TD TYPES

1 Expressions typables

Exercice 1 – Types simples :

1. `fun f = fun x -> x+1 in f (f 1)`

$$\frac{\dfrac{\dfrac{}{f : int \to int \vdash \texttt{f} : int \to int} \quad \dfrac{\dfrac{}{f : int \to int \vdash \texttt{f} : int \to int} \quad \dfrac{}{f : int \to int \vdash \texttt{1} : int}}{f : int \to int \vdash \texttt{f 1} : int}}{\vdash \texttt{fun x -> x + 1} : int \to int} \qquad \dfrac{\dfrac{}{f : int \to int \vdash \texttt{f} : int \to int} \quad \dfrac{\dfrac{}{f : int \to int \vdash \texttt{f} : int \to int} \quad \dfrac{}{f : int \to int \vdash \texttt{1} : int}}{f : int \to int \vdash \texttt{f 1} : int}}{f : int \to int \vdash \texttt{f (f 1)} : int}}{\vdash \texttt{let f = fun x -> x+1 in f (f 1)} : int}$$

2. `let f = fun x -> x + 1 in f f`
   Impossible car f ne peut pas être appliqué à lui même.
3. `let f = fun x -> fun y -> x in f 1`

$$\frac{\dfrac{\dfrac{\dfrac{}{x : int, y : int \vdash x : int}}{x : int \vdash \texttt{fun y -> x} : int \to int} \quad \dfrac{\dfrac{}{f : int \to int \to int \vdash f : int \to int \to int} \quad \dfrac{}{f : int \to int \to int \vdash 1 : int}}{f : int \to int \to int \vdash \texttt{f 1} : int \to int}}{\vdash \texttt{fun x -> fun y -> x} : int \to int \to int}}{\vdash \texttt{let f = fun x -> fun y -> x in f 1} : int \to int}$$

4. `let f = fun x -> fun y -> x in f 1 2 3`
   Impossible car f est de type $int \to int \to int$ et il est appliqué à 3 int au lieu de 2.
5. `let f = fun x -> fun y -> x in f f 2 3`
   Impossible f devrai être de type $\tau = \tau \to int \to \tau$
6. `let f = fun x -> fun y -> x in f (fun z -> z) 2 3`
   Tytpable avec f de type $(int \to int) \to int \to (int \to int)$
7. `fun x -> fun y -> fun z -> x z (y z)`
   Tytpalble $(\tau_1 \to \tau_2 \to \tau_3) \to (\tau_1 \to \tau_2) \to \tau_1 \to \tau_3$

Exercice 2 – Types polymorphes

1. `fun x -> x x`
   Typable avec $(\forall\alpha \to \alpha) \to (\forall\alpha \to \alpha)$
   H-M :
   Impossible car x devrai avoir un type infini $\alpha_1 \to (\alpha_1 \to (\ldots))$
   car on applique x à lui même donc doit se renvoyer lui même
2. `let f = fun x -> fun y -> x + 1 in f f 1`
   f : $int \to \alpha \to int$ non applicable à elle même
3. `let f = fun x -> fun y -> x + 1 in f 1 f`
   f : $int \to \alpha \to int$ avec α instancié avec $int \to \beta \to int$
4. `let f = fun x -> fun y -> x in f f 2 3`
   f : $\forall\alpha\beta, \alpha \to \beta \to \alpha$

2 Extensions

Exercice 3 – Paires

$$\frac{\Gamma \vdash e1 : \tau_1 \quad \Gamma \vdash e2 : \tau_2}{\Gamma \vdash \texttt{(e1, e2)} : \tau_1 \times \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{fst(e)} : \tau_1}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{snd(e)} : \tau_2}$$

$$\frac{\Gamma \vdash e1 : \tau_1 \times \tau_2 \quad \Gamma, x : \tau_1, y : \tau_2 \vdash e2 : \tau}{\Gamma \vdash \texttt{let x, y = e1 in e2} : \tau}$$

Exercice 4 – Booléens

$$\frac{}{\Gamma \vdash \texttt{true} : bool} \; \frac{}{\Gamma \vdash \texttt{false} : bool}$$

$$\frac{\Gamma \vdash \texttt{e1} : int \quad \Gamma \vdash \texttt{e2} : int}{\Gamma \vdash \texttt{e1 < e2} : bool}$$

$$\frac{\Gamma \vdash \texttt{e1} : \tau \quad \Gamma \vdash \texttt{e2} : \tau}{\Gamma \vdash \texttt{e1 = e2} : bool}$$

$$\frac{\Gamma \vdash \texttt{e1} : bool \quad \Gamma \vdash \texttt{e2} : bool}{\Gamma \vdash \texttt{e1 \&\& e2} : bool}$$

$$\frac{\Gamma \vdash \texttt{e0} : bool \quad \Gamma \vdash \texttt{e1} : \Gamma \vdash \tau \quad \texttt{e2} : \tau}{\Gamma \vdash \texttt{if e0 then e1 else e2} : \tau}$$

Exercice 5 – Point fixe

$$\frac{\texttt{x} : \tau \vdash \texttt{e1} : \tau \quad \texttt{e2} : \tau_1}{\vdash \texttt{let rec x = e1 in e2} : \tau_1}$$

$$\frac{}{\vdash \texttt{let rec f = fun x -> 1 + f x in f 0} : int}$$

Exercice 6 – Mise en oeuvre

3 Raisonnements

Exercice 7 – Opérateur paresseux

On a $F(e_1\&\&e_2) =$ if $F(e_1)$ then $F(e_2)$ else $false$
et $F(e) = e$
On va montrer $\Gamma \vdash e : \tau \Rightarrow \Gamma \vdash F(e) : \tau$ Par récurence :
Seul cas intéressent $\Gamma \vdash e_1\&\&e_2 : bool$ avec les prémisses $\Gamma \vdash e_1 : bool$ et $\Gamma \vdash e_2 : bool$.
Or la règle de typage pour le if nous donne bien $F(e_1\&\&e_2) : bool$.

Exercice 8 – Affaiblissement

On va procéder par induvtion structurelle sur les règles de dérivations :
On a $\Gamma \subseteq \Delta$

- $\Gamma \vdash x : \Gamma(x)$ par défintion de l'inclusion.
- $\Gamma \vdash \texttt{fun x -> e} : \tau_1 \to \tau_2$ On a bien $\Gamma, x : \tau_1 \subseteq \Delta, x : \tau_1$.
  Donc par hypothèse de récurence on à bien $\Delta \vdash \texttt{fun x -> e} : \tau_1 \to \tau_2$
- …

Exercice 9 – Adéquation

Démonstration par récurence assez simples.

Exercice 10 – Préservation des types