Typage — arith. + logiques

$$\frac{}{\texttt{vrai} : B} \qquad \frac{}{\texttt{faux} : B} \qquad \frac{e : B}{\texttt{non}(e) : B}$$

$$\frac{e_1 : B \quad e_2 : B}{\texttt{ou}(e_1, e_2) : B} \qquad \frac{e_1 : B \quad e_2 : B}{\texttt{et}(e_1, e_2) : B}$$

$$\frac{}{\texttt{Cte}(n) : N} \qquad \frac{e_1 : N \quad e_2 : N}{\texttt{+}(e_1, e_2) : N} \qquad \frac{e_1 : N \quad e_2 : N}{\texttt{*}(e_1, e_2) : N}$$

$$\frac{e_1 : N \quad e_2 : N}{\texttt{eq}(e_1, e_2) : B} \qquad \frac{e_1 : N \quad e_2 : N}{\texttt{gt}(e_1, e_2) : B}$$

$$\frac{e_1 : B \quad e_2 : N \quad e_3 : N}{\texttt{if}(e_1, e_2, e_3) : N}$$

Sémantique — variables

Introduction de la construction « $\texttt{let}\ id = e_1\ \texttt{in}\ e_2$ »

Dans e_2 le nom id représente e_1

Ensemble $\mathcal{AV}$ inductif...

- $\to \texttt{Cte}(n) \qquad A_1, A_2 \to \texttt{+}(A_1, A_2) \qquad A_1, A_2 \to \texttt{*}(A_1, A_2)$
- $A_1, A_2 \to \texttt{let}(id, A_1, A_2)$ où id un nom $\qquad \to \texttt{Var}(id)$

(syntaxe abstraite)

Conserver (assurer) lien nom $\leftrightarrow$ expression ? $\rightsquigarrow$ environnement

Environnement de typage : liste de couples $id : t$ — ici tête à droite

Jugement de typage : $\Gamma \vdash e : t$

Sémantique — variables

Ensemble $\mathcal{AV}$ inductif...

- $\to \texttt{Cte}(n) \qquad A_1, A_2 \to \texttt{+}(A_1, A_2) \qquad A_1, A_2 \to \texttt{*}(A_1, A_2)$
- $A_1, A_2 \to \texttt{let}(id, A_1, A_2)$ où id un nom $\qquad \to \texttt{Var}(id)$

(syntaxe abstraite)

Environnement de typage : liste de couples $id : t$ — tête à droite $l \cdot e$

$$\frac{}{\Gamma \vdash \texttt{Var}(x) : N}\ x : N \in \Gamma \qquad \frac{}{\Gamma \vdash \texttt{Cte}(n) : N}$$

$$\frac{\Gamma \vdash e_1 : N \quad \Gamma \vdash e_2 : N}{\Gamma \vdash \texttt{+}(e_1, e_2) : N} \qquad \frac{\Gamma \vdash e_1 : N \quad \Gamma \vdash e_2 : N}{\Gamma \vdash \texttt{*}(e_1, e_2) : N}$$

$$\frac{\Gamma \vdash e_1 : N \quad \Gamma \cdot (x : N) \vdash e_2 : N}{\Gamma \vdash \texttt{let}(x, e_1, e_2) : N}$$

Sémantique — variables

Ensemble $\mathcal{AV}$ inductif...

- $\to \texttt{Cte}(n) \qquad A_1, A_2 \to \texttt{+}(A_1, A_2) \qquad A_1, A_2 \to \texttt{*}(A_1, A_2)$
- $A_1, A_2 \to \texttt{let}(id, A_1, A_2)$ où id un nom $\qquad \to \texttt{Var}(id)$

(syntaxe abstraite)

Environnement d'évaluation : liste de couples $id = v$ — ici tête à droite $l \cdot e$

$$\frac{}{\Gamma \vdash \texttt{Var}(x) \rightsquigarrow v}\ x = v \in \Gamma \qquad \frac{}{\Gamma \vdash \texttt{Cte}(n) \rightsquigarrow n}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{+}(e_1, e_2) \rightsquigarrow v_1 +_{\mathbb{N}} v_2} \qquad \frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{*}(e_1, e_2) \rightsquigarrow v_1 \times_{\mathbb{N}} v_2}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{let}(x, e_1, e_2) \rightsquigarrow v_2}$$

Sémantique variables

Fortement typé

Portée des définitions

$\texttt{let}(n, \texttt{Cte}(2), \texttt{+}(\texttt{let}(n, \texttt{+}(\texttt{Var}(n), \texttt{Cte}(2)), \texttt{Var}(n)), \texttt{Var}(n)))$?

$[\ ] \vdash \texttt{let}(n, \texttt{Cte}(2), \texttt{+}(\texttt{let}(n, \texttt{+}(\texttt{Var}(n), \texttt{Cte}(2)), \texttt{Var}(n)), \texttt{Var}(n))) \rightsquigarrow 6$

$\mathcal{EV}$ en exo.

Sémantique et les fonctions ?

(Pour simplifier : un argument seulement)

Types :

- $\to B$ $\quad \to N$ $\quad T_1, T_2 \to T_1 \to T_2$

Fonctions. . .

- $E_1, E_2 \to \texttt{letFun}(id_1, id_2, T_1, T_2, E_1, E_2)$ $\quad E \to \texttt{App}(id, E)$

$$\frac{\Gamma \vdash e : t_1}{\Gamma \vdash \texttt{App}(f, e) : t_2} \text{ si } f : t_1 \to t_2 \in \Gamma$$

$$\frac{\Gamma \cdot (x : t_1) \vdash e_1 : t_2 \quad \Gamma \cdot (f : t_1 \to t_2) \vdash e_2 : t_3}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) : t_3}$$

Sémantique et les fonctions ?

(Pour simplifier : un argument seulement)

Idée : associer nom et corps dans env.

Types :

- $\to B$ $\quad \to N$ $\quad T_1, T_2 \to T_1 \to T_2$

Fonctions. . .

- $E_1, E_2 \to \texttt{letFun}(id_1, id_2, T_1, T_2, E_1, E_2)$ $\quad E \to \texttt{App}(id, E)$

$$\frac{\Gamma \cdot (f = (x, e_1)) \vdash e_2 \rightsquigarrow v}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) \rightsquigarrow v}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{App}(f, e_1) \rightsquigarrow v_2} \text{ si } f = (x, e_2) \in \Gamma$$

Sémantique et les fonctions ?

$$\frac{\Gamma \cdot (f = (x, e_1)) \vdash e_2 \rightsquigarrow v}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) \rightsquigarrow v}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{App}(f, e_1) \rightsquigarrow v_2} \text{ si } f = (x, e_2) \in \Gamma$$

`let x=1 in let fun f(y:N):N = x+y in let x=2 in f(3)` ?

Sémantique — et les fonctions ?

$$\frac{\Gamma \cdot (f = (x, e_1)) \vdash e_2 \rightsquigarrow v}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) \rightsquigarrow v}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{App}(f, e_1) \rightsquigarrow v_2} \text{ si } f = (x, e_2) \in \Gamma$$

Dynamique ↑… et statique ↓

$$\frac{\Gamma \cdot (f = (x, e_1, \Gamma)) \vdash e_2 \rightsquigarrow v}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) \rightsquigarrow v}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma ++ \Delta \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{App}(f, e_1) \rightsquigarrow v_2} \text{ si } f = (x, e_2, \Delta) \in \Gamma$$

Sémantique — et les fonctions récursives ?

(ici un nouveau mot-clef)

Idée : changer la portée du nom de fonction

$$\frac{\Gamma \cdot (x : t_1) \vdash e_1 : t_2 \quad \Gamma \cdot (f : t_1 \to t_2) \vdash e_2 : t_3}{\Gamma \vdash \texttt{letFun}(f, x, t_1, t_2, e_1, e_2) : t_3}$$

$$\frac{\Gamma \cdot (x : t_1) \cdot (f : t_1 \to t_2) \vdash e_1 : t_2 \quad \Gamma \cdot (f : t_1 \to t_2) \vdash e_2 : t_3}{\Gamma \vdash \texttt{letFunRec}(f, x, t_1, t_2, e_1, e_2) : t_3}$$

Sémantique — et les fonctions récursives ?

(ici un nouveau mot-clef)

Idée : changer la portée du nom de fonction

$$\frac{\Gamma \cdot (f = (x, e_1, \Gamma)) \vdash e_2 \rightsquigarrow v}{\Gamma \vdash \texttt{letFunRec}(f, x, t_1, t_2, e_1, e_2) \rightsquigarrow v}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow v_1 \quad \Gamma ++ \Delta \cdot (x = v_1) \vdash e_2 \rightsquigarrow v_2}{\Gamma \vdash \texttt{App}(f, e_1) \rightsquigarrow v_2} \text{ si } f = (x, e_2, \Delta) \in \Gamma$$

La même chose…

Env. liste donc bon paramètre pour appel récursif !

Sémantique — langages fonctionnels

Ici : fonctions comme paramètres et résultats

Abstraction $E \to \texttt{fun}(id, T, E)$ ex. :`fun` $x : T$ `=>` corps

Application $E_1, E_2 \to \texttt{App}(E_1, E_2)$ ex. : `((delta q) c)`

$$\frac{\Gamma \cdot (x : t_1) \vdash e : t_2}{\Gamma \vdash \texttt{fun}(x, t_1, e) : t_1 \to t_2}$$

$$\frac{\Gamma \vdash e_1 : t_1 \to t_2 \quad \Gamma \vdash e_2 : t_1}{\Gamma \vdash \texttt{App}(e_1, e_2) : t_2}$$

Sémantique — langages fonctionnels

Ici : fonctions comme paramètres et résultats $\rightsquigarrow$ et donc curryfication

$$\frac{}{\Gamma \vdash \mathbf{fun}(x, t_1, e) \rightsquigarrow (x, e)}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow (x, c) \quad \Gamma \vdash e_2 \rightsquigarrow v_2 \quad \Gamma \cdot (x = v_2) \vdash c \rightsquigarrow v_3}{\Gamma \vdash \mathbf{App}(e_1, e_2) \rightsquigarrow v_3}$$

```
((fun x => (fun y => x * y)) 5) 4
```
?

Sémantique — langages fonctionnels

Ici : fonctions comme paramètres et résultats $\rightsquigarrow$ et donc curryfication

$$\frac{}{\Gamma \vdash \mathbf{fun}(x, t_1, e) \rightsquigarrow (x, e)}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow (x, c) \quad \Gamma \vdash e_2 \rightsquigarrow v_2 \quad \Gamma \cdot (x = v_2) \vdash c \rightsquigarrow v_3}{\Gamma \vdash \mathbf{App}(e_1, e_2) \rightsquigarrow v_3}$$

$$\frac{}{\Gamma \vdash \mathbf{fun}(x, t_1, e) \rightsquigarrow (x, e, \Gamma)}$$

$$\frac{\Gamma \vdash e_1 \rightsquigarrow (x, c, \Delta) \quad \Gamma \vdash e_2 \rightsquigarrow v_2 \quad \Gamma ++ \Delta \cdot (x = v_2) \vdash c \rightsquigarrow v_3}{\Gamma \vdash \mathbf{App}(e_1, e_2) \rightsquigarrow v_3}$$