

Types récurrents.

Documents autorisés (en particulier le poly).

Les questions sont annotées par une estimation du temps de leur résolution en minutes. Les questions les plus rapides sont aussi (généralement) celles qui rapportent le moins de points.

On se propose d'examiner une variante du système des types simples autorisant d'écrire des *types récurrents*. C'est déjà ce que propose CaML, où un type défini peut être défini en fonction de lui-même, par exemple :

```
type arbre = FEUILLE of int
           | NOEUD of arbre * arbre;;
type paradox = P of paradox -> paradox;;
type nat = 0
          | S of nat;;
```

On va, pour représenter cette récursivité, ajouter au langage des types simples une construction $\mu X \cdot f(X)$, qui dénote en gros l'unique type τ que l'on définirait en CaML en écrivant `type $\tau = f(\tau)$;;`. Par exemple, on coderait les types ci-dessus sous la forme de $\text{arbre} = \mu X \cdot \text{int} + (X \times X)$, $\text{paradox} = \mu X \cdot X \Rightarrow X$, $\text{nat} = \mu X \cdot \text{unit} + X$.

Formellement, les types sont :

$$F ::= b|X|F \Rightarrow G|\mu X \cdot F$$

où les b sont des types de base, les X des variables de type et où la variable de type X est liée dans $\mu X \cdot F$. On supposera que l' α -renommage est appliqué aux types implicitement.

On note \sim la plus petite congruence sur les types telle que $\mu X \cdot F \sim F[X := \mu X \cdot F]$. On rappelle qu'une congruence est une relation d'équivalence qui passe au contexte. On peut caractériser \sim comme la relation définie par les règles :

$$\begin{array}{c} \frac{}{\mu X \cdot F \sim F[X := \mu X \cdot F]} (Fold) \quad \frac{}{F \sim F} (Refl) \\ \\ \frac{F \sim G}{G \sim F} (Sym) \quad \frac{F \sim G \quad G \sim H}{F \sim H} (Trans) \\ \\ \frac{F \sim F' \quad G \sim G'}{F \Rightarrow G \sim F' \Rightarrow G'} (\Rightarrow) \quad \frac{F \sim G}{\mu X \cdot F \sim \mu X \cdot G} (\mu) \end{array}$$

Autrement dit, $F \sim G$ si et seulement si on peut le déduire par une dérivation (finie) utilisant les règles ci-dessus.

Les règles de typage sont celles des types simples plus une règle portant sur les types récur-sifs. Appelons le système suivant *système* μ :

$$\frac{}{\Gamma, x : F \vdash x : F} (Var)$$

$$\frac{\Gamma \vdash u : F_1 \Rightarrow F_2 \quad \Gamma \vdash v : F_1}{\Gamma \vdash uv : F_2} (App) \quad \frac{\Gamma, y : F_1 \vdash u[x := y] : F_2}{\Gamma \vdash \lambda x \cdot u : F_1 \Rightarrow F_2} (Abs)$$

$$\frac{\Gamma \vdash u : F \quad F \sim G}{\Gamma \vdash u : G} (TEq)$$

On ne considérera que la règle β , pas η , comme règle de réduction du λ -calcul dans cette partie.

1. (5 min.) Soit P le type $\mu X \cdot X \Rightarrow X$ (c'est le type paradox plus haut). Un P -contexte est un contexte Γ où toutes les variables sont de type P , c'est-à-dire un contexte de la forme $x_1 : P, \dots, x_n : P$. Montrer que, dans n'importe quel P -contexte Γ , pour tout λ -terme (non typé) u dont les variables libres apparaissent toutes dans Γ , on peut dériver le jugement $\Gamma \vdash u : P$. dans le système μ .
2. (1 min.) En déduire qu'il existe des λ -termes non fortement normalisants, et même non faiblement normalisants, dans le système μ . On exhibera un contre-exemple.
3. (15 min.) On va réparer ce défaut en considérant un sous-système μ^+ du système μ , où dans $\mu X \cdot F$ la variable X ne peut avoir que des occurrences positives dans F . Informel-lement, si l'on voit les types comme des formules, les occurrences positives sont celles qui apparaissent sous un nombre pair de négations, chaque implication $F \Rightarrow G$ comptant comme une négation de F .

Formellement, on dit que X apparaît positivement dans une variable de type Y si et seule-ment si $X = Y$; dans $F \Rightarrow G$ si X apparaît positivement dans G ou négativement dans F ; dans $\mu Y \cdot F$ (avec $Y \neq X$) si X apparaît positivement dans F . Et que, symétriquement, X apparaît négativement dans un type si et seulement si : ce type est $F \Rightarrow G$ et X apparaît négativement dans G ou positivement dans F ; ou ce type est $\mu Y \cdot F$ (avec $Y \neq X$) et X apparaît négativement dans F .

Le système μ^+ est la restriction du système μ où, dans tout type $\mu X \cdot F$, X n'apparaît pas négativement dans F .

Lister les variables apparaissant positivement, resp. négativement, dans les types suivants. On rappelle que $A \Rightarrow B \Rightarrow C$ signifie $A \Rightarrow (B \Rightarrow C)$, et que la portée de μ , comme celle de λ , s'étend aussi loin à droite que possible.

$$\begin{aligned} (i) X \Rightarrow (X \Rightarrow Y) \quad (ii) (X \Rightarrow X) \Rightarrow Y \quad (iii) (X \Rightarrow Y) \Rightarrow X \\ (iv) X \Rightarrow Z \Rightarrow Y \quad (v) (\mu Y \cdot X \Rightarrow Z \Rightarrow Y) \Rightarrow Y \Rightarrow X \\ (vi) \mu X \cdot (\mu Y \cdot X \Rightarrow Z \Rightarrow Y) \Rightarrow Y \Rightarrow X \end{aligned}$$

Lesquels sont des types de μ^+ ? Répondre sous forme d'un tableau :

Formule	Apparaissant positivement	Apparaissant négativement	Dans μ^+ ?
(i)			
(ii)			
(iii)			
(iv)			
(v)			
(vi)			

4. (5 min.) On rappelle qu'un *candidat de réductibilité* est un ensemble de λ -termes S tel que :

(CR1) Si $u \in S$, alors $u \in SN$.

(CR2) Si $u \in S$ et $u \rightarrow u'$, alors $u' \in S$.

(CR3) Si u est neutre et pour tout u' tel que $u \rightarrow u'$, $u' \in S$, alors $u \in S$.

On notera CR l'ensemble de tous les candidats de réductibilité, ordonné par l'ordre d'inclusion \subseteq . Montrer qu'il existe un plus grand candidat de réductibilité S_\top , c'est-à-dire un candidat de réductibilité S_\top tel que $S \subseteq S_\top$ pour tout $S \in CR$.

5. (10 min.) Montrer que CR est un *treillis complet*, c'est-à-dire que toute famille $(S_i)_{i \in I}$ de candidats de réductibilité, il existe un plus grand candidat de réductibilité S tel que $S \subseteq S_i$ pour tout $i \in I$ — la *borne inférieure* de la famille $(S_i)_{i \in I}$. (Indication : $S = \bigcap_{i \in I} S_i$ convient-il ? Que se passe-t-il lorsque I est vide ?)

6. (10 min.) Vous avez déjà vu dans d'autres cours le théorème du point fixe de Tarski. Un treillis complet est un ensemble ordonné \mathcal{L}, \sqsubseteq tel que toute famille $(z_i)_{i \in I}$ d'éléments de \mathcal{L} a une borne inférieure $\bigwedge_{i \in I} z_i$ dans \mathcal{L} . Une fonction $f : \mathcal{L} \rightarrow \mathcal{L}$ est dite *monotone* si et seulement si $f(z) \sqsubseteq f(z')$ pour tous $z \sqsubseteq z'$. On a :

Théorème (Tarski) Soit \mathcal{L}, \sqsubseteq un treillis complet, $f : \mathcal{L} \rightarrow \mathcal{L}$ une fonction. Posons $lfp(f) = \bigwedge_{z \in \mathcal{L}, f(z) \sqsubseteq z} z$. Si f est monotone, alors $lfp(f)$ est le *plus petit point fixe* de f . (On rappelle qu'un point fixe z de f est un élément tel que $f(z) = z$.)

On définit alors une notion RED_F adaptée à μ . Pour tout *contexte de candidats* C qui à chaque variable de type X associe un candidat de réductibilité, pour tout type F de μ , on définit RED_F^C par :

$$\begin{aligned}
 RED_b^C &\hat{=} SN && \text{(ensemble des termes fortement normalisants)} \\
 RED_X^C &\hat{=} C(X) \\
 RED_{F \Rightarrow G}^C &\hat{=} \{u \mid \forall v \in RED_F^C \cdot uv \in RED_G^C\} \\
 RED_{\mu X.F}^C &\hat{=} lfp(S \in CR \mapsto RED_F^{C[X:=S]})
 \end{aligned}$$

où la notation $S \in CR \mapsto RED_F^{C[X:=S]}$ dénote la fonction qui à tout candidat S associe $RED_F^{C[X:=S]}$.

Montrer que RED_F^C est bien défini et est un candidat de réductibilité pour tout contexte de candidats C et tout type F de μ . (Certains des cas ont déjà été vus en cours. On pourra alors

s'économiser une démonstration à condition de citer précisément l'argument — numéro de théorème, notamment x1.)

7. (15 min.) Montrer que, lorsque F est un type de μ^+ :

- (a) si X n'apparaît pas négativement dans F alors RED_F^C est monotone en $C(X)$, au sens où pour tout S tel que $C(X) \subseteq S$, $RED_F^C \subseteq RED_F^{C[X:=S]}$;
- (b) si X n'apparaît pas positivement dans F alors RED_F^C est antitone en $C(X)$, au sens où pour tout S tel que $C(X) \subseteq S$, $RED_F^C \supseteq RED_F^{C[X:=S]}$.

On admettra le résultat (trivial) selon lequel : (*) si f et g sont deux fonctions monotones d'un treillis complet \mathcal{L} vers \mathcal{L} telles que $f(z) \sqsubseteq g(z)$ pour tout $z \in \mathcal{L}$, alors $lfp(f) \sqsubseteq lfp(g)$.

- 8. (20 min.) En déduire que tout terme typé dans le système μ^+ est fortement normalisant. (Comme à la question 6, on pourra se référer à certaines preuves du cours pour les parties de la preuve qui ne changent pas ici. Citer précisément l'argument.)
- 9. (10 min.) On étend l'algèbre de types par une quantification au second ordre, comme dans le système F_2 :

$$F ::= b|X|F \Rightarrow G|\mu X \cdot F|\forall X \cdot F$$

La relation \sim est étendue par la règle :

$$\frac{F \sim G}{\forall X \cdot F \sim \forall X \cdot G} (\forall)$$

On ajoute les deux règles de typage :

$$\frac{\Gamma \vdash u : \forall X \cdot F}{\Gamma \vdash uG : F[X := G]} (TApp) \qquad \frac{\Gamma \vdash u : F}{\Gamma \vdash \lambda X \cdot u : \forall X \cdot F} (TAbs)$$

(X non libre dans aucune formule de Γ)

Le système $\forall\mu$ est le système de typage ainsi obtenu. Le système $\forall\mu^+$ est le sous-système où tous les sous-types de la forme $\mu X \cdot F$ sont tels que X n'a aucune occurrence négative dans F . (X ayant une occurrence positive, resp. négative, dans $\forall Y \cdot G$ si et seulement si $X \neq Y$ et X a une occurrence positive, resp. négative, dans G .)

On considère désormais les règles de réduction :

$$\begin{aligned} (\beta) \quad & (\lambda x \cdot u)v \rightarrow u[x := v] \\ (Beta) \quad & (\lambda X \cdot u)F \rightarrow u[X := F] \end{aligned}$$

Montrer que tout terme typé dans le système $\forall\mu^+$ est fortement normalisant pour la réduction définie par (β) et $(Beta)$.