!])(V') \gtrsim \lambda V_1'.\langle e, E'[x \leftarrow V_1'], \mathrm{extCont}(d ; \lceil T_1 \Rightarrow^l T\rceil, \kappa')\rangle \\
& = & [\![\lceil S \Rightarrow^l S_1\rceil]\!](V') \gtrsim [\![d]\!] \gtrsim \lambda V_1'.\langle e, E'[x \leftarrow V_1'], \mathrm{extCont}(d, \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa'))\rangle
\end{array}$$

The step marked with $(\star)$ follows from the basic property of $\mathrm{seq}(\cdot, \cdot)$

$$[\![c ; d]\!] = [\![d]\!] \circ [\![c]\!]$$

By Lemma 3, we can case-split $[\![S \Rightarrow^l S_1]\!](V)$ and $[\![\lceil S \Rightarrow^l S_1\rceil]\!](V')$ at the same time.

Case $[\![S \Rightarrow^l S_1]\!](V) = \mathrm{Error}\, l$ and $[\![\lceil S \Rightarrow^l S_1\rceil]\!](V') = \mathrm{Error}\, l$

In this case

$$\begin{array}{rl}
& \mathrm{doApp}(U, V, \kappa) \\
\longrightarrow & \mathrm{Error}\, l \gtrsim \lambda V_1.\langle V_1, [(U_1\,\square)][\square\langle T_1 \Rightarrow^l T\rangle]\kappa\rangle \\
\longrightarrow & \mathrm{Error}\, l
\end{array}$$

$$\begin{array}{rl}
& \mathrm{doApp}'(U', V', \kappa') \\
= & \mathrm{Error}\, l \gtrsim [\![c]\!] \gtrsim \lambda V_1'.\langle e, E'[x \leftarrow V_1'], \mathrm{extCont}(d ; \lceil T_1 \Rightarrow^l T\rceil, \kappa')\rangle \\
= & \mathrm{Error}\, l
\end{array}$$

By definition, $\mathrm{Error}\, l \approx \mathrm{Error}\, l$.

Thus $\mathrm{doApp}(U, V, \kappa) \approx^* \mathrm{doApp}'(U', V', \kappa')$.

Case $[\![S \Rightarrow^l S_1]\!](V) = \mathrm{Just}\, V_1$ and $[\![\lceil S \Rightarrow^l S_1\rceil]\!](V') = \mathrm{Just}\, V_1'$ and $V_1 \approx V_1' : S_1$

In this case

$$\begin{array}{rl}
& \mathrm{doApp}(U, V, \kappa) \\
\longrightarrow & \mathrm{Just}\, V_1 \gtrsim \lambda V_1.\langle V_1, [(U_1\,\square)][\square\langle T_1 \Rightarrow^l T\rangle]\kappa\rangle \\
\longrightarrow & \langle V_1, [(U_1\,\square)][\square\langle T_1 \Rightarrow^l T\rangle]\kappa\rangle \\
\longrightarrow & \mathrm{doApp}(U_1, V_1, [\square\langle T_1 \Rightarrow^l T\rangle]\kappa)
\end{array}$$

$$\begin{array}{rrl}
& & \mathrm{doApp}'(U', V', \kappa') \\
& = & \mathrm{Just}\, V_1' \gtrsim [\![c]\!] \gtrsim \lambda V_2'.\langle e, E'[x \leftarrow V_2'], \mathrm{extCont}(d ; \lceil T_1 \Rightarrow^l T\rceil, \kappa')\rangle \\
& = & [\![c]\!](V_1') \gtrsim \lambda V_2'.\langle e, E'[x \leftarrow V_2'], \mathrm{extCont}(d ; \lceil T_1 \Rightarrow^l T\rceil, \kappa')\rangle \\
(\star) & = & [\![c]\!](V_1') \gtrsim \lambda V_2'.\langle e, E'[x \leftarrow V_2'], \mathrm{extCont}(d, \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa'))\rangle \\
& = & \mathrm{doApp}'(\langle \lambda(x : c) : d.\, e, E'\rangle, V_1', \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa')) \\
& = & \mathrm{doApp}'(U_1', V_1', \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa'))
\end{array}$$

The step marked with $(\star)$ follows from Lemma 5.

Because $U_1 \approx U_1'$ and $V_1 \approx V_1'$ and, by Lemma 6, $[\Box\langle T_1 \Rightarrow^l T\rangle]\kappa \approx \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa')$, now we can use the induction hypothesis too show that

$$\mathrm{doApp}(U_1, V_1, [\Box\langle T_1 \Rightarrow^l T\rangle]\kappa) \approx^* \mathrm{doApp}'(U_1', V_1', \mathrm{extCont}(\lceil T_1 \Rightarrow^l T\rceil, \kappa'))$$

Thus $\mathrm{doApp}(U, V, \kappa) \approx^* \mathrm{doApp}'(U', V', \kappa')$.

$\square$