Types linéaires.

Le but des types linéaires est de ***contrôler l'usage des ressources***. Avec un tel système de typage, on pourrait :

- ▷ désallouer une zone mémoire, fermer un canal/un fichier ; on se limite à un seul usage : les deux termes suivants ne sont pas typables
  - $(\texttt{free}\ x, \texttt{free}\ x)$
  - $(\lambda y.\ \lambda z.\ (\texttt{free}\ y, \texttt{free}\ z))\ x\ x$
- ▷ l'initialisation d'une zone mémoire ; au moins un usage
- ▷ si on sait au plus un usage, alors on peut désallouer après un usage.

On se donne la syntaxe suivante.

- ▷ Les booléens sont :

$$b ::= \texttt{true} \mid \texttt{false}.$$

- ▷ Les termes sont :

$$\begin{aligned} M, N ::= {} & \eta b \mid \eta(M, N) \mid \eta\lambda x.\ M \\ & \mid \texttt{if}\ b\ \texttt{then}\ M\ \texttt{else}\ N \mid \texttt{let}(x, y) = M\ \texttt{in}\ N \mid M\ N \\ & \mid x \mid \texttt{let}\ x = M\ \texttt{in}\ N. \end{aligned}$$

- ▷ Les usages sont :

$$\eta ::= \mathsf{lin} \mid \mathsf{un}.$$

Les *usages* représentent une utilisation *linéaire* (lin) ou non restrictive (un*restricted*). Un usage lin est utilisé exactement ***une*** fois.

On s'autorise le filtrage sur les couples au lieu de `fst` M et `snd` M car, dans le cas linéaire, on ne pourrait utiliser qu'une des deux composantes… zut !

On accepte, par exemple, $\mathsf{lin}(\texttt{lintrue}, \mathsf{lin}\texttt{false})$ ou $\mathsf{un}(\texttt{lintrue}, \texttt{unfalse})$.

Les types sont définis par la grammaire

$$T, U ::= \texttt{bool} \mid A \to B \mid A * B \qquad A, B ::= \eta\, T.$$

Les T, U sont des *prétypes*, les A, B sont les *types*.

La séparation dans la grammaire permet de rejeter une expression comme $\mathsf{un}(\mathsf{lin}\,\texttt{bool} * \mathsf{un}\,\texttt{bool})$.

On se donne les règles de typages pour cette version linéaire du λ-calcul.

Première tentative.

On propose la règle pour les couples linéaires :

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : \mathsf{lin}(A * B)}.$$

Cependant, on ne mets pas de restrictions sur A : on peut avoir $A = \mathsf{lin}T$ ou $A = \mathsf{un}T$… Zut !

Seconde tentative.

On propose un règle pour les couples

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B \qquad \eta(A) \qquad \eta(B)}{\Gamma \vdash (M, N) : \eta(A * B)},$$

où l'on définit le prédicat $\eta(A)$ par :

- ▷ si $\eta = \mathsf{un}$ alors A ne peut pas être $\mathsf{lin}T$;
- ▷ si $\eta = \mathsf{lin}$ alors A peut être quelconque.

Puis, on se donne une règle pour le filtrage

$$\frac{\Gamma \vdash M : \eta(A_1 * A_2) \qquad \Gamma, x : A_1, y : A_2 \vdash N : B}{\Gamma \vdash \texttt{let}(x, y) = M \ \texttt{in} \ N : B} \ .$$

Ici, η n'a pas d'importance. On peut donc typer

$$\texttt{let}(x, y) = \mathsf{un}(\mathsf{lin}\texttt{true}, \texttt{unfalse}) \ \texttt{in} \ \mathsf{lin}(x, x),$$

mais on ne veut pas pouvoir le faire !

Tentative finale.

Définition 1. On définit $\Gamma_1 \circ \Gamma_2$ est défini comme « $\Gamma_1 \cup \Gamma_2$ » à ceci près que :

- si $\Gamma_1(x) = \mathsf{lin}T$ alors $x \notin \mathrm{dom}(\Gamma_2)$;
- si $\Gamma_2(x) = \mathsf{lin}T$ alors $x \notin \mathrm{dom}(\Gamma_1)$;
- si $x \in \mathrm{dom}(\Gamma_1) \cap \mathrm{dom}(\Gamma_2)$ alors il existe T tel que $\Gamma_1(x) = \Gamma_2(x) = \mathsf{un}T$.

Les vraies règles de typage sont :

$$\frac{\Gamma_1 \vdash M : A \qquad \Gamma_2 \vdash N : B \qquad \eta(A) \qquad \eta(B)}{\Gamma_1 \circ \Gamma_2 \vdash (M, N) : \eta(A * B)}$$

$$\frac{\Gamma_1 \vdash M : \eta(A * B) \qquad \Gamma_2, x : A, y : B \vdash N : C}{\Gamma_1 \circ \Gamma_2 \vdash \texttt{let}(x, y) = M \ \texttt{in} \ N : C}$$

$$\frac{\mathsf{un}(\Gamma)}{\Gamma, x : A \vdash x : A} \qquad \frac{\mathsf{un}(\Gamma)}{\Gamma \vdash \eta b : \eta\texttt{bool}}$$

$$\frac{\Gamma_1 \vdash M : \eta\texttt{bool} \qquad \Gamma_2 \vdash N_1 : A \qquad \Gamma_2 \vdash N_2 : A}{\Gamma_1 \circ \Gamma_2 \vdash \texttt{if} \ M \ \texttt{then} \ N_1 \ \texttt{else} \ N_2 : A}$$

$$\frac{\Gamma, x : A \vdash M : B \qquad \eta(\Gamma)}{\Gamma \vdash \eta\lambda x.\, M : \eta(A \to B)}$$

$$\frac{\Gamma_1 \vdash M : \eta(A \to B) \qquad \Gamma_2 \vdash N : A}{\Gamma_1 \circ \Gamma_2 \vdash M\, N : B}$$

On note ici $\mathsf{un}(\Gamma)$ (et éventuellement $\eta(\Gamma)$) le prédicat disant que Γ le contient des types que de la forme $\mathsf{un}T$.

Dans le cas de l'axiome, on assure que l'on n'a pas de variables inutilisées. Dans le cas de l'abstraction, on contrôle l'usage des variables qui se retrouvent dans la clôture associée à $\lambda x.\, M$.

Exercice 1. Peut-on trouver un terme ayant pour type

1. $\mathsf{un}(\mathtt{unbool} \to \mathsf{lin}\mathtt{bool})$?
2. $\mathsf{un}(\mathsf{lin}\mathtt{bool} \to \mathtt{unbool})$?

On ne peut pas proposer :

1. $\mathsf{un}(\lambda x.\, x)$ car on n'a pas de lin
2. $\mathsf{un}(\lambda x.\, \mathtt{unfalse})$ car on n'utilise pas le x.

Oui, on peut :

1. $\mathsf{un}(\lambda x.\, \mathsf{lin}\mathtt{false})$
2. $\mathsf{un}(\lambda x.\, \mathtt{if}\ x\ \mathtt{then\ unfalse\ else\ untrue})$

Remarque 1 (Obligation de linéarité)**.** On a des règles de « bonne formation » : $\eta(A)$ et $\eta(\Gamma)$. La décomposition $\Gamma_1 \circ \Gamma_2$ correspond à un aiguillage.

Les types linéaires correspondent à des *types substructurels*. Ceci est relié à la *logique linéaire*.

Lemme 1. ▷ *Affaiblissement.* Si $\Gamma \vdash M : A$ alors pour $x \notin \mathrm{dom}(\Gamma)$ on a $\Gamma, x : \mathsf{un}T \vdash M : A$.

▷ *Contraction.* Si $\Gamma, x : \mathsf{un}T, y : \mathsf{un}T \vdash M : B$ alors on a $\Gamma, x : \mathsf{un}T \vdash M[x/y] : B$. Ceci n'est pas vrai avec lin.