Assignment #3

Diego Oniarti - 257835

1 Polymorphic behaviour

Prove that for any closed term f of type $\forall\alpha.\forall\beta.\alpha \to (\alpha \uplus \beta)$ and for any closed types τ_1, τ_2 value $v : \tau_1$, we have $f\ \tau_1\ \tau_2\ v \rightsquigarrow^* \ \mathit{inl}\ v$

Assuming:

$$
\begin{array}{ll}
\Delta, \Gamma \vDash f : \forall\alpha.\forall\beta.\alpha \to (\alpha \uplus \beta) & HP1 \\
\Delta \vdash \tau_1 & HP2 \\
\Delta \vdash \tau_2 & HP3 \\
\vdash v : \tau_1 & HP4
\end{array}
$$

Prove:

$$
f\ \tau_1\ \tau_2\ v \rightsquigarrow^* \ \mathit{inl}\ v \qquad\qquad THS
$$

$$
\begin{array}{rl}
\text{by HP1 w.h.} & \forall\delta \in D[\Delta], \forall\gamma \in G[\Gamma]^{\delta}, f\gamma\delta \in \mathcal{E}[\forall\alpha.\forall\beta.\alpha \to (\alpha \uplus \beta)]^{\delta} \\
\text{by def. of } \mathcal{E} \text{ w.h.} & \exists v'.f\gamma\delta \rightsquigarrow^* v' \text{ and } v' \in \mathcal{V}[\forall\alpha.\forall\beta.\alpha \to (\alpha \uplus \beta)]^{\delta} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & v' \equiv \Lambda\alpha.t \text{ and } \forall\tau'.\forall S \in \text{semty}(\tau').t[\tau'/\alpha] \in \mathcal{E}[\forall\beta.\alpha \to (\alpha \uplus \beta)]^{\substack{\delta, \\ \alpha\mapsto\tau', S}} \\
& \quad \text{pick } \tau' = \tau_1, S = \{v\} \\
\text{by def. of } \mathcal{E} \text{ w.h.} & \exists v''.t[\tau'/\alpha] \rightsquigarrow^* v'' \text{ and } v'' \in \mathcal{V}[\forall\beta.\alpha \to (\alpha \uplus \beta)]^{\substack{\delta, \\ \alpha\mapsto\tau_1, \{v\}}} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & v'' \equiv \Lambda\beta.t' \text{ and } \forall\tau''.\forall S' \in \text{semty}(\tau'').t'[\tau''/\beta] \in \mathcal{E}[\alpha \to (\alpha \uplus \beta)]^{\overbrace{\substack{\delta, \\ \alpha\mapsto\tau', S, \\ \beta\mapsto\tau'', S'}}^{\delta'}} \\
& \quad \text{pick } \beta = \tau_2, S' = \emptyset \\
\text{by def. of } \mathcal{E} \text{ w.h.} & \exists v'''.t'[\tau''/\beta] \rightsquigarrow^* v''' \text{ and } v''' \in \mathcal{V}[\alpha \to (\alpha \uplus \beta)]^{\delta'} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & v''' \equiv \lambda x : \alpha.t'' \text{ and } \forall v'''' \in \mathcal{V}[\alpha]^{\delta'}.t''[v''''/x] \in \mathcal{E}[\alpha \uplus \beta]^{\delta'} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & v''' \equiv \lambda x : \alpha.t'' \text{ and } \forall v'''' \in \{v\}.t''[v''''/x] \in \mathcal{E}[\alpha \uplus \beta]^{\delta'} \\
\text{by } v'''' = v \text{ w.h.} & v''' \equiv \lambda x : \alpha.t'' \text{ and } t''[v/x] \in \mathcal{E}[\alpha \uplus \beta]^{\delta'} \\
\text{by def. of } \mathcal{E} \text{ w.h.} & \exists v'''''.t''[v/x] \rightsquigarrow^* v''''' \text{ and } v''''' \in \mathcal{V}[\alpha \uplus \beta]^{\delta'} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & \begin{array}{rl} \text{either} & v''''' \equiv \ \mathit{inl}\ v'''''' \text{ and } v'''''' \in \mathcal{V}[\alpha]^{\delta'} \\ \text{or} & v''''' \equiv \ \mathit{inr}\ v'''''' \text{ and } v'''''' \in \mathcal{V}[\beta]^{\delta'} \end{array} \\
\text{by def. of } \mathcal{V} \text{ w.h.} & \begin{array}{rl} \text{either} & v''''' \equiv \ \mathit{inl}\ v'''''' \text{ and } v'''''' \in \{v\} \\ \text{or} & \cancel{v''''' \equiv \ \mathit{inr}\ v'''''' \text{ and } v'''''' \in \emptyset} \end{array} \\
\text{by no val in } \emptyset \text{ w.h.} & v'''' \equiv \ \mathit{inl}\ v \square
\end{array}
$$

2 Free Theorems

3 A Register Machine Language

$$
\begin{aligned}
t ::= & r := n \\
& |sum\ r\ r \\
& |sub\ r\ r \\
& |cmp\ r\ r \\
& |jmp\ r \\
& |jiz\ r \\
& |jeq\ r \\
r ::= & ar|br|cr|dr|er|fr|gr|hr \\
& |ir|jr|kr|lr|mr|nr|or \\
C ::= & \emptyset|C, \mathbb{N} \mapsto t \\
F ::= & \emptyset|F, \mathbb{N} \mapsto b \\
R ::= & \emptyset|R, r \mapsto \mathbb{N}
\end{aligned}
$$

Judgement:

$$n; C; R; F \rightrightarrows n; C; R; F$$

Codebase, registers, and flags:

$$\frac{C = C', n \mapsto t}{C(n) = t} \qquad \frac{R = R', r \mapsto n}{R(r) = n} \qquad \frac{F = F', n \mapsto B}{F(n) = B}$$

$$\frac{C = C', n' \mapsto \_ \quad C'(n) = t}{C(n) = t} \qquad \frac{R = R', r' \mapsto \_ \quad R'(r) = n}{R(r) = n} \qquad \frac{F = F', n' \mapsto \_ \quad F'(n) = B}{F(n) = B}$$

Rules:

$$\frac{C(n) = r_1 := n_1 \quad R' = R, r_1 \mapsto n_1 \quad n' = n + 1}{n; C; R; F \rightrightarrows n'; C; R'; F}\text{load}$$

$$\frac{C(n) = sum\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ R' = R, r_1 \mapsto n_1 + n_2\ n' = n + 1}{n; C; R; F \rightrightarrows n'; C; R'; F}\text{sum}$$

$$\frac{C(n) = sub\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ R' = R, r_1 \mapsto n_1 - n_2\ n' = n + 1\ F' = F, 0 \mapsto n_2 > n_1}{n; C; R; F \rightrightarrows n'; C; R'; F'}\text{sub}$$

$$\frac{C(n) = cmp\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ F' = F, 1 \mapsto n_1 == n_2\ n' = n + 1}{n; C; R; F \rightrightarrows n'; C; R; F'}\text{cmp}$$

$$\frac{C(n) = jmp\ r_1 \quad R(r_1) = n_1 \quad n' = n_1}{n; C; R; F \rightrightarrows n'; C; R; F}\text{jmp}$$

$$\frac{C(n) = jiz\ r_1 \quad R(r_1) = n_1 \quad F(1) = b \quad n' = \text{ if } b \text{ then } n_1 \text{ else } n + 1}{n; C; R; F \rightrightarrows n'; C; R; F}\text{jiz}$$

$$\frac{C(n) = jeq\ r_1 \quad R(r_1) = n_1 \quad F(0) = b \quad n' = \text{ if } b \text{ then } n_1 \text{ else } n + 1}{n; C; R; F \rightrightarrows n'; C; R; F}\text{jeq}$$

4 From the Register Machine to an Assembly Language

$$
\begin{aligned}
t ::=& r := n \\
&|sum\ r\ r \\
&|sub\ r\ r \\
&|cmp\ r\ r \\
&|jmp\ r \\
&|jiz\ r \\
&|jeq\ r \\
&|read\ r\ r \\
&|write\ r\ r \\
r ::=& ar|br|cr|dr|er|fr|gr|hr \\
&|ir|jr|kr|lr|mr|nr|or \\
C ::=& \emptyset|C, \mathbb{N} \mapsto t \\
F ::=& \emptyset|F, \mathbb{N} \mapsto b \\
R ::=& \emptyset|R, r \mapsto \mathbb{N} \\
M ::=& \emptyset|M, \mathbb{N} \mapsto \mathbb{N}
\end{aligned}
$$

Judgement:

$$n; C; R; F; M \rightrightarrows n; C; R; F; M$$

Codebase, registers, flags, and memory:

$$\frac{C = C', n \mapsto t}{C(n) = t} \qquad \frac{R = R', r \mapsto n}{R(r) = n} \qquad \frac{F = F', n \mapsto B}{F(n) = B}$$

$$\frac{C = C', n' \mapsto \_ \quad C'(n) = t}{C(n) = t} \qquad \frac{R = R', r' \mapsto \_ \quad R'(r) = n}{R(r) = n} \qquad \frac{F = F', n' \mapsto \_ \quad F'(n) = B}{F(n) = B}$$

$$\frac{M = M', n \mapsto n'}{M(n) = n'} \qquad \frac{M = M', n" \mapsto \_ \quad M'(n) = n'}{M(n) = n'}$$

Rules:

$$\frac{C(n) = r_1 := n_1 \quad R' = R, r_1 \mapsto n_1 \quad n' = n + 1}{n; C; R; F; M \rightrightarrows n'; C; R'; F; M}\text{load}$$

$$\frac{C(n) = sum\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ R' = R, r_1 \mapsto n_1 + n_2\ n' = n + 1}{n; C; R; F; M \rightrightarrows n'; C; R'; F; M}\text{sum}$$

$$\frac{C(n) = sub\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ R' = R, r_1 \mapsto n_1 - n_2\ n' = n + 1\ F' = F, 0 \mapsto n_2 > n_1}{n; C; R; F; M \rightrightarrows n'; C; R'; F'; M}\text{sub}$$

$$\frac{C(n) = cmp\ r_1\ r_2\ R(r_1) = n_1\ R(r_2) = n_2\ F' = F, 1 \mapsto n_1 == n_2\ n' = n + 1}{n; C; R; F; M \rightrightarrows n'; C; R; F'; M}\text{cmp}$$

$$\frac{C(n) = jmp\ r_1 \quad R(r_1) = n_1 \quad n' = n_1}{n; C; R; F; M \rightrightarrows n'; C; R; F; M}\text{jmp}$$

$$\frac{C(n) = jiz\ r_1 \quad R(r_1) = n_1 \quad F(1) = b \quad n' = \text{ if } b \text{ then } n_1 \text{ else } n + 1}{n; C; R; F; M \rightrightarrows n'; C; R; F; M}\text{jiz}$$

$$\frac{C(n) = jeq\ r_1 \quad R(r_1) = n_1 \quad F(0) = b \quad n' = \text{ if } b \text{ then } n_1 \text{ else } n + 1}{n; C; R; F; M \rightrightarrows n'; C; R; F; M}\text{jeq}$$

$$\frac{C(n) = read\ r_1\ r_2 \quad R(r_2) = n_2 \quad M(n_2) = n" \quad R' = R, r_1 \mapsto n" \quad n' = n + 1}{n; C; R; F; M \rightrightarrows n'; C; R'; F; M}\text{read}$$

$$\frac{C(n) = write\ r_1\ r_2 \quad R(r_1) = n_1 \quad R(r_2) = n_2 \quad M' = M, n_1 \mapsto n_2 \quad n' = n + 1}{n; C; R; F; M \rightrightarrows n'; C; R; F; M'}\text{write}$$