Examen de λ-calcul 2017

Correction.

Documents écrits autorisés (en particulier le poly).

1 Le λ-calcul par valeur

On appelle *valeur* tout terme qui est une variable ou une abstraction ; en somme, tout terme qui n'est pas une application.

On notera en général V les valeurs, pour les distinguer des termes généraux s, t, u, v, … En suivant cette convention, la règle de β*-réduction par valeur* est :

$$(\beta_V) \quad (\lambda x.u)V \to u[x := V].$$

Donc, par exemple, $(\lambda x.u)(vw)$ n'est pas un β_V-rédex. Le calcul obtenu en considérant la règle (β_V) est le λ*-calcul par valeur*. Celui qui se fonde sur la règle (β) usuelle sera parfois appelé λ*-calcul par nom* pour le distinguer de ce dernier.

1. [trivial] Donner un terme qui ne termine pas (même faiblement) en λ-calcul par valeur.

   $\Omega = \delta\delta$, où $\delta = \lambda x.xx\ldots$ qui est une valeur.

On définit une traduction par passage à la continuation comme suit. Dans chaque cas, k est un λ-terme arbitraire, appelé la continuation.

$$\begin{array}{rclrcl} (V)^*k & = & k(V^\dagger) & x^\dagger & = & x \\ (uv)^*k & = & u^*(\lambda f.(v)^*(fk)) & (\lambda x.u)^\dagger & = & \lambda k', x.(u)^*k' \end{array}$$

Les variables k', f sont bien sûr fraîches sur les côtés droits. La variable x ne l'est pas sur le côté droit de la dernière définition : on prend la même variable x que dans le côté gauche.

On admettra dans la suite que pour tous termes u, v, pour toute valeur V :

(A) $(u)^*k\ [x := V^\dagger] = (u[x := V])^*(k[x := V^\dagger])$.

(A') si x n'est pas libre dans u, alors $(u)^*k\ [x := v] = (u)^*(k[x := v])$.

(J'économise des parenthèses : $(u)^*k\ [x := V^\dagger]$ signifie $((u)^*k)[x := V^\dagger]$ par exemple.)

2. [long] Montrer que si $u \to v$ en λ-calcul par valeur, alors $(u)^*k \to^+ (v)^*k$ par nom.

Il faut pour cela démontrer un lemme intermédiaire : (B) si $k \to k'$ (par nom), alors $(u)^*k \to (u)^*k'$. On le démontre par récurrence sur u. Si u est une valeur V, alors $(u)^*k = k(V^\dagger) \to k'(V^\dagger) = (u)^*k'$. Sinon, u est de la forme vw, et $(u)^*k = (v)^*(\lambda f.(w)^*(fk))$. Comme $fk \to fk'$, par hypothèse de récurrence $\lambda f.(w)^*(fk) \to \lambda f.(w)^*(fk')$, et par hypothèse de récurrence de nouveau $(v)^*(\lambda f.(w)^*(fk)) \to (v)^*(\lambda f.(w)^*(fk')) = (u)^*k'$.

On montre maintenant la propriété demandée par récurrence sur la profondeur du rédex contracté.

Pour le cas de base, on a :

$$\begin{array}{rcl} ((\lambda x.u)V)^*k & = & (\lambda x.u)^*(\lambda f.(V)^*(fk)) \\ & = & (\lambda f.(V)^*(fk))(\lambda x.u)^\dagger \\ & \to & (V)^*(fk)[f := (\lambda x.u)^\dagger] \\ & = & (V)^*((\lambda x.u)^\dagger k) \quad \textit{par } (A) \textit{ ou par } (A'), \end{array}$$

en utilisant par ailleurs que f n'est pas libre dans V, f étant fraîche. Or $(\lambda x.u)^\dagger k = (\lambda k', x.(u)^*k')k \to \lambda x.(u)^*k'[k' := k] = \lambda x.(u)^*k$ par (A'), k' étant fraîche. Donc, en utilisant (B), $((\lambda x.u)V)^*k \to^2 (V)^*(\lambda x.(u)^*k)$.

Or $(V)^*(\lambda x.(u)^*k) = (\lambda x.(u)^*k)((V)^\dagger) \to (u)^*k[x := V^\dagger] = (u[x := V])^*(k[x := V^\dagger])$ par (A). Comme x est fraîche, $k[x := V^\dagger] = k$, donc $((\lambda x.u)V)^*k \to^3 (u[x := V])^*k$.

On a trois cas de récurrence, qui sont :

— *si $u \to u'$ par valeur, alors :*

$$\begin{array}{rcl} (uv)^*k & = & (u)^*(\lambda f.(v)^*(fk)) \\ & \to^+ & (u')^*(\lambda f.(v)^*(fk)) = (u'v)^*k \end{array}$$

par hypothèse de récurrence ;

— *si $v \to v'$ par valeur, alors :*

$$\begin{array}{rcl} (uv)^*k & = & (u)^*(\lambda f.(v)^*(fk)) \\ & \to^+ & (u)^*(\lambda f.(v')^*(fk)) \quad \textit{par hypothèse de récurrence et } (B) \\ & = & (uv')^*k; \end{array}$$

— *si $u \to u'$ par valeur, alors :*

$$\begin{array}{rcl} (\lambda x.u)^*k & = & k(\lambda k', x.(u)^*k') \\ & \to^+ & k(\lambda k', x.(u')^*k') = (\lambda x.u')^*k \end{array}$$

par hypothèse de récurrence.

3. [variante du cours] On considère une variante du λ^*-calcul vu en cours, où l'unique règle est $(\lambda^* x.u)V \to u[x := V]$. On rappelle que la construction syntaxique $(\lambda^* x.u)V$ n'est *pas* une application, mais une astuce notationnelle pour une construction que l'on noterait `let` $x = V$ `in` u en Caml (permise uniquement lorsque V est une valeur, ici, c'est-à-dire une variable ou une abstraction, comme plus haut). Montrer que cette variante termine. (Vous pouvez vous baser sur des résultats du cours.) On admettra qu'elle est localement confluente : en déduire que le λ-calcul par valeur est confluent.

   Elle termine, parce que toute réduction dans la variante est une réduction dans le λ^*-calcul du cours, qui termine.

   Le reste de la démonstration est comme dans le cours. Par le lemme de Newman, notre variante du λ^*-calcul est confluente, et tout terme u a une forme normale unique $u \downarrow$. De plus, $u \downarrow$ n'a pas d'étoile, et est donc un λ-terme.

   On pose $u \to_1 v$ si et seulement u est obtenu à partir d'un λ^*-terme en effaçant ses étoiles et $v = u \downarrow$. La relation $\to_1$ est alors fortement confluente, par le même argument que dans le cours.

   De plus, si $u \to v$ par valeur, alors $u \to_1 v$ et si $u \to_1 v$ alors $u \to^* v$ par valeur, donc le même argument que dans le cours implique que $\to$ est confluente.

2 Le λ-calcul avec clôtures explicites

On considère une extension du λ-calcul, qu'on appellera le λ_{S4}-calcul. Appelons *pré-termes* les expressions de syntaxe :

$$\begin{array}{rcll} s, t, u, v, \ldots & ::= & x, y, z, \ldots & \text{variables} \\ & | & uv & \text{applications} \\ & | & \lambda x.u & \text{abstractions} \\ & | & \boxed{s} \cdot \theta & \text{boîtes} \\ & | & \partial s & \text{évaluations} \end{array}$$

où θ est une *substitution explicite*, c'est-à-dire une fonction des variables vers les pré-termes de λ_{S4}, de domaine fini. Si on souhaite expliciter une telle substitution, on écrira $\{x_1 := t_1, x_2 := t_2, \ldots, x_n := t_n\}$ la substitution (explicite) qui à x_1 associe t_1, à x_2 associe t_2, …, x_n associe t_n. Il est entendu que x_1, x_2, …, x_n sont des variables deux à deux distinctes. On notera alors $\mathtt{dom}\,\theta$ l'ensemble $\{x_1, x_2, \ldots, x_n\}$; c'est le *domaine* de θ.

Comme en λ-calcul, la variable x est liée dans $\lambda x.u$. La nouveauté est que les variables $x_1, x_2, \ldots, x_n$ sont elles aussi toutes liées dans $\boxed{s} \cdot \theta$, où $\theta = \{x_1 := t_1, x_2 := t_2, \ldots, x_n := t_n\}$.

On identifie les pré-termes modulo α-renommage, qui est la plus petite congruence rendant égale les pré-termes à renommage de leurs variables liées près. Nous ne chercherons pas à rendre cette notion formelle, pour éviter la bureaucracie, mais supposerons qu'elle est comprise.

L'ensemble $\mathrm{fv}(s)$ des variables *libres* dans le pré-terme s est défini par récurrence structurelle sur s par :

$$\mathrm{fv}(x) = \{x\} \quad \mathrm{fv}(uv) = \mathrm{fv}(u) \cup \mathrm{fv}(v) \quad \mathrm{fv}(\lambda x.u) = \mathrm{fv}(u) \setminus \{x\}$$
$$\mathrm{fv}(\boxed{s} \cdot \theta) = \bigcup_{x \in \mathtt{dom}\,\theta} \mathrm{fv}(\theta(x)) \qquad \mathrm{fv}(\partial s) = \mathrm{fv}(s)$$

On note ici $\setminus$ la différence ensembliste, et $\theta(x)$ l'image par θ de x. Si $\theta = \{x_1 := t_1, x_2 := t_2, \ldots, x_n := t_n\}$, on a donc $\mathrm{fv}(\boxed{s} \cdot \theta) = \mathrm{fv}(t_1) \cup \mathrm{fv}(t_2) \cup \ldots \cup \mathrm{fv}(t_n)$.

Intuitivement, $\boxed{s} \cdot \theta$ est une boîte contenant le code qui permettrait de calculer s, et θ décrit les valeurs des variables libres de s. Ceci s'appelle aussi parfois une *clôture*. La construction ∂t, lorsque t est une telle boîte, permet de forcer l'évaluation du code contenu dans la boîte, et est similaire à l'opération `eval` de Lisp.

On appellera *terme* de λ_{S4} tout pré-terme dont toute sous-expression de la forme $\boxed{s} \cdot \theta$ est telle que $\mathrm{fv}(s) \subseteq \mathtt{dom}\,\theta$. Intuitivement, les variables libres de s sont contraintes à recevoir une valeur via θ. Aucune variable libre de s ne peut rester indéfinie par θ. (C'est la raison pour laquelle on n'a pas besoin d'écrire $\mathrm{fv}(\boxed{s} \cdot \theta) = (\mathrm{fv}(s) \setminus \mathtt{dom}\,\theta) \cup \bigcup_{x \in \mathtt{dom}\,\theta} \mathrm{fv}(\theta(x))$.)

Les règles de réduction sont :

$$\begin{array}{lrcl} (\beta) & (\lambda x.u)v & \to & u\{x := v\} \\ (\mathsf{unbox}) & \partial(\boxed{s} \cdot \theta) & \to & s\theta \end{array}$$

où la substitution $u\{x := v\}$ est un cas particulier de la notion de substitution $u\theta$ avec $\theta = \{x := v\}$, et $s\theta$ est la substitution parallèle, définie par récurrence sur la taille de s par :

$$x\theta = \begin{cases} \theta(x) & \text{si } x \in \mathtt{dom}\,\theta \\ x & \text{sinon} \end{cases}$$

$$(uv)\theta = (u\theta)(v\theta) \quad (\partial s)\theta = \partial(s\theta) \quad (\lambda x.u)\theta = \lambda x.(u\theta) \text{ (où } x \notin \mathtt{dom}\,\theta \cup \textstyle\bigcup_{z \in \mathtt{dom}\,\theta} \mathrm{fv}(\theta(z)))$$

$$(\boxed{s} \cdot \{x_1 := t_1, \ldots, x_n := t_n\})\theta = \boxed{s} \cdot \{x_1 := t_1\theta, \ldots, x_n := t_n\theta\}$$

Dans le cas de $(\lambda x.u)\theta$, l'opération ne sera donc définie qu'à α-renommage près.

On étend la discipline des types simples du λ-calcul en considérant d'abord une algèbre de types étendue :

$$F \quad ::= \quad b \mid F \to F \mid \Box F$$

où b parcourt un ensemble dit de types de base, et $\Box$ est un nouvel opérateur logique, "box".

Les règles de typage simple du λ_{S4}-calcul sont les suivantes :

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

$$\frac{\Gamma \vdash u : F \to G \quad \Gamma \vdash v : F}{\Gamma \vdash uv : G}\,(App) \qquad \frac{\Gamma, x : F \vdash u : G}{\Gamma \vdash \lambda x.u : F \to G}\,(Abs)$$

$$\frac{\Gamma \vdash u : \Box F}{\Gamma \vdash \partial u : F}\,(\Box E) \qquad \frac{x_1 : \Box F_1, \ldots, x_n : \Box F_n \vdash s : F \quad \overbrace{\Gamma \vdash t_i : \Box F_i}^{1 \leq i \leq n}}{\Gamma \vdash \boxed{s} \cdot \{x_1 := t_1, \ldots, x_n := t_n\} : \Box F}\,(\Box I)$$

Dans la règle $(\Box I)$, il y a $n+1$ prémisses. On fera très attention au fait que la prémisse de gauche ne s'applique que si toutes les variables x_i sont supposées de types dit *boxés*, c'est-à-dire commençant par un opérateur "box" $\Box$. Ceci définit une logique, la *logique modale S4*.

On dira qu'un terme est *neutre* s'il n'est ni une λ-abstraction ni une boîte $\boxed{s} \cdot \theta$.

On pose :

$$RED_b = SN \quad RED_{F_1 \to F_2} = REF_{F_1} \Rightarrow RED_{F_2}$$
$$RED_{\Box F} = \Box RED_F$$

où SN est l'ensemble des λ_{S4}-termes fortement normalisables, $S \Rightarrow S' = \{u \mid \forall v \in S.uv \in S'\}$, et $\Box S = \{u \mid \partial u \in S\}$. Ceci définit les ensembles RED_F par récurrence structurelle sur le type F.

On rappelle qu'un candidat est un ensemble 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 si tous les réduits en une étape de u sont dans S, alors $u \in S$.

On admettra que $S \Rightarrow S'$ est un candidat, pour tous candidats S et S'. La démonstration est comme dans le cours.

4. [variante du cours] Montrer que, pour tout candidat S, $\Box S$ est un candidat.

   (CR1) *Si $u \in \Box S$, c'est que $\partial u \in S$, donc par la propriété (CR1) de S, ∂u est dans SN. Ceci implique que le sous-terme u est encore dans SN.*

   (CR2) *Si $u \in \Box S$ et $u \to u'$, alors par définition ∂u est dans S. Comme $\partial u \to \partial u'$, par la propriété (CR2) de S, on a $\partial u' \in S$. Par définition, u' est donc dans $\Box S$.*

   (CR3) *Si u est neutre et pour tout u' tel que $u \to u'$, $u' \in S$, montrons que $u \in S$. Pour ceci, il suffit de montrer que ∂u est dans S. Comme ∂u est neutre, il suffit de montrer que tous les réduits en une étape de ∂u sont dans S, grâce à la propriété (CR3) de S. Or, u étant neutre, n'est pas une boîte. Les seuls réduits en une étape de ∂u sont donc de la forme $\partial u'$, avec $u \to u'$. Par hypothèse, u' est dans S, donc par définition $\partial u'$ est dans $\Box S$, et l'on conclut.*

On admettra aussi la propriété :

$(*)$ si $u\{x := v\} \in RED_G$ pour tout $v \in RED_F$, alors $\lambda x.u \in RED_{F \to G}$.

Elle se démontre comme dans le cours.

5. [variante du cours] Pour tout contexte Δ de la forme $x_1 : F_1, \ldots, x_n : F_n$, on notera $\overline{RED}_\Delta$ l'ensemble des substitutions θ de la forme $\{x_1 := t_1, \ldots, x_n := t_n\}$, où $t_1 \in RED_{F_1}, \ldots, t_n \in RED_{F_n}$. Montrer que, si : $(**)$ $s\theta$ est dans RED_F pour toute substitution $\theta \in \overline{RED}_\Delta$, alors $\boxed{s} \cdot \theta$ est dans $RED_{\Box F}$ pour toute substitution $\theta \in \overline{RED}_\Delta$.

   Il suffit de montrer que $\partial(\boxed{s} \cdot \theta)$ est dans RED_F, par (CR3). La substitution $\{x_1 := x_1, \ldots, x_n := x_n\}$ étant dans $\overline{RED}_\Delta$, toute variable étant réductible à tout type par (CR3), l'hypothèse $(**)$ et (CR1) impliquent que s est fortement normalisable.

   De même, tous les t_i sont fortement normalisables par (CR1). On peut donc démontrer que $\partial(\boxed{s} \cdot \theta)$ est dans RED_F pour tout s vérifiant $(**)$ et pour tout $\theta = \{x_1 := t_1, \ldots, x_n := t_n\}$ dans $\overline{RED}_\Delta$, par récurrence sur $(s, t_1, \ldots, t_n)$ le long de $\nu(s) + \sum_{i=1}^n \nu(t_i)$ par exemple. Pour tout réduit en une étape de $\partial(\boxed{s} \cdot \theta)$, soit il est *obtenu en contractant un rédex dans s ou dans l'un des t_i, et l'on conclut en utilisant (CR2) et l'hypothèse de récurrence ; soit ce réduit est $s\theta$, qui est dans RED_F par hypothèse.*

6. [variante du cours] En déduire que tout terme u de λ_{S4} qui est typable est fortement normalisable.

On montre que, pour toute dérivation π de $\Gamma \vdash u : F$, pour toute substitution $\theta \in \overline{RED}_\Gamma$, $u\theta$ est dans RED_F. Comme la substitution identité est dans $\overline{RED}_\Gamma$ par (CR3), ceci permettra de déduire que u lui-même sera dans RED_F, donc dans SN par (CR1).
Si la dernière règle est (Var), (App) ou (Abs), on procède comme dans le poly. Le cas de (Abs) notamment est réglé par $(*)$.
Si la dernière règle est $(\Box E)$, on sait par hypothèse de récurrence que $u = \partial u'$ avec $u'\theta \in RED_{\Box F}$. Mais ceci signifie directement que $u\theta$ est dans RED_F, par définition de $RED_{\Box F}$.
Si la dernière règle est $(\Box I)$, u est de la forme $\boxed{s} \cdot \theta$, $F = \Box F'$, et l'on a dérivé :

$$\dfrac{x_1 : \Box F_1, \ldots, x_n : \Box F_n \vdash s : F \qquad \underbrace{\begin{array}{c} \vdots\, \pi_i \\ \Gamma \vdash t_i : \Box F_i \end{array}}_{1 \leq i \leq n}}{\Gamma \vdash \boxed{s} \cdot \{x_1 := t_1, \ldots, x_n := t_n\} : \Box F} \ (\Box I)$$

Par hypothèse de récurrence, $t_i\theta$ est dans $RED_{\Box F_i}$ pour tout i. En notant Δ le contexte $x_1 : \Box F_1, \ldots, x_n : \Box F_n$, la substitution $\{x_1 := t_1\theta, \ldots, x_n := t_n\theta\}$ est donc dans $\overline{RED}_\Delta$. Par hypothèse de récurrence de nouveau, s vérifie l'hypothèse $(**)$ de la question 5, donc $\boxed{s} \cdot \theta'$ est dans $RED_{\Box F}$ pour toute substitution $\theta' \in \overline{RED}_\Delta$. C'est en particulier le cas pour $\theta' = \{x_1 := t_1\theta, \ldots, x_n := t_n\theta\}$, d'où la conclusion.

3 La traduction de Gödel-Tarski-Givant

On traduit toute formule F de logique (minimale) intuitionniste en une formule F° de S4 comme suit :

$$b^\circ = \Box b \qquad (F \to G)^\circ = \Box(F^\circ \to G^\circ)$$

Autrement dit, on ajoute des $\Box$ devant toute sous-formule. On rappelle que les types de la logique minimale sont ceux des types simples, c'est-à-dire tous ceux qui ne contiennent pas $\Box$. Un λ-terme est aussi vu comme un λ_{S4}-terme qui ne contient ni boîte ni opérateur ∂.

Pour tout contexte de typage $\Gamma = x_1 : F_1, \ldots, x_n : F_n$, on note Γ° le contexte $x_1 : F_1^\circ, \ldots, x_n : F_n^\circ$. On pose :

$$x^\circ = x \quad (uv)^\circ = (\partial u^\circ)v^\circ \quad (\lambda x.u)^\circ = \boxed{(\lambda x.u^\circ)} \cdot id$$

où id est la substitution identité qui associe à chaque variable libre de $\lambda x.u$, elle-même.

7. [long] Montrer que si $\Gamma \vdash t : F$ est dérivable dans la discipline de types simples, alors $\Gamma^\circ \vdash t^\circ : F^\circ$ est dérivable dans le système de typage de λ_{S4}. Donc la traduction $F \mapsto F^\circ$ préserve la prouvabilité. (Réciproquement, si F° est prouvable, et ce en logique S4 intuitionniste ou même classique, alors F est prouvable en logique intuitionniste. Ceci ne sera pas utilisé dans la suite.)

On démontre le résultat souhaité par récurrence sur la dérivation de typage de $\Gamma \vdash t : F$. Si t est une variable, c'est évident. Si t est une application uv, c'est que la dérivation se termine par :

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ \Gamma \vdash u : G \Rightarrow F\end{array} \quad \begin{array}{c}\vdots\,\pi_2\\ \Gamma \vdash v : G\end{array}}{\Gamma \vdash uv : F}\,(App)$$

ce que l'on traduit en :

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_1^\circ\\ \Gamma^\circ \vdash u^\circ : \Box(G^\circ \Rightarrow F^\circ)\end{array}}{\Gamma^\circ \vdash \partial u^\circ : G^\circ \Rightarrow F^\circ}\,(\Box E) \quad \begin{array}{c}\vdots\,\pi_2^\circ\\ \Gamma^\circ \vdash v^\circ : G^\circ\end{array}}{\Gamma^\circ \vdash (\partial u^\circ)v^\circ : F^\circ}\,(App)$$

où π_1°, π_2° sont obtenues par hypothèse de récurrence.
Si t est de la forme $\lambda x.u$ et $F = F_1 \Rightarrow F_2$, c'est-à-dire que la dérivation de type se termine par :

$$\dfrac{\begin{array}{c}\vdots\,\pi\\ \Gamma, x : F_1 \vdash u : F_2\end{array}}{\Gamma \vdash \lambda x.u : F_1 \Rightarrow F_2}\,(Abs)$$

alors on produit la dérivation suivante, où $x_1, \ldots, x_n$ sont les variables libres de u. D'abord, par hypothèse de récurrence, on obtient une dérivation π° de $\Gamma^\circ, x : F_1^\circ \vdash u^\circ : F_2^\circ$. En appliquant (Abs), on obtient une dérivation de $\Gamma^\circ \vdash \lambda x.u^\circ : F_1^\circ \Rightarrow F_2^\circ$. Par affaiblissement éventuel, on peut supposer que Γ° contient au moins un contexte de la forme $x_1 : G_1^\circ, \ldots, x_n : G_n^\circ$. (En fait, on peut démontrer qu'il contient un tel contexte, nécessairement.)
La logique S4 intuitionniste a d'autre part la propriété, opposée de l'affaiblissement, et appelée amincissement *: si $\Delta, z : G \vdash w : H$ est dérivable et z n'est pas libre dans w, alors $\Delta \vdash w : H$ est aussi dérivable. C'est une récurrence facile sur la structure de la dérivation, similaire à la démonstration de la propriété d'affaiblissement.*
Par amincissement, on obtient ainsi une dérivation π_1 de $x_1 : G_1^\circ, \ldots, x_n : G_n^\circ \vdash \lambda x.u^\circ : F_1^\circ \Rightarrow F_2^\circ$. Comme tous les types G_1°, …, G_n° commencent par $\Box$, on peut

appliquer $(\Box I)$ *:*

$$\dfrac{\begin{array}{c}\vdots\\ x_1 : G_1^\circ, \ldots, x_n : G_n^\circ \vdash \lambda x.u^\circ : F_1^\circ \Rightarrow F_2^\circ\end{array} \quad \overbrace{\dfrac{}{\Gamma^\circ \vdash x_i : G_1^\circ}}^{1\leq i\leq n}}{\Gamma^\circ \vdash \boxed{(\lambda x.u^\circ)} \cdot id}$$

8. [facile] Montrer qu'en revanche, la traduction ne préserve pas la réduction : il existe des λ-termes s, t tels que $s \to t$ en λ-calcul, mais on n'a pas $s^\circ \to^* t^\circ$.

   Soit $s = (\lambda x.u)v$, $t = u\{x := v\}$. *Le problème est que* $s^\circ = (\partial \boxed{(\lambda x.u^\circ)} \cdot id)v^\circ$ *se réduit en* $u^\circ\{x := v^\circ\}$, *mais ce dernier terme ne se réduit pas forcément en* $(u\{x := v\})^\circ$. *Si* u *est lui-même une* λ*-abstraction, par exemple* $\lambda y.x$, $u^\circ\{x := v^\circ\} = \boxed{\lambda y.x} \cdot \{x := v^\circ\}$, *alors que* $(u\{x := v\})^\circ = \boxed{\lambda y.v^\circ} \cdot id$. *Si* v *n'est pas une variable, ces deux termes sont différents, et si* v *est normal,* v° *aussi. Un contre-exemple est donc donné par* $s = (\lambda x.\lambda y.x)(zz')$.

On ajoute au λ_{S4}-calcul la règle de réduction suivante, dite de *promotion* :

$$\boxed{u} \cdot (\theta \uplus \{x := \boxed{v} \cdot \theta'\}) \to \boxed{u[x := \boxed{v} \cdot id]} \cdot (\theta \uplus \theta')$$

où $\theta \uplus \theta'$ est l'union disjointe de θ et θ' — disjointe au sens où l'on suppose les variables liées renommées de sorte à ce que $\mathtt{dom}\,\theta \cap \mathtt{dom}\,\theta' = \emptyset$.

Même avec ces nouvelles règles, il existe des λ-termes s, t tels que $s \to t$ en λ-calcul, mais on n'a pas $s^\circ \to^* t^\circ$.

9. [sans difficulté] Montrer que $s^\circ \to^+ t^\circ$ dans le nouveau calcul dès que $s \to t$ en λ-calcul *par valeur*.

   On a :

$$\begin{array}{rcl} ((\lambda x.u)V)^\circ & = & \partial(\boxed{\lambda x.u^\circ} \cdot id)V^\circ \\ & \to & (\lambda x.u^\circ)V^\circ \\ & \to & u^\circ[x := V^\circ] \end{array}$$

   et tout le problème consiste à montrer que si V *est une valeur, alors* $u^\circ[x := V^\circ] \to^* (u[x := V])^\circ$.

   Ceci se fait par récurrence sur la structure de u. *Tous les cas sont évidents, sauf celui de la* λ*-abstraction* $u = \lambda y.w$ *:*

$$\begin{array}{rcl} u^\circ[x := V^\circ] & = & \boxed{\lambda y.w^\circ} \cdot id[x := V^\circ] \\ & = & \boxed{\lambda y.w^\circ} \cdot id \uplus \{x := V^\circ\} \end{array}$$

   où dans ce dernier cas id *porte sur toutes les variables libres de* $\lambda y.w^\circ$ *sauf* x. *Si* V *est une variable* z, *ceci est égal à* $\boxed{\lambda y.w^\circ[x := z]} \cdot id$ *par* α*-renommage. Si* V *est* *une λ-expression $\lambda z.v$, $V^\circ = \boxed{\lambda z.v} \cdot id$, donc la nouvelle règle de promotion permet d'écrire :*

$$u^\circ[x := V^\circ] \quad \to \quad \boxed{\lambda y.w^\circ[x := V^\circ]} \cdot id$$

qui se réécrit par hypothèse de récurrence en $\boxed{\lambda y.(w[x := V])^\circ} \cdot id = (u[x := V])^\circ$.