Types récursifs.

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 récursifs*. 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 = O
         | 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` τ `=` $f(\tau)$`;;`. Par exemple, on coderait les types ci-dessus sous la forme de `arbre` $= \mu X \cdot$ `int` $+ (X \times X)$, `paradox` $= \mu X \cdot X \Rightarrow X$, `nat` $= \mu X \cdot$ `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 :

$$\frac{}{\mu X \cdot F \sim F[X := \mu X \cdot F]}\,(Fold) \quad \frac{}{F \sim F}\,(Refl)$$

$$\frac{F \sim G}{G \sim F}\,(Sym) \quad \frac{F \sim G \quad G \sim H}{F \sim H}\,(Trans)$$

$$\frac{F \sim F' \quad G \sim G'}{F \Rightarrow G \sim F' \Rightarrow G'}\,(\Rightarrow) \quad \frac{F \sim G}{\mu X \cdot F \sim \mu X \cdot G}\,(\mu)$$

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 récursifs. Appelons le système suivant *système μ* :

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

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

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

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 $x_1 : P, \ldots, x_n : P$. 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.

   Par récurrence structurelle sur le λ-terme u. Si u est une variable, c'est par hypothèse sur Γ. Les applications sont typées comme suit, en notant que $P \sim P \Rightarrow P$:

$$\frac{\dfrac{\vdots}{\Gamma \vdash u : P}\,(TEq) \qquad \dfrac{\vdots}{\Gamma \vdash v : P}}{\Gamma \vdash uv : P}\,(App)$$

   Les abstractions sont typées comme suit :

$$\frac{\dfrac{\vdots}{\Gamma, x : P \vdash u : P}\,(Abs)}{\Gamma \vdash \lambda x \cdot u : P}\,(TEq)$$

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.

   Le terme $\delta\delta$, avec $\delta \hat{=} \lambda x \cdot xx$, est un contre-exemple connu en λ-calcul non typé, donc aussi en système μ.

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. Informellement, 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 seulement 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.

$$(i)\ X \Rightarrow (X \Rightarrow Y) \qquad (ii)\ (X \Rightarrow X) \Rightarrow Y \qquad (iii)\ (X \Rightarrow Y) \Rightarrow X$$
$$(iv)\ X \Rightarrow Z \Rightarrow Y \qquad (v)\ (\mu Y \cdot X \Rightarrow Z \Rightarrow Y) \Rightarrow Y \Rightarrow X$$
$$(vi)\ \mu X \cdot (\mu Y \cdot X \Rightarrow Z \Rightarrow Y) \Rightarrow Y \Rightarrow X$$

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)			

Formule	*Apparaissant positivement*	*Apparaissant négativement*	*Dans μ^+ ?*
(i)	Y	X	*oui*
(ii)	X, Y	X	*oui*
(iii)	X	Y	*oui*
(iv)	Y	X, Z	*oui*
(v)	X, Z	Y	*oui*
(vi)	Z	Y	*oui*

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

On notera $\mathcal{C}R$ 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 $S \in \mathcal{C}R$.

$S_\top = SN$ est clairement un candidat de réductibilité, et $S \subseteq SN$ pour tout candidat S, par la propriété (CR1).

5. (10 min.) Montrer que $\mathcal{C}R$ 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 ?)

   On commence par regarder ce qui se passe lorsque I est vide. Alors l'intersection $\bigcap_{i\in I} S_i$ n'est pas définie. La définir, logiquement, comme l'ensemble de tous les λ-termes qui sont dans tous les S_i, fournirait l'ensemble Λ de tous les λ-termes. Mais Λ n'est pas un candidat de réductibilité, puisque $\delta\ \delta$ est dans Λ mais pas dans SN.

   À la place, nous définissons (pour I quelconque) la borne inférieure S comme étant l'ensemble des λ-termes fortement normalisants *qui sont dans tous les S_i, $i \in I$. Une autre façon de le dire est que S est l'intersection des S_i, $i \in I$,* et *de SN. Si $u \in S$ alors $u \in SN$ par définition, donc S vérifie (CR1). Si $u \in S$ et $u \to v$, alors comme $u \in SN$ et $u \in S_i$ pour tout i, et que SN et S_i sont des candidats donc vérifient (CR2), nécessairement $v \in SN$ et $v \in S_i$ pour tout i, donc $v \in S$. Donc S vérifie (CR2). Si u est neutre et tous ses réduits en une étape sont dans S, ils sont dans SN et dans chaque S_i. Comme SN et les S_i sont des candidats, par (CR3), u est dans SN et dans chaque S_i, donc dans S. Donc S vérifie (CR3).*

   S est donc un candidat, et clairement le plus grand tel que S soit inclus à la fois dans les S_i et dans SN. Comme tout candidat est nécessairement inclus dans SN par (CR1), S est le plus grand inclus dans tous les S_i, c'est donc bien la borne inférieure des S_i.

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 $\bigsqcap_{i\in I} z_i$ dans $\mathcal{L}$. Une fonction $f : \mathcal{L} \to \mathcal{L}$ 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, $f : \mathcal{L} \to \mathcal{L}$ une fonction. Posons $lfp(f) = \bigsqcap_{z\in\mathcal{L}, f(z)\sqsubseteq z} z$. 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^C_F par :

$$\begin{array}{rcl} RED^C_b & \hat{=} & SN \\ & & \text{(ensemble des termes fortement normalisants)} \\ RED^C_X & \hat{=} & C(X) \\ RED^C_{F\Rightarrow G} & \hat{=} & \{u \mid \forall v \in RED^C_F \cdot uv \in RED^C_G\} \\ RED^C_{\mu X\cdot F} & \hat{=} & lfp(S \in \mathcal{C}R \mapsto RED^{C[X:=S]}_F) \end{array}$$

où la notation $S \in \mathcal{C}R \mapsto RED_F^{C[X:=S]}$ 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, notammentx1.)

Noter que ceci contient deux difficultés : montrer que la définition est une récurrence bien fondée comme dans le cours d'une part, et montrer que $S \in \mathcal{C}R \mapsto RED_F^{C[X:=S]}$ est bien une fonction de $\mathcal{C}R$ dans $\mathcal{C}R$ dans le cas de la dernière ligne. Pour cette dernière raison, nous devons démontrer que RED_F^C est bien défini et est un candidat simultanément. *Ceci est une récurrence sur la taille de la formule F.*

Si F est un type de base b, $RED_F^C = SN$ est un candidat, comme il l'a déjà été remarqué dans le cours.

Si F est une variable Y, c'est parce que C est un contexte de candidats.

Si F est de la forme $G \Rightarrow H$, alors REF_F^C est bien défini, et est un candidat par les arguments usuels déjà utilisés dans la preuve de normalisation forte du système F_2 du cours.

Si F est de la forme $\mu Y \cdot G$, alors par l'hypothèse de récurrence la fonction $S \in \mathcal{C}R \mapsto RED_G^{C[Y:=S]}$ est bien une fonction de $\mathcal{C}R$ dans $\mathcal{C}R$. Alors $lfp(S \in \mathcal{C}R \mapsto RED_G^{C[Y:=S]})$ est bien défini, et est dans le treillis complet $\mathcal{C}R$. Donc REF_F^C est bien défini et un candidat de réductibilité.

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 $lfp(f) \sqsubseteq lfp(g)$.

   On montre (a) et (b) simultanément par récurrence structurelle sur F.

   Si F est un type de base b, c'est évident, puisque $RED_F^C = SN = RED_F^{C[X:=S]}$.

   Si F est une variable Y, (a) et (b) sont triviaux dès que $X \neq Y$. Si $Y = X$, X apparaît positivement mais pas négativement dans Y : (a) est clair, et (b) est trivialement vrai (hypothèse "X n'apparaît pas positivement" fausse).

   Si F est de la forme $G \Rightarrow H$, montrons (a) : supposons que X n'apparaisse pas négativement dans $G \Rightarrow H$, autrement dit X n'apparaît pas positivement dans G et pas négativement dans H. Pour tout $u \in RED_F^C$, pour tout candidat S tel que $C(X) \subseteq S$, on souhaite montrer que $u \in RED_F^{C[X:=S]}$. Soit donc v quelconque dans $RED_G^{C[X:=S]}$: par hypothèse de récurrence (b), puisque X n'apparaît pas positivement dans G, v est

dans RED^C_G. Comme $u \in RED^C_F$, il s'ensuit que $uv \in RED^C_H$. Par hypothèse de récurrence (a), puisque X n'apparaît pas négativement dans H, $uv \in RED^{C[X:=S]}_H$. Comme v est quelconque, u est donc dans $RED^{C[X:=S]}_F$, et (a) est prouvé. La preuve de (b) est symétrique.

Si F est de la forme $\mu Y \cdot G$, montrons (a). Supposons que X n'apparaisse pas négativement dans $\mu Y \cdot G$, autrement dit X n'apparaît pas négativement dans G. Par α-renommage, on peut supposer $X \neq Y$. Soit d'autre part $S' \supseteq C(X)$. Alors la fonction $f \hat{=} (S \in \mathcal{CR} \mapsto RED^{C[Y:=S]}_G)$ est inférieure point à point à $g \hat{=} (S$ candidat $\mapsto RED^{C[X:=S',Y:=S]}_G)$, puisque X n'apparaît pas négativement dans G, en utilisant l'hypothèse de récurrence (a). De même, par hypothèse de récurrence (a), f et g sont monotones, cette fois-ci en utilisant que Y n'a pas d'occurrence négative dans G, puisque F est un type de μ^+. On peut donc appliquer $(*)$, et en déduire $lfp(f) \subseteq lfp(g)$, autrement dit $RED^C_F \subseteq RED^{C[X:=S']}_F$, ce qui prouve (a). Le cas (b) est symétrique.

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.)

   On montre par récurrence structurelle sur la dérivation de typage donnée de $x_1 : F_1, \ldots, x_n : F_n \vdash u : F$ en système μ^+ que pour tout contexte de candidats C, pour tous $v_1 \in RED^C_{F_1}, \ldots, v_n \in RED^C_{F_n}$, alors $u[x_1 := v_1, \ldots, x_n := v_n] \in RED^C_F$.

   Les cas où la dernière règle appliquée est (Var), (App) ou (Abs) sont comme dans la preuve pour le système F_2, et même comme dans la preuve pour le système des types simples. La seule difficulté est le cas où la dernière règle appliquée est (TEq). Il suffit pour régler ce cas de montrer que $F \sim G$ implique $RED^C_F = RED^C_G$.

   Ceci se démontre par récurrence sur la dérivation donnée de $F \sim G$. Les cas de $(Refl)$, (Sym), $(Trans)$, $(\Rightarrow)$ et (μ) sont évidents. Il ne reste donc qu'à démontrer que : $(\dagger)$ $RED^C_{\mu X \cdot F} = RED^C_{F[X:=\mu X \cdot F]}$.

   Je prétends qu'en général : $(**)$ $RED^C_{G[X:=H]} = RED^{C[X:=RED^C_H]}_G$, comme dans le cas du système F_2 du cours. Alors $(\dagger)$ en est une conséquence facile : $RED^C_{\mu X \cdot F} = lfp(S \in \mathcal{CR} \mapsto RED^{C[X:=S]}_F) = RED^{C[X:=lfp(S \in \mathcal{CR} \mapsto RED^{C[X:=S]}_F)]}_F$ (… car c'est un point fixe) $= RED^{C[X:=RED^C_{\mu X \cdot F}]}_F = RED^C_{F[X:=\mu X \cdot F]}$ (par $(**)$).

   Montrons d'abord, comme dans le point $(*)$ de la preuve de normalisation forte du système F_2 du cours, que : $(*)$ si Y n'est pas libre dans H, alors $RED^C_H = RED^{C[Y:=S]}_H$ pour tout candidat S. C'est évident lorsque H est un type de base, et est comme dans le cours pour les types variables ou implications. Lorsque H est de la forme $\mu Z \cdot K$, $RED^{C[Y:=S]}_H = lfp(S' \in \mathcal{CR} \mapsto RED^{C[Y:=S][Z:=S']}_K) = lfp(S' \in \mathcal{CR} \mapsto RED^{C[Z:=S'][Y:=S]}_K) = lfp(S' \in \mathcal{CR} \mapsto RED^{C[Z:=S']}_K)$ (car Y n'est pas libre non plus dans K, et l'on peut donc appliquer l'hypothèse de récurrence) $= RED^C_H$.

   Il ne reste donc finalement qu'à démontrer $(**)$, ce que l'on fait par récurrence structurelle sur G. Le cas où G est un type de base est évident (les deux côtés sont SN). Les

cas où G est une variable de type ou une implication sont comme dans le point $(**)$ de la preuve de normalisation forte du système F_2 du cours. Le cas où G est de la forme $\mu Y \cdot G'$ se traite comme suit : $RED^C_{G[X:=H]} = lfp(S \text{ candidat} \mapsto RED^{C[Y:=S]}_{G'[X:=H]}) = lfp(S \text{ candidat} \mapsto RED^{C[Y:=S][X:=RED^{C[Y:=S]}_H]}_{G'})$ *(par hypothèse de récurrence)* $= lfp(S \in \mathcal{C}R \mapsto RED^{C[Y:=S,X:=RED^C_H]}_{G'})$ *(par $(*)$)* $= RED^{C[X:=RED^C_H]}_G$.

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 :

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

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

Je ne le développe pas ici. C'est un mélange du théorème de normalisation forte pour le système F_2 et des arguments plus haut. Il suffit notamment d'étendre la définition de RED^C_F plus haut par la clause du cours :

$$RED^C_{\forall X \cdot F} \;\hat{=}\; \{u|\forall G, \forall S \in \mathcal{C}R \cdot uG \in RED^{C[X:=S]}_F\}$$