Equivalence of Expressions

Operational equivalence \} \quad \text{Observational Equivalence}

Testing equivalence \}

Two program expressions are said to be equivalent if they behave the same when placed in any observing context.

An observing context is a complete program with a hole, such that all free variables in expressions being evaluated become captured, when placed in the hole.
A new event primitive operator is introduced.

The $\rightarrow$ reduction relation is extended:

\[
\langle e; a \rangle \\
\ll \alpha, [R[event()]]a \mid M \gg^x \\
\rightarrow \ll \alpha, [R[nid]]a \mid M \gg^x
\]

An observing configuration is one of the form:

\[
\ll \alpha, [C]a \mid M \gg
\]

where $C$ is a hole-containing expression, or context.
Observations

Let $K$ be a configuration of the extended language, and let $\pi = [k_i \xrightarrow{l_i} k_{i+1} | i < \infty]$ be a fair path, i.e. $\pi \in F(k)$. Define:

$$\text{obs}(\pi) = \begin{cases} \mathsf{S} & \text{if } (\exists i < \infty, a)(l_i = \langle e : a \rangle) \\ \mathsf{F} & \text{otherwise} \end{cases}$$

$$\text{Obs}(\pi) = \begin{cases} \mathsf{S} & \text{if } (\forall \pi \in F(k)) (\text{obs}(\pi) = \mathsf{S}) \\ \mathsf{F} & \text{otherwise} \end{cases}$$
Three Equivalences

The natural equivalence is equal observations are made in all closing configuration contexts.

Other two equivalences (weaker) arise if sf observations are considered as good as s observations; or if sf observations are considered as bad as f observations.

Testing or convex or Plotkin or Egly-Milner
\[ e_0 \equiv_1 e_1 \text{ iff } \text{Obs}(O[e_0]) = \text{Obs}(O[e_1]). \]

Must or upper or Smyth
\[ e_0 \equiv_2 e_1 \text{ iff } \text{Obs}(O[e_0]) = s \iff \text{Obs}(O[e_1]) = s \]

May or lower or Hoare
\[ e_0 \equiv_3 e_1 \text{ iff } \text{Obs}(O[e_0]) = f \iff \text{Obs}(O[e_1]) = f \]
**Congruence**

\[ e_0 \equiv_i e_1 \Rightarrow c[e_0] \equiv_i c[e_1] \text{ for } i=1,2,3 \]

By construction, all equivalences defined are congruences.

**Partial Collapse**

\[
\begin{array}{c|ccc}
  & s & sf & f \\
\hline
s & \checkmark & x & x \\
sf & x & \checkmark & x \\
f & x & x & \checkmark \\
\end{array}
\]

\[
\begin{array}{c|ccc}
  & s & sf & f \\
\hline
s & \checkmark & x & x \\
sf & x & \checkmark & * \\
f & x & * & \checkmark \\
\end{array}
\]

\[\equiv_1\]

\[
\begin{array}{c|ccc}
  & s & sf & f \\
\hline
s & \checkmark & x & x \\
sf & \checkmark & \checkmark & x \\
f & x & x & \checkmark \\
\end{array}
\]

\[\equiv_2\]

\[\equiv_3\]

(1=2) \( e_0 \equiv_1 e_1 \) if \( e_0 \equiv_2 e_1 \) (due to fairness)

(1=3) \( e_0 \equiv_1 e_1 \) implies \( e_0 \equiv_3 e_1 \)