

TD1 : Systèmes de types

Description du langage

Expressions $e ::= x \mid \text{fun } x \rightarrow e \mid (e_1 \ e_2) \mid i \mid (e_1 + e_2) \mid (e_1 < e_2) \mid \text{not } e$
 Instructions $c ::= \text{while } e \text{ do } c \mid \text{print } e \mid \text{if } e \text{ then } c_1 \text{ else } c_2 \mid \{c_1; c_2\}$

Typage Types $T ::= \text{int} \mid \text{bool} \mid T \rightarrow T \mid \text{instr}$

$$\begin{array}{c} \frac{}{\Gamma, x:T \vdash x:T} \text{ (Var)} \quad \frac{\Gamma, x:T_1 \vdash e:T_2}{\Gamma \vdash \text{fun } x \rightarrow e : T_1 \rightarrow T_2} \text{ (Fun)} \quad \frac{\Gamma \vdash e_1:T_1 \rightarrow T_2 \quad \Gamma \vdash e_2:T_1}{\Gamma \vdash (e_1 \ e_2): T_2} \text{ (Call)} \\ \\ \frac{}{\Gamma \vdash i : \text{int}} \text{ (Int)} \quad \frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash (e_1 + e_2) : \text{int}} \text{ (Add)} \\ \\ \frac{\Gamma \vdash e_1 : T \quad \Gamma \vdash e_2 : T}{\Gamma \vdash (e_1 < e_2) : \text{bool}} \text{ (Lt)} \quad \frac{\Gamma \vdash e : \text{bool}}{\Gamma \vdash \text{not } e : \text{bool}} \text{ (Not)} \end{array}$$

Instructions

$$\begin{array}{c} \frac{\Gamma \vdash e : \text{bool} \quad \Gamma \vdash c : \text{instr}}{\Gamma \vdash \text{while } e \text{ do } c : \text{instr}} \text{ (While)} \quad \frac{\Gamma \vdash e : \text{int}}{\Gamma \vdash \text{print } e : \text{instr}} \text{ (Print)} \\ \\ \frac{\Gamma \vdash e : \text{bool} \quad \Gamma \vdash c_1 : T \quad \Gamma \vdash c_2 : T}{\Gamma \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : T} \text{ (If)} \\ \\ \frac{\Gamma \vdash c_1 : \text{instr} \quad \Gamma \vdash c_2 : \text{instr}}{\Gamma \vdash (c_1 ; c_2) : \text{instr}} \text{ (Seq)} \quad \frac{\Gamma, x : T \vdash e : T}{\Gamma \vdash x := e : \text{instr}} \text{ (Aff)} \end{array}$$

Exercice 1 Typage d'expressions simples

Déterminez le type des expressions suivantes en utilisant les règles de typage fournies. Si le type dépend de l'environnement, précisez-le.

- $(1+(2+3))$
- $(x+y)$
- $\text{fun } x \rightarrow (x + 1)$
- $\text{fun } x \rightarrow \text{if } (x < 5) \text{ then } (x + 1) \text{ else } (x + 2)$
- $\text{fun } x \rightarrow \text{not } (x < 5)$
- $(\text{fun } x \rightarrow x+x) (y+2)$
- $(f \ x) + 1$
- $\text{fun } x \rightarrow (((\text{fun } x \rightarrow x+1) \ 2) < \text{not } x)$

Exercice 2 Typage d'instructions composées

Considérez les instructions suivantes, et vérifiez que qu'elles sont bien typées en utilisant les règles de typage.

- $\text{while } (x < 10) \text{ do print } x; x := (x + 1)$
- $\text{if } (x < 5) \text{ then print } x \text{ else print } (x + 1)$

Exercice 3 Typage avec des fonctions d'ordre supérieur

Déterminez le type de la fonction suivante en appliquant les règles.

```
fun f -> fun x -> f (x + 1)
```

Exercice 4 Extension du langage

Étendez le langage en ajoutant les constructions suivantes :

- a) Boucle for avec la syntaxe : `for x from e1 to e2 do c`
- b) Listes et opérations de base avec les syntaxes :

```
nil
cons e1 e2
head e
tail e
```

Questions :

1. Modifiez la grammaire pour inclure la syntaxe de chaque nouvelle construction.
2. Proposez des règles de typage pour ces constructions.
3. Donnez un exemple d'utilisation de chaque construction et vérifiez leur typage.

Exercice 5 Fonctions Comment adapteriez-vous le système de types pour qu'il permette la vérification du typage des fonctions écrites avec la syntaxe suivante : `T Id (T Id, ..., T Id) {c}` ? On ajoutera pour cela au moins l'instruction `return e`.