# Fondements de la programmation

## Partiel 2 (2h)

*Document autorisé : la nouvelle feuille de rappels fournie.*

**Question A. Inférences de types.** Parmi les expressions suivantes, lesquelles sont typables et avec quel type et lesquelles ne sont pas typables et pourquoi ? 10 pt 60 min

1. `fun x => (1 + x) * (1 + x)`
2. `let x = (10, "") in (proj`$_1$`(x) + 1, proj`$_2$`(x))`
3. `let y = ref 1 in let x = y in (x := 2; !x)`
4. $\Lambda$ A. $\Lambda$ B. `fun (f:A -> B) => fun (g:B -> C) => (x: A) => g(f(x))`
5. Trouver un contexte $\Gamma$ pour lequel l'expression `fix x` est typable ou bien expliquer pouquoi il n'en existe pas.

### 0.1 Réponses **CORRECTION**

`let x = 21 in Some(x) as {Some: num | None: Unit}` est typable. Posons

$$T = \mathrm{sum}(\mathrm{Some}\,.\,\mathrm{num}; \mathrm{None}\,.\,\mathrm{prod}())$$

$$\dfrac{\dfrac{}{\vdash \mathrm{num}[21] : \mathrm{num}}\,\text{num} \qquad \dfrac{\dfrac{}{x : \mathrm{num} \vdash x : \mathrm{num}}\,\text{id}}{x : \mathrm{num} \vdash \mathrm{Some}\{T\}(x) : T}\,\text{sum}}{\Gamma \vdash \mathrm{let}(\mathrm{tuple}(\mathrm{num}[22], \mathrm{num}[23]), x.\,\mathrm{Some}\{T\}(x)) : T}\,\text{let}$$

`ref 1 := 3` est typable

$$\text{assign}\,\dfrac{\text{ref}\,\dfrac{\dfrac{}{\vdash \mathrm{num}[1] : \mathrm{num}}\,\text{num}}{\vdash \mathrm{ref}\,\mathrm{num}[1] : \mathrm{Ref}\,\mathrm{num}} \qquad \dfrac{}{\vdash \mathrm{num}[3] : \mathrm{num}}\,\text{num}}{\vdash \mathrm{ref}\,\mathrm{num}[1] := \mathrm{num}[3] : \texttt{Unit}}$$

`x := ref 1; !!x + 1` est typable. Posons $\Gamma = x : \mathrm{Ref}\,\mathrm{Ref}\,\mathrm{num}$ et $\Sigma = \emptyset$.

$$\text{seq}\,\dfrac{\dfrac{\dfrac{\text{id}}{\Gamma \vdash x : \mathrm{Ref}\,\mathrm{Ref}\,\mathrm{num}} \qquad \dfrac{\dfrac{\text{num}}{\Gamma \vdash 1 : \mathrm{num}}}{\Gamma \vdash \mathrm{ref}\,1 : \mathrm{num}}\,\text{ref}}{\Gamma \vdash x := \mathrm{ref}\,1 : \texttt{Unit}}\,\text{assign} \qquad \text{plus}\,\dfrac{\text{Deref}\,\dfrac{\text{Deref}\,\dfrac{\dfrac{\text{id}}{\Gamma \vdash x : \mathrm{Ref}\,\mathrm{Ref}\,\mathrm{num}}}{\Gamma \vdash !x : \mathrm{Ref}\,\mathrm{num}}}{\Gamma \vdash !!x : \mathrm{num}} \qquad \dfrac{\text{num}}{\Gamma \vdash 1 : \mathrm{num}}}{\Gamma \vdash !!x + 1 : \mathrm{num}}}{\Gamma \vdash x := \mathrm{ref}\,1; !!x + 1 : \mathrm{num}}$$

`let x = 10 in let y = 5 in (x + y) * x` est typable. Pour gagner de la place, on note $Xx : \mathrm{num}$, $Yy : \mathrm{num}$ et $\Gamma = X, Y$.

$$\text{let}\,\dfrac{\dfrac{\text{num}}{\vdash 10 : \mathrm{num}} \qquad \text{let}\,\dfrac{\dfrac{\text{num}}{X \vdash 5 : \mathrm{num}} \qquad \text{mult}\,\dfrac{\text{plus}\,\dfrac{\dfrac{\text{id}}{\Gamma \vdash x : \mathrm{num}} \quad \dfrac{\text{id}}{\Gamma \vdash y : \mathrm{num}}}{\Gamma \vdash x + y : \mathrm{num}} \qquad \dfrac{\text{id}}{X, Y \vdash x : \mathrm{num}}}{X, Y \vdash (x + y) * x : \mathrm{num}}}{X \vdash \texttt{let y = 5 in (x + y) * x} : \mathrm{num}}}{\vdash \texttt{let x = 10 in let y = 5 in (x + y) * x} : \mathrm{num}}$$

$\Lambda$ A.$\Lambda$ B. `fun (x: A) => fun (f: A -> B) => f(x)` est typable

$$\text{T-abs}\,\dfrac{\text{T-abs}\,\dfrac{\text{abs}\,\dfrac{\text{abs}\,\dfrac{\text{app}\,\dfrac{\dfrac{\text{id}}{A,B,x:A,f:A\to B\vdash x:A}}{A,B,x:A,f:A\to B\vdash f(x):B}}{A,B,x:A\vdash \texttt{fun (f: A->B) => f(x)}:(A\to B)\to B}}{A,B\vdash \texttt{fun(x: A) => fun (f: A->B) => f(x)}:A\to(A\to B)\to B}}{A\vdash \Lambda B\texttt{fun(x: A) => fun (f: A->B) => f(x)}:\forall B.A\to(A\to B)\to B}}{\vdash \Lambda A\Lambda B\texttt{fun(x: A) => fun (f: A->B) => f(x)}:\forall A\forall B.A\to(A\to B)\to B}$$

**Question B. Dynamique structurelle.** réduire à une valeur ou une erreur les expressions suivantes à l'aide de la dynamique structurelle en justifiant chaque étape de réduction :

1. `let y = ref 1 in let x = y in (x := 2; !y)`
2. Soit `T` le type `{Some: num | None: Unit}` 4 pt 24 min

```
let x = None(unit) as T in match x with
 | Some(x1) => x1 + 1
 | None(x2) => 0
```

Pour vous aider et limiter la durée du partiel, la première étape de l'évaluation du premier terme, que l'on note ici $\mathrm{t}_1$, est :

$$\dfrac{\dfrac{1\ \text{val}\qquad c\notin \text{keys}(\emptyset)}{[\emptyset]\texttt{ref 1}\mapsto[(c\to 1)]\texttt{c}}\,\text{ref}}{[\emptyset]t_1\mapsto[(c\to 1)]\texttt{let y = c in let x = y in (x := 2; !y))}}\,\text{letG}$$

Comme un lieu mémoire tel que `c` est toujours une valeur, la règle suivante sera un *let*. À vous de jouer !

**Question C. Machine abstraite de Krivine.** Évaluer le terme $(\lambda x.y)((\lambda z.z)(w))$ sur la KAM par nom, puis sur la KAM par valeur. 4 pt 24 min

Pour vous aider et limiter la durée du partiel, voici les trois premières étapes de l'exécution del a KAM par valeur sur ce terme.

$$\dfrac{\dfrac{((\lambda x.y)((\lambda z.z)w))\quad \emptyset\quad \varepsilon}{\lambda x.y\quad \emptyset\quad A(((\lambda z.z)w),\emptyset)}\,\text{push}}{((\lambda z.z)w)\quad \emptyset\quad F(\lambda x.y,\emptyset)}\,\text{swap}$$

## 0.2 Réponses

CORRECTION

Évaluation de $((\lambda x.y)((\lambda z.z)w))$ par nom :

$$\dfrac{\dfrac{((\lambda x.y)((\lambda z.z)w))\quad \emptyset\quad \varepsilon}{\lambda x.y\quad \emptyset\quad (((\lambda z.z)w),\emptyset)}\,\text{push}}{y\quad x\to(((\lambda z.z)w),\emptyset)\quad \varepsilon}\,\text{pop}$$

Évaluation de $((\lambda x.y)((\lambda z.z)w))$ par valeur :

$$
\begin{array}{c|l}
((\lambda x.y)((\lambda z.z)w)) \quad \emptyset \quad \varepsilon & \\
\cline{1-1}
\lambda x.y \quad \emptyset \quad A(((\lambda z.z)w),\emptyset) & \text{push} \\
\cline{1-1}
((\lambda z.z)w) \quad \emptyset \quad F(\lambda x.y,\emptyset) & \text{swap} \\
\cline{1-1}
\lambda z.z \quad \emptyset \quad A(w,\emptyset)F(\lambda x.y,\emptyset) & \text{push} \\
\cline{1-1}
w \quad \emptyset \quad F(\lambda z.z,\emptyset)F(\lambda x.y,\emptyset) & \text{swap} \\
\cline{1-1}
z \quad z \to (w,\emptyset) \quad F(\lambda x.y,\emptyset) & \text{pop} \\
\cline{1-1}
w \quad \emptyset \quad F(\lambda x.y,\emptyset) & \text{deref} \\
\cline{1-1}
y \quad x \to (w,\emptyset) \quad \varepsilon & \text{pop}
\end{array}
$$

**Question D. Langage.** Que signifie pour un langage de programmation que les fonctions y sont des *citoyennes de première classe* ? Donner un exemple de langage dans lesquels les fonctions ne sont pas des citoyennes de première classe et un exemple de langage dans lequel les fonctions sont des citoyennes de première classe. Oui, c'est rigoureusement la même question qu'au premier partiel. 2 pt 12 min

## 0.3 Réponse **CORRECTION**

**Question E. Démonstration (bonus).** Si $e$ est une valeur et que $\vdash e : \mathrm{num}$ alors $e = \mathrm{num}[n]$ pour un certain nombre $n$. Prouvez-le. 2 pt 12 min

## 0.4 Réponse **CORRECTION**