Regulile de deducție big-step

$$\frac{}{(\mathbf{skip}, \sigma) \Downarrow \sigma}$$

$$\frac{}{(X := a, \sigma) \Downarrow \sigma_{X \mapsto e_\sigma(a)}} \qquad \frac{(c_0, \sigma) \Downarrow \sigma'' \quad (c_1, \sigma'') \Downarrow \sigma'}{(c_0; c_1, \sigma) \Downarrow \sigma'}$$

$$\frac{e_\sigma(b) = 1 \quad (c_0, \sigma) \Downarrow \sigma'}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma) \Downarrow \sigma'} \qquad \frac{e_\sigma(b) = 0 \quad (c_1, \sigma) \Downarrow \sigma'}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma) \Downarrow \sigma'}$$

$$\frac{e_\sigma(b) = 0}{(\mathbf{while}\ b\ \mathbf{do}\ c, \sigma) \Downarrow \sigma}$$

$$\frac{e_\sigma(b) = 1 \quad (c, \sigma) \Downarrow \sigma'' \quad (\mathbf{while}\ b\ \mathbf{do}\ c, \sigma'') \Downarrow \sigma'}{(\mathbf{while}\ b\ \mathbf{do}\ c, \sigma) \Downarrow \sigma'}$$

Regulile de deducție small-step

$$\frac{}{(X := a, \sigma) \to (\mathbf{skip}, \sigma_{X \mapsto e_\sigma(a)})}$$

$$\frac{}{(\mathbf{skip}; c_1, \sigma) \to (c_1, \sigma)} \qquad \frac{(c_0, \sigma) \to (c_0', \sigma')}{(c_0; c_1, \sigma) \to (c_0'; c_1, \sigma')}$$

$$\frac{e_\sigma(b) = 1}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma) \to (c_0, \sigma)}$$

$$\frac{e_\sigma(b) = 0}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma) \to (c_1, \sigma)}$$

$$\frac{}{(\mathbf{while}\ b\ \mathbf{do}\ c, \sigma) \to (\mathbf{if}\ b\ \mathbf{then}\ (c; (\mathbf{while}\ b\ \mathbf{do}\ c))\ \mathbf{else\ skip}, \sigma)}$$

Regulile de deducție Hoare

$$\frac{}{\{A\}\textbf{skip}\{A\}} \qquad \frac{}{\{B[X:=a]\}X:=a\{B\}}$$

$$\frac{\{A\}c_0\{C\} \quad \{C\}c_1\{B\}}{\{A\}c_0;c_1\{B\}} \qquad \frac{\{A\wedge b\}c_0\{B\} \quad \{A\wedge\neg b\}c_1\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1\{B\}}$$

$$\frac{\{A\wedge b\}c\{A\}}{\{A\}\textbf{while } b \textbf{ do } c\{A\wedge\neg b\}} \qquad \frac{\models A\to A' \quad \{A'\}c\{B'\} \quad \models B'\to B}{\{A\}c\{B\}}$$

Notăm faptul că o aserțiune $\{A\}c\{B\}$ poate fi dedusă prin aceste reguli cu $\vdash \{A\}c\{B\}$. Observăm și că ultima regulă, numită **regula consecinței**, este de fapt o **pseudo-regulă**, având în vedere că nu este decidabil dacă o aplicare a ei este corectă (de ce?).

Algoritm de unificare

Vom descrie acum un algoritm de unificare. El are ca intrare o mulțime de ecuații $\mathcal{E}$, iar la ieșire, un cgu dat de o mulțime de ecuații ca în propoziția precedentă, sau „eșec", dacă un cgu nu există. Algoritmul trece printr-un ciclu care, la fiecare pas, verifică dacă există vreo ecuație în mulțime căruia i se poate aplica vreuna dintre următoarele operații – în caz contrar, el se oprește:

- o ecuație de forma $f(s_1,\ldots,s_n)=f(t_1,\ldots,t_n)$ este înlocuită de ecuațiile $s_1=t_1,\ldots,s_n=t_n$;
- o ecuație de forma $f(s_1,\ldots,s_n)=g(t_1,\ldots,t_m)$, unde $f\neq g$, produce „eșec";
- o ecuație de forma $x=x$, cu $x\in V$, este eliminată;
- o ecuație de forma $t=x$ cu $x\in V$ și $t\notin V$ este înlocuită cu $x=t$;
- o ecuație de forma $x=t$ cu $x\in V$, $t\neq x$ și $x\in Var(t)$ produce „eșec";
- o ecuație de forma $x=t$ cu $x\in V$, $t\neq x$, $x\notin Var(t)$, dar cu x apărând și în alte ecuații din mulțime, face ca toate celelalte sale apariții să fie substituite cu t.

Reguli de deducție

Regulile de deducție vor fi următoarele:

$$\overline{\Gamma \cup \{x : \sigma\} \vdash x : \sigma}$$

$$\frac{\Gamma \cup \{x : \sigma\} \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

Încercați să deduceți tipul termenului „de mai devreme" $(\lambda z.(\lambda u.z))(yx)$ folosind aceste reguli. Acest exemplu ne arată că, într-un anume fel (în ce fel? detaliați!), cele două moduri de a gândi tipurile sunt echivalente.

Finalizare

Definim:

$$\begin{aligned} c(x, \Gamma \cup \{x : \tau\}, Z) &:= \{\tau = Z\} \\ c(\lambda x : \sigma.M, \Gamma, Z) &:= c(M, \Gamma \cup \{x : \sigma\}, W) \cup \{Z = \sigma \to W\} \\ c(MN, \Gamma, Z) &:= c(M, \Gamma, W_1) \cup c(M, \Gamma, W_2) \cup \{W_1 = W_2 \to Z\} \end{aligned}$$

Algoritmul va face apelul $c(M', \Gamma_M, Z)$ și va obține o mulțime de ecuații. Pentru ea, se caută un cgu θ. În caz că nu există, se returnează „eșec". În caz de succes, rezultatul algoritmului va fi $\widetilde{\theta}(\Gamma_M) \vdash M : \widetilde{\theta}(Z)$ (unde $\widetilde{\theta}$ are semnificația firească).

Putem rula algoritmul pe termenii $(\lambda z.(\lambda u.z))(yx)$ și xx, evidențiați mai devreme.

Alocarea tipurilor

Acestea sunt regulile de deducție pentru alocarea de tipuri termenilor:

$$\frac{}{\Gamma \cup \{x : \sigma\} \vdash x : \sigma}$$

$$\frac{\Gamma \cup \{x : \sigma\} \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$\frac{}{\Gamma \vdash z : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash s(M) : \mathbb{N}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \quad \Gamma \vdash N : \tau \quad \Gamma \cup \{w : \mathbb{N}\} \vdash P : \tau}{\Gamma \vdash \mathit{ifz}(M, N, w, P) : \tau} \qquad \frac{\Gamma \cup \{w : \tau\} \vdash M : \tau}{\Gamma \vdash \mathit{fix}(w, M) : \tau}$$

Reguli de evaluare

Următoarele reguli stabilesc evaluarea termenilor în regim *eager*:

$$\frac{M \to M'}{s(M) \to s(M')} \qquad \frac{M \to M'}{\mathit{ifz}(M, N, w, P) \to \mathit{ifz}(M', N, w, P)}$$

$$\frac{}{\mathit{ifz}(z, N, w, P) \to N} \qquad \frac{\mathit{val}(s(M))}{\mathit{ifz}(s(M), N, w, P) \to P[w := M]}$$

$$\frac{M \to M'}{MN \to M'N} \qquad \frac{\mathit{val}(M) \quad N \to N'}{MN \to MN'}$$

$$\frac{\mathit{val}(N)}{(\lambda x.M)N \to M[x := N]} \qquad \frac{}{\mathit{fix}(w, M) \to M[w := \mathit{fix}(w, M)]}$$