The Program Feature in FL

Alexander Aiken, John H. Williams, Edward L. Wimmers
IBM Research - Almaden Research Center
K53-803
650 Harry Road
San Jose, CA 95120-6099

October 16, 1990

1 Definition of prog

Recall the following definitions.

Definition 1.1

1. *An expression* e *is* FSEF *iff* $\exists e_0 \in \mathtt{C} \forall V \forall h(\mu(\mathtt{e}, V, h) = (e_0, h))$.
2. $\mathtt{f} \approx \mathtt{g}$ *iff* $\forall V \forall h(\mu(\mathtt{f}, V, h) = \mu(\mathtt{g}, V, h))$.

Throughout the rest of this paper, unless stated otherwise, f and g (possibly with subscripts) are assumed to be FSEF expressions in the core part of FL. Also, i (possibly with subscripts) is assumed to be a positive integer.

The next definition extends the set of core FL expressions by introducing the construct $\mathtt{Bind}_\mathtt{i}$. Intuitively, $\mathtt{Bind}_\mathtt{i}(\mathtt{e})$ binds the "variable" $\sigma_\mathtt{i}$ to the argument to $\mathtt{Bind}_\mathtt{i}(\mathtt{e})$ and then evaluates e.

Definition 1.2
If $\mathtt{e}_0$ *is an expression, then* $\mathtt{Bind}_\mathtt{i}(\mathtt{e}_0)$ *is also an expression whose meaning is given by:*

$$\mathtt{Bind}_\mathtt{i}(\mathtt{e}_0) \quad \approx \quad \lambda \mathtt{x}.(\quad (\mathtt{e}_0\mathtt{:void}) \quad \mathtt{where} \quad \{\mathtt{def}\ \sigma_\mathtt{i} \equiv \tilde{}\mathtt{x}\} \quad)$$

where the variable x *is chosen so that it is not free in* $\mathtt{e}_0$.

It is necessary to know which subscripts on σ occure freely in an expression. The next definition formalizes this.

Definition 1.3

$\mathtt{Free(e)} = \emptyset$ *if* $\mathtt{e}$ *is an atom or a function name other than* $\sigma_{\mathtt{i}}$ *for some* $\mathtt{i}$

$\mathtt{Free}(\sigma_{\mathtt{i}}) = \{\mathtt{i}\}$

$\mathtt{Free(Bind_i(e))} = \mathtt{Free(e)} - \{\mathtt{i}\}$

$\mathtt{Free(e_1{:}e_2)} = \mathtt{Free(e_1)} \cup \mathtt{Free(e_2)}$

$\mathtt{Free}(\langle \mathtt{e_1}, \ldots, \mathtt{e_n} \rangle) = \mathtt{Free(e_1)} \cup \ldots \cup \mathtt{Free(e_n)}$

Proposition 1.4

1. $\mathtt{Bind_i(g)} \approx \mathtt{Bind_i(g \circ \tilde{}void)}$
2. *if* $\mathtt{Free(f)} \cap \{\mathtt{i}\} = \emptyset$, *then* $\mathtt{f} \approx \mathtt{Bind_i}(\mathtt{f} \circ \sigma_{\mathtt{i}})$
3. *if* $\mathtt{Free(f)} \cap \{\mathtt{i}\} = \emptyset$, *then* $\mathtt{f} \circ \mathtt{Bind_i(g)} \approx \mathtt{Bind_i(f \circ g)}$

σ_i is in Bind$_i$( )

4. *if* $\mathtt{Free}(\mathtt{E}(\sigma_{\mathtt{i}} \circ \mathtt{f})) \cap \{\mathtt{i}\} = \emptyset$, *then* $\mathtt{E}(\sigma_{\mathtt{i}} \circ \mathtt{f}) \approx \mathtt{E}(\sigma_{\mathtt{i}} \circ \mathrm{dom}{:}\,\mathtt{f})$
5. *if* $\mathtt{Free(f)} \cap \{\mathtt{i}\} = \emptyset$, *then* $\mathtt{Bind_i(f)} \approx \mathtt{f} \circ \mathtt{\tilde{}void}$

Definition 1.5

$$\mathtt{prog}(\mathtt{i_0}, (\mathtt{i_1}, \mathtt{e_1}), \ldots, (\mathtt{i_j}, \mathtt{e_j}), \mathtt{e}) =$$
$$\mathtt{Bind_{i_0}}(\mathtt{Bind_{i_1}}(\ldots \mathtt{Bind_{i_{j-1}}}(\mathtt{Bind_{i_j}}(\mathtt{e}) \circ \mathtt{e_j}) \circ \mathtt{e_{j-1}} \ldots) \circ \mathtt{e_1})$$

Proposition 1.6
Let $\mathtt{i_0}, \mathtt{i_1}, \ldots, \mathtt{i_n}, \mathtt{I}$ *be distinct positive integers. Then*

1. *if* $\mathtt{Free(f_j)} \cap \{\mathtt{i_0}, \ldots, \mathtt{i_{j-1}}\} = \emptyset$ *for* $\mathtt{j} = 1, \ldots, \mathtt{n}$,
   then $[\mathtt{f_1}, \ldots, \mathtt{f_n}] \approx \mathtt{prog}(\mathtt{i_0}, (\mathtt{i_1}, \mathtt{f_1} \circ \sigma_{\mathtt{i_0}}), \ldots, (\mathtt{i_n}, \mathtt{f_n} \circ \sigma_{\mathtt{i_0}}), [\sigma_{\mathtt{i_1}}, \ldots, \sigma_{\mathtt{i_n}}])$
2. *if* $\mathtt{Free(f)} \cap \{\mathtt{i_0}, \ldots, \mathtt{i_n}\} = \emptyset$,
   then $\mathtt{f} \circ \mathtt{prog}(\mathtt{i_0}, (\mathtt{i_1}, \mathtt{g_1}), \ldots, (\mathtt{i_n}, \mathtt{g_n}), \mathtt{g}) \approx \mathtt{prog}(\mathtt{i_0}, (\mathtt{i_1}, \mathtt{g_1}), \ldots, (\mathtt{i_n}, \mathtt{g_n}), \mathtt{f} \circ \mathtt{g})$
3. *if* $\mathtt{Free(g)} \cap \{\mathtt{I}\} = \emptyset$ *and* $\mathtt{Free(g_j)} \cap \{\mathtt{I}\} = \emptyset$ *for* $\mathtt{j} = 0, \ldots, \mathtt{n}$,
   then $\mathtt{prog}(\mathtt{i_0}, (\mathtt{i_1}, \mathtt{g_1}), \ldots, (\mathtt{i_n}, \mathtt{g_n}), \mathtt{g}) \circ \mathtt{g_0} \approx \mathtt{prog}(\mathtt{I}, (\mathtt{i_0}, \mathtt{g_0} \circ \sigma_{\mathtt{I}}), (\mathtt{i_1}, \mathtt{g_1}), \ldots, (\mathtt{i_n}, \mathtt{g_n}), \mathtt{g})$

Definition 1.7

$\mathtt{forb(i, f, g)} = \mathrm{aa}{:}\,(\mathtt{Bind_i(f)}) \circ \mathtt{intsto} \circ \mathtt{g}$

Proposition 1.8
If $\mathtt{Free(f)} \cap \{\mathtt{i}\} = \emptyset$, *then* $\mathtt{for}{:}\,\langle \mathtt{f}, \mathtt{g} \rangle \circ \mathtt{\tilde{}void} \approx \mathtt{forb}(\mathtt{i}, \mathtt{f} \circ [\sigma_{\mathtt{i}}, \mathtt{\tilde{}void}], \mathtt{g}) \circ \mathtt{\tilde{}void}$.

Example 1.9

Consider the example $\mathtt{distl_{safe}} \circ [\mathtt{f}, \mathtt{g}]$. *Then the following all have the same meaning:*

1. $\mathtt{distl} \circ [\mathtt{f}, \mathtt{g}]$
2. $\mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]], \mathtt{len} \circ \mathtt{s2}\rangle \circ [\mathtt{f}, \mathtt{g}]$
3. $\mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]], \mathtt{len} \circ \mathtt{s2}\rangle \circ \mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), [\sigma_1, \sigma_2])$
4. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]], \mathtt{len} \circ \mathtt{s2}\rangle \circ [\sigma_1, \sigma_2])$
5. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]] \circ [\mathtt{s1}, [\sigma_1, \sigma_2] \circ \mathtt{s2}], \mathtt{len} \circ \mathtt{s2} \circ [\sigma_1, \sigma_2]\rangle)$
6. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]] \circ [\mathtt{s1}, [\sigma_1 \circ \mathtt{s2}, \sigma_2 \circ \mathtt{s2}]], \mathtt{len} \circ \sigma_2\rangle)$
7. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{for}{:}\langle[\mathtt{s1} \circ \mathtt{s2}, \mathtt{sel} \circ [\mathtt{s1}, \mathtt{s2} \circ \mathtt{s2}]] \circ [\mathtt{s1}, [\sigma_1, \sigma_2]], \mathtt{len} \circ \sigma_2\rangle)$
8. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{for}{:}\langle[\sigma_1, \mathtt{sel} \circ [\mathtt{s1}, \sigma_2]], \mathtt{len} \circ \sigma_2\rangle)$
9. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{forb}(3, [\sigma_1, \mathtt{sel} \circ [\mathtt{s1}, \sigma_2]] \circ [\sigma_3, \tilde{\ }\mathtt{void}], \mathtt{len} \circ \sigma_2))$
10. $\mathtt{prog}(0, (1, \mathtt{f} \circ \sigma_0), (2, \mathtt{g} \circ \sigma_0), \mathtt{forb}(3, [\sigma_1, \mathtt{sel} \circ [\sigma_3, \sigma_2]], \mathtt{len} \circ \sigma_2))$