

INF889J Semaine 3

1 α -équivalence et substitution

PROBLÈME 1. Soit la grammaire abstraite suivante

$$M, N, L ::= x \mid k() \mid r(M; N) \mid s(x.M; x, y.N)$$

Soulignez les variables liées et encerclez les variables libres des arbres suivants :

- $r(x; y)$
- $s(z.x; w, y.r(x; y))$
- $r(z; s(x.x; h, y.s(z.h; y, x.r(h; x))))$

Calculez :

- $[k()/x](r(x; y))$
- $[r(x; y)/x](s(z.x; w, y.r(x; y)))$
- $[s(x.x; z, h.r(z; h))/x](r(z; s(x.x; h, y.s(z.h; y, x.r(h; x))))$

PROBLÈME 2 (COMMUTATIVITÉ). Soit la grammaire abstraite suivante :

$$M, N, L ::= x \mid f(M; N) \mid g(x.M)$$

Démontrez que si $x \neq y$ et si x n'est pas libre dans L , alors pour tout M, N et L

$$[L/y]([N/x]M) = [[L/y]N/x]([L/y]M).$$

L'hypothèse « x n'est pas libre dans L » est-elle vraiment requise ?

2 Opérateur de somme

L'APL (*A Programming Language*, 1966) est doté d'un opérateur de somme. Pour pratiquer les concepts vus en classe, vous êtes invités à ajouter un opérateur de somme au langage d'expressions de l'annexe A :

$$e ::= \dots \qquad \dots \\ \mid \text{sum}(e_1; e_2; x.e_3) \qquad \sum_{x=e_1}^{e_2} e_3$$

PROBLÈME 3. Donnez la ou les règles de typage pour ce nouvel opérateur, ainsi qu'une sémantique à petites étapes où les variables représentent des valeurs closes. N'oubliez pas que si $k > n$, alors $\sum_{x=k}^n f(x) = 0$.

PROBLÈME 4. Énoncez et démontrez les propriétés de sûreté communément appelées « le progrès » et « la préservation » pour le langage d'expressions de l'annexe A avec l'opérateur de somme.

Il se peut que vous ayez besoin de propriétés auxiliaires, telles que la propriété de substitution ou des formes canoniques. Si vous utilisez ces propriétés, donnez leurs énoncés. (Vous êtes encouragés à essayer de les démontrer.)

A Expressions

A.1 Grammaire

Où $n \in \mathbb{N}$, les arbres e de syntaxe abstraite avec liage pour notre langage d'expressions et les types τ sont donnés par les grammaires (abstraites et concrètes) suivantes :

$e ::= x$	x
$\text{num}[n]$	n
true	true
false	false
$\text{plus}(e_1; e_2)$	$e_1 + e_2$
$\text{lt}(e_1; e_2)$	$e_1 < e_2$
$\text{if}(e_1; e_2; e_3)$	if e_1 then e_2 else e_3
$\text{let}(x; e_1.e_2)$	let $x = e_1$ in e_2
$\tau ::= \text{int}$	int
bool	bool

A.2 Typage

$\boxed{\Gamma \vdash x : e}$ L'expression e a le type τ en supposant Γ

$$\begin{array}{c}
\frac{(x : \tau) \in \Gamma}{\Gamma \vdash x : \tau} \text{ (T-VAR)} \quad \frac{}{\Gamma \vdash \text{num}[n] : \text{int}} \text{ (T-NUM)} \\
\\
\frac{}{\Gamma \vdash \text{true} : \text{bool}} \text{ (T-TRUE)} \quad \frac{}{\Gamma \vdash \text{false} : \text{bool}} \text{ (T-FALSE)} \\
\\
\frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash \text{plus}(e_1; e_2) : \text{int}} \text{ (T-PLUS)} \quad \frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash \text{lt}(e_1; e_2) : \text{bool}} \text{ (T-LT)} \\
\\
\frac{\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \text{if}(e_1; e_2; e_3) : \tau} \text{ (T-IF)} \quad \frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \text{let}(e_1; x.e_2) : \tau_2} \text{ (T-LET)}
\end{array}$$

A.3 Sémantique opérationnelle : petites étapes

$\boxed{e \text{ val}}$ e est une valeur

$$\frac{}{\text{num}[n] \text{ val}} \text{ (VAL-NUM)} \quad \frac{}{\text{true} \text{ val}} \text{ (VAL-TRUE)} \quad \frac{}{\text{false} \text{ val}} \text{ (VAL-FALSE)}$$

$e_1 \mapsto e_2$ e_1 se réduit à e_2 en une étape

$$\begin{array}{c}
\overline{\text{plus}(\text{num}[m]; \text{num}[n]) \mapsto \text{num}[m+n]} \quad (\text{SOS-PLUS}) \\
\\
\frac{e_1 \mapsto e'_1}{\text{plus}(e_1; e_2) \mapsto \text{plus}(e'_1; e_2)} \quad (\text{SOS-PLUS-1}) \qquad \frac{e_1 \text{ val } e_2 \mapsto e'_2}{\text{plus}(e_1; e_2) \mapsto \text{plus}(e_1; e'_2)} \quad (\text{SOS-PLUS-2}) \\
\\
\frac{m < n}{\text{lt}(\text{num}[m]; \text{num}[n]) \mapsto \text{true}} \quad (\text{SOS-LT-TRUE}) \qquad \frac{\neg(m < n)}{\text{lt}(\text{num}[m]; \text{num}[n]) \mapsto \text{false}} \quad (\text{SOS-LT-FALSE}) \\
\\
\frac{e_1 \mapsto e'_1}{\text{lt}(e_1; e_2) \mapsto \text{lt}(e'_1; e_2)} \quad (\text{SOS-LT-1}) \qquad \frac{e_1 \text{ val } e_2 \mapsto e'_2}{\text{lt}(e_1; e_2) \mapsto \text{lt}(e_1; e'_2)} \quad (\text{SOS-LT-2}) \\
\\
\overline{\text{if}(\text{true}; e_1; e_2) \mapsto e_1} \quad (\text{SOS-IF-TRUE}) \qquad \overline{\text{if}(\text{false}; e_1; e_2) \mapsto e_2} \quad (\text{SOS-IF-FALSE}) \\
\\
\frac{e_0 \mapsto e'_0}{\text{if}(e_0; e_1; e_2) \mapsto \text{if}(e'_0; e_1; e_2)} \quad (\text{SOS-IF}) \\
\\
\frac{e_1 \text{ val}}{\text{let}(e_1; x.e_2) \mapsto [e_1/x]e_2} \quad (\text{SOS-LET}) \qquad \frac{e_1 \mapsto e'_1}{\text{let}(e_1; x.e_2) \mapsto \text{let}(e'_1; x.e_2)} \quad (\text{SOS-LET1})
\end{array}$$