Typage et programmation
Corrigé 2002-2003

François Pottier
Francois.Pottier@inria.fr

19 décembre 2002

Question 1 On définit

$$\begin{aligned}
\mathrm{ftv}(\alpha) &= \{\alpha\} \\
\mathrm{ftv}(\mathrm{int}) &= \varnothing \\
\mathrm{ftv}(\tau_1 \to \tau_2) &= \mathrm{ftv}(\tau_1) \cup \mathrm{ftv}(\tau_2) \\
\mathrm{ftv}(\forall\alpha.\tau) &= \mathrm{ftv}(\tau) \setminus \{\alpha\} \\
\mathrm{ftv}(\bullet) &= \varnothing \\
\mathrm{ftv}(\Gamma; x : \tau) &= \mathrm{ftv}(\Gamma) \cup \mathrm{ftv}(\tau)
\end{aligned}$$

Question 2 Voici une dérivation de typage pour id :

$$\dfrac{\dfrac{x : \alpha \vdash_{\mathrm{F}} x : \alpha}{\bullet \vdash_{\mathrm{F}} \lambda x : \alpha.x : \alpha \to \alpha}\ \textsc{f-Abs}}{\bullet \vdash_{\mathrm{F}} \mathrm{id} : \forall\alpha.\alpha \to \alpha}\ \textsc{f-TAbs}$$

Soit Γ l'environnement $f : \alpha \to \alpha; x : \alpha$. Voici une dérivation de typage pour double :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash_{\mathrm{F}} f : \alpha \to \alpha \qquad \dfrac{\Gamma \vdash_{\mathrm{F}} f : \alpha \to \alpha \quad \Gamma \vdash_{\mathrm{F}} x : \alpha}{\Gamma \vdash_{\mathrm{F}} f\,x : \alpha}\ \textsc{f-App}}{\Gamma \vdash_{\mathrm{F}} f\,(f\,x) : \alpha}\ \textsc{f-App}}{f : \alpha \to \alpha \vdash_{\mathrm{F}} \lambda x : \alpha.f\,(f\,x) : \alpha \to \alpha}\ \textsc{f-Abs}}{\bullet \vdash_{\mathrm{F}} \lambda f : \alpha \to \alpha.\lambda x : \alpha.f\,(f\,x) : (\alpha \to \alpha) \to \alpha \to \alpha}\ \textsc{f-Abs}}{\bullet \vdash_{\mathrm{F}} \mathrm{double} : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha}\ \textsc{f-TAbs}$$

Voici enfin une dérivation de typage pour zero :

$$\dfrac{\dfrac{\dfrac{\bullet \vdash_{\mathrm{F}} \mathrm{double} : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha}{\bullet \vdash_{\mathrm{F}} \mathrm{double}\,\mathrm{int} : (\mathrm{int} \to \mathrm{int}) \to \mathrm{int} \to \mathrm{int}}\ \textsc{f-TApp} \qquad \dfrac{\bullet \vdash_{\mathrm{F}} \mathrm{id} : \forall\alpha.\alpha \to \alpha}{\bullet \vdash_{\mathrm{F}} \mathrm{id}\,\mathrm{int} : \mathrm{int} \to \mathrm{int}}\ \textsc{f-TApp}}{\bullet \vdash_{\mathrm{F}} \mathrm{double}\,\mathrm{int}\,(\mathrm{id}\,\mathrm{int}) : \mathrm{int} \to \mathrm{int}}\ \textsc{f-App} \qquad \bullet \vdash_{\mathrm{F}} 0 : \mathrm{int}}{\bullet \vdash_{\mathrm{F}} \mathrm{zero} : \mathrm{int}}\ \textsc{f-App}$$

D'après la définition des valeurs dans le système F, toute série de Λ terminée par un λ est une valeur. Ainsi, id et double sont des valeurs ; on peut vérifier qu'elles ne se réduisent pas. Pour zero, on peut effectuer la série de réductions suivante :

$$\begin{array}{lcll}
\text{zero} & = & \text{double int (id int) } 0 & \\
& \to_{\text{F}} & (\lambda f : \text{int} \to \text{int}.\lambda x : \text{int}.f\ (f\ x))\ (\text{id int})\ 0 & (\beta\textit{-type}) \\
& \to_{\text{F}} & (\lambda f : \text{int} \to \text{int}.\lambda x : \text{int}.f\ (f\ x))\ (\lambda x : \text{int}.x)\ 0 & (\beta\textit{-type}) \\
& \to_{\text{F}} & (\lambda x : \text{int}.(\lambda x : \text{int}.x)\ ((\lambda x : \text{int}.x)\ x))\ 0 & (\beta\textit{-valeur}) \\
& \to_{\text{F}} & (\lambda x : \text{int}.x)\ ((\lambda x : \text{int}.x)\ 0) & (\beta\textit{-valeur}) \\
& \to_{\text{F}} & (\lambda x : \text{int}.x)\ 0 & (\beta\textit{-valeur}) \\
& \to_{\text{F}} & 0 & (\beta\textit{-valeur})
\end{array}$$

Question 3 Dans l'environnement $y : \alpha$, la règle F-TABS ne s'applique pas directement à l'expression $\Lambda\alpha.\lambda x : \alpha.y$, car la variable α apparaît libre dans l'environnement, violant la seconde prémisse. Cependant, on peut appliquer F-TABS à une expression α-équivalente. Soit β une variable de types distincte de α. On peut écrire

$$\dfrac{\dfrac{y : \alpha; x : \beta \vdash_{\text{F}} y : \alpha}{y : \alpha \vdash_{\text{F}} \lambda x : \beta.y : \beta \to \alpha}\ \text{F-ABS} \qquad \beta \notin \text{ftv}(y : \alpha)}{y : \alpha \vdash_{\text{F}} \Lambda\beta.\lambda x : \beta.y : \forall\beta.\beta \to \alpha}\ \text{F-TABS}$$

Le type attendu est donc $\forall\beta.\beta \to \alpha$. Notons que, puisque β est liée dans ce type, il ne dépend pas du choix de β. □

Question 4 Supposons $x : \forall\alpha_1 \ldots \alpha_n.\tau \in \Gamma$. Soient $\tau_1, \ldots, \tau_n$ des types arbitraires donnés. Notre but est de démontrer que le jugement $\Gamma \vdash_{\text{F}} x\,\tau_1\ \ldots\ \tau_n : \tau[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]$ est dérivable dans le système F.

Nous allons d'abord vérifier que l'on peut supposer, sans perte de généralité, que les variables $\alpha_1, \ldots, \alpha_n$ n'apparaissent pas libres dans les types $\tau_1, \ldots, \tau_n$. En effet, soient $\beta_1, \ldots, \beta_n$ des variables distinctes deux à deux et n'apparaissant pas libres dans $\alpha_1, \ldots, \alpha_n$, dans τ ni dans $\tau_1, \ldots, \tau_n$. Soit ρ la transposition $(\bar{\alpha}\,\bar{\beta})$, c'est-à-dire le renommage défini par $\rho(\alpha_i) = \beta_i$ et $\rho(\beta_i) = \alpha_i$ pour tout $i \in \{1, \ldots, n\}$. Alors, les types $\forall\alpha_1 \ldots \alpha_n.\tau$ et $\forall\beta_1 \ldots \beta_n.\rho(\tau)$ sont α-équivalents. De plus notre but peut s'écrire sous la forme $\Gamma \vdash_{\text{F}} x\,\tau_1\ \ldots\ \tau_n : \rho(\tau)[\tau_1/\beta_1, \ldots, \tau_n/\beta_n]$. Nous nous sommes donc ramenés à une situation identique à la situation initiale, à cela près que les variables $\beta_1, \ldots, \beta_n$ vérifient l'hypothèse supplémentaire souhaitée. Dans ce qui suit, nous conservons donc les notations initiales, mais supposons que les variables $\alpha_1, \ldots, \alpha_n$ n'apparaissent pas libres dans les types $\tau_1, \ldots, \tau_n$.

Nous commençons par la déduction suivante :

$$\dfrac{\Gamma \vdash_{\text{F}} x : \forall\alpha_1.\ldots.\forall\alpha_n.\tau}{\Gamma \vdash_{\text{F}} x\,\tau_1 : (\forall\alpha_2.\ldots.\forall\alpha_n.\tau)[\tau_1/\alpha_1]}\ \text{F-TAPP}$$

Du fait que les variables $\alpha_1, \ldots, \alpha_n$ sont supposées deux à deux distinctes, et du fait que $\alpha_2, \ldots, \alpha_n$ n'apparaissent pas libres dans τ_1, la conclusion de cette dérivation peut s'écrire $\Gamma \vdash_{\text{F}} x\,\tau_1 : \forall\alpha_2.\ldots.\forall\alpha_n.(\tau[\tau_1/\alpha_1])$. Nous pouvons donc à nouveau appliquer F-TAPP à ce jugement, pour obtenir

$$\Gamma \vdash_{\text{F}} x\,\tau_1\,\tau_2 : \forall\alpha_3.\ldots.\forall\alpha_n.(\tau[\tau_1/\alpha_1][\tau_2/\alpha_2]),$$

et ainsi de suite jusqu'à

$$\Gamma \vdash_{\text{F}} x\,\tau_1\,\ldots\,\tau_n : \tau[\tau_1/\alpha_1]\,\ldots\,[\tau_n/\alpha_n].$$

Il reste à remarquer que, du fait que les variables $\alpha_1, \ldots, \alpha_n$ n'apparaissent pas libres dans les types $\tau_1, \ldots, \tau_n$, la substitution composée $[\tau_n/\alpha_n] \circ \ldots \circ [\tau_1/\alpha_1]$ coïncide avec la substitution simultanée $[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]$. Le jugement obtenu est donc notre but. □

Question 5 Supposons $\Gamma \vdash_{\text{F}} a : \tau$ et $\{\alpha_1, \ldots, \alpha_n\} = \text{ftv}(\tau) \setminus \text{ftv}(\Gamma)$. Nous pouvons construire la dérivation

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash_{\text{F}} a : \tau \qquad \alpha_n \notin \text{ftv}(\Gamma)}{\Gamma \vdash_{\text{F}} \Lambda\alpha_n.a : \forall\alpha_n.\tau}\ \text{F-TABS}}{\vdots}}{\Gamma \vdash_{\text{F}} \Lambda\alpha_2.\ldots.\Lambda\alpha_n.a : \forall\alpha_2.\ldots.\forall\alpha_n.\tau \qquad \alpha_1 \notin \text{ftv}(\Gamma)}\ \text{F-TABS}$$
$$\Gamma \vdash_{\text{F}} \Lambda\alpha_1.\ldots.\Lambda\alpha_n.a : \forall\alpha_1.\ldots.\forall\alpha_n.\tau$$

Question 6 Les deux questions précédentes suggèrent de traduire l'accès à une variable de ML par l'accès à une variable suivi d'une série d'applications de types, et de traduire la généralisation de ML par une série d'abstractions de types. Par ailleurs, la construction let, qui n'existe pas dans le système F, peut être traduite par un simple β-redex. Nous posons donc :

$$\dfrac{x : \forall\alpha_1 \ldots \alpha_n.\tau \in \Gamma}{\Gamma \vdash x \rightsquigarrow x\,\tau_1\,\ldots\,\tau_n : \tau[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]} \qquad \Gamma \vdash k \rightsquigarrow k : \text{int}$$

$$\dfrac{\Gamma; x : \tau' \vdash e \rightsquigarrow a : \tau}{\Gamma \vdash \lambda x.e \rightsquigarrow \lambda x : \tau'.a : \tau' \to \tau} \qquad \dfrac{\Gamma \vdash e \rightsquigarrow a : \tau' \to \tau \qquad \Gamma \vdash e' \rightsquigarrow a' : \tau'}{\Gamma \vdash e\,e' \rightsquigarrow a\,a' : \tau}$$

$$\dfrac{\Gamma \vdash e \rightsquigarrow a : \tau \qquad \bar{\alpha} = \text{ftv}(\tau) \setminus \text{ftv}(\Gamma) \qquad \Gamma; x : \forall\bar{\alpha}.\tau \vdash e' \rightsquigarrow a' : \tau'}{\Gamma \vdash \text{let } x = e \text{ in } e' \rightsquigarrow (\lambda x : \forall\bar{\alpha}.\tau.a')\,(\Lambda\bar{\alpha}.a) : \tau'}$$

Il est facile de vérifier, par induction sur la structure de e, que $\Gamma \vdash_{\text{ML}} e : \tau$ est dérivable si et seulement si il existe une expression a telle que $\Gamma \vdash e \rightsquigarrow a : \tau$ est dérivable. Intuitivement, cela provient du fait que les règles qui définissent le nouveau jugement imposent exactement les mêmes conditions sur Γ, e et τ que les règles de typage de ML, et se contentent de spécifier de surcroît la forme de a.

Reste à démontrer que $\Gamma \vdash e \rightsquigarrow a : \tau$ implique $\Gamma \vdash_{\text{F}} a : \tau$, ce que nous effectuons par induction sur la forme de e.

- *Cas* d'une variable x. Le résultat a été établi par la question 4.
- *Cas* d'un entier k. Immédiat.
- *Cas* d'une abstraction $\lambda x.e$. Nous disposons donc d'un jugement de la forme $\Gamma \vdash \lambda x.e \rightsquigarrow \lambda x : \tau'.a : \tau' \to \tau$, lequel découle d'une prémisse de la forme $\Gamma; x : \tau' \vdash e \rightsquigarrow a : \tau$. L'hypothèse d'induction, appliquée à cette prémisse, donne $\Gamma; x : \tau' \vdash_{\text{F}} a : \tau$. Par application de la règle F-ABS, il en découle $\Gamma \vdash_{\text{F}} \lambda x : \tau'.a : \tau' \to \tau$, qui est le résultat souhaité.

∘ *Cas* d'une application $e\,e'$. Nous disposons donc d'un jugement de la forme $\Gamma \vdash e\,e' \rightsquigarrow a\,a' : \tau$, lequel découle de prémisses de la forme $\Gamma \vdash e \rightsquigarrow a : \tau' \to \tau$ et $\Gamma \vdash e' \rightsquigarrow a' : \tau'$. L'hypothèse d'induction, appliquée à celles-ci, donne $\Gamma \vdash_{\text{F}} a : \tau' \to \tau$ et $\Gamma \vdash_{\text{F}} a' : \tau'$. Par application de la règle F-App, il en découle $\Gamma \vdash_{\text{F}} a\,a' : \tau$, qui est le résultat souhaité.

∘ *Cas* d'une liaison locale let $x = e$ in e'. Nous disposons donc d'un jugement de la forme $\Gamma \vdash \text{let } x = e \text{ in } e' \rightsquigarrow (\lambda x : \forall\bar{\alpha}.\tau.a')\,(\Lambda\bar{\alpha}.a) : \tau'$, lequel découle des prémisses $\Gamma \vdash e \rightsquigarrow a : \tau$ et $\bar{\alpha} = \text{ftv}(\tau) \setminus \text{ftv}(\Gamma)$ et $\Gamma; x : \forall\bar{\alpha}.\tau \vdash e' \rightsquigarrow a' : \tau'$. L'hypothèse d'induction, appliquée à la première prémisse, donne $\Gamma \vdash_{\text{F}} a : \tau$. De ceci, et de la deuxième prémisse, on déduit, grâce au résultat établi par la question 5, le jugement $\Gamma \vdash_{\text{F}} \Lambda\bar{\alpha}.a : \forall\bar{\alpha}.\tau$. Par ailleurs, l'hypothèse d'induction, appliquée à la troisième prémisse, donne $\Gamma; x : \forall\bar{\alpha}.\tau \vdash_{\text{F}} a' : \tau'$, d'où l'on dérive, à l'aide de la règle F-Abs, le jugement $\Gamma \vdash_{\text{F}} \lambda x : \forall\bar{\alpha}.\tau.a' : (\forall\bar{\alpha}.\tau) \to \tau'$. Nous sommes donc à même de construire la dérivation suivante :

$$\frac{\Gamma \vdash_{\text{F}} \lambda x : \forall\bar{\alpha}.\tau.a' : (\forall\bar{\alpha}.\tau) \to \tau' \qquad \Gamma \vdash_{\text{F}} \Lambda\bar{\alpha}.a : \forall\bar{\alpha}.\tau}{\Gamma \vdash_{\text{F}} (\lambda x : \forall\bar{\alpha}.\tau.a')\,(\Lambda\bar{\alpha}.a) : \tau'}\ \text{F-App}$$

La conclusion de cette dérivation n'est autre que le résultat souhaité. □

Question 7 On pose naturellement

$$\begin{aligned}
[\![x]\!] &= x \\
[\![\lambda x : \tau.a]\!] &= \lambda x.[\![a]\!] \\
[\![a_1\,a_2]\!] &= [\![a_1]\!]\,[\![a_2]\!] \\
[\![\Lambda\alpha.a]\!] &= [\![a]\!] \\
[\![a\,\tau]\!] &= [\![a]\!]
\end{aligned}$$

Question 8 Démontrons par induction que l'effacement d'une valeur est une valeur. Deux cas se présentent : d'abord, on a $[\![\lambda x : \tau.a]\!] = \lambda x.[\![a]\!]$, qui est une valeur ; ensuite, on a $[\![\Lambda\alpha.w]\!] = [\![w]\!]$, qui est une valeur par hypothèse d'induction.

Démontrons la première assertion de commutation entre effacement et substitution, dans les deux cas requis. D'abord, nous avons $[\![x[w/x]]\!] = [\![w]\!] = x[[\![w]\!]/x]$. Par ailleurs, en supposant (sans perte de généralité) y distinct de x et non libre dans w, nous avons

$$\begin{array}{rcll}
[\![(\lambda y : \tau.a)[w/x]]\!] & = & [\![\lambda y : \tau.(a[w/x])]\!] & \text{par déf. de la substitution} \\
& = & \lambda y.[\![(a[w/x])]\!] & \text{par déf. de } [\![\cdot]\!] \\
& = & \lambda y.([\![a]\!][[\![w]\!]/x]) & \text{par hyp. d'induction} \\
& = & (\lambda y.[\![a]\!])[[\![w]\!]/x] & \text{par déf. de la substitution} \\
& = & [\![\lambda y : \tau.a]\!][[\![w]\!]/x] & \text{par déf. de } [\![\cdot]\!]
\end{array}$$

Démontrons à présent la seconde assertion de commutation entre effacement et substitution, dans les deux cas requis. D'abord, nous avons $[\![x[\tau/\alpha]]\!] = x = [\![x]\!]$. Par ailleurs, nous avons

$$\begin{array}{rcll}
[\![(\lambda y : \tau'.a)[\tau/\alpha]]\!] & = & [\![\lambda y : \tau'[\tau/\alpha].(a[\tau/\alpha])]\!] & \text{par déf. de la substitution} \\
& = & \lambda y.[\![a[\tau/\alpha]]\!] & \text{par déf. de } [\![\cdot]\!] \\
& = & \lambda y.[\![a]\!] & \text{par hyp. d'induction} \\
& = & [\![\lambda y : \tau'.a]\!] & \text{par déf. de } [\![\cdot]\!]
\end{array}$$

Question 9 Deux cas se présentent, selon que la réduction est une instance de (β-*valeur*) ou de (β-*type*). Dans le premier cas, on doit vérifier que $[\![(\lambda x : \tau.a)\, w]\!]$ se réduit en $[\![a[w/x]]\!]$. Par définition de $[\![\cdot]\!]$ et d'après le second résultat établi par la question précédente, cela revient à vérifier que $(\lambda x.[\![a]\!])\, [\![w]\!]$ se réduit en $[\![a]\!][[\![w]\!]/x]$, ce qui effectivement une instance de (β), puisque $[\![w]\!]$ est une valeur. Dans le second cas, on doit vérifier que $[\![(\Lambda\alpha.a)\, \tau]\!]$ se réduit en $[\![a[\tau/\alpha]]\!]$. Par définition de $[\![\cdot]\!]$ et d'après le troisième résultat établi par la question précédente, cela revient à vérifier que $[\![a]\!]$ se réduit en lui-même, ce qui est vrai ; il s'agit d'une réduction en zéro étapes. □

Question 10 On définit bien sûr $[\![[\,]]\!] = [\,]$. Cette définition constitue le cas de base lorsque nous démontrons, par induction sur la structure de A, que $[\![A]\!]$ est un contexte d'évaluation. Les autres cas sont les suivants. D'abord, $[\![A\, a]\!] = [\![A]\!]\, [\![a]\!]$ est un contexte d'évaluation, car (par hypothèse d'induction) $[\![A]\!]$ en est un. Ensuite, $[\![w\, A]\!] = [\![w]\!]\, [\![A]\!]$ est un contexte d'évaluation, car (par hypothèse d'induction) $[\![A]\!]$ en est un et car $[\![w]\!]$ est une valeur. Enfin, $[\![\Lambda\alpha.A]\!] = [\![A\, \tau]\!] = [\![A]\!]$ sont des contextes d'évaluation par hypothèse d'induction.

Considérons à présent une réduction quelconque dans le système F. D'après (*contexte*), celle-ci est de la forme $A[a] \to_{\mathrm{F}} A[a']$, où $a \xrightarrow{e}_{\mathrm{F}} a'$. D'après la question précédente, $[\![a]\!]$ se réduit (en zéro ou une étape) en $[\![a']\!]$. D'après ce qui précède, $[\![A]\!]$ est un contexte d'évaluation. Par conséquent, d'après (*contexte*), $[\![A]\!][[\![a]\!]]$ se réduit (en zéro ou une étape) en $[\![A]\!][[\![a']\!]]$. Il est facile de vérifier que l'application d'un contexte commute avec l'effacement ; ainsi, nous avons vérifié que $[\![A[a]]\!]$ se réduit (en zéro ou une étape) en $[\![A[a']]\!]$. □

Question 11 Supposons $a \to_{\mathrm{F}} a_1 \to_{\mathrm{F}} \ldots \to_{\mathrm{F}} k$. Alors, d'après les résultats précédents, nous avons $[\![a]\!] \to_{\mathrm{F}} [\![a_1]\!] \to_{\mathrm{F}} \ldots \to_{\mathrm{F}} [\![k]\!] = k$. □

Question 12 La démonstration se fait par induction sur la dérivation de $\Gamma \vdash e \rightsquigarrow a : \tau$.

∘ *Cas* d'une variable x. Le jugement considéré est $\Gamma \vdash x \rightsquigarrow x\, \tau_1 \ldots \tau_n : \tau[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n]$. Or nous avons $[\![x]\!] = [\![x\, \tau_1 \ldots \tau_n]\!]$, donc *a fortiori* $[\![x]\!] \approx [\![x\, \tau_1 \ldots \tau_n]\!]$.

∘ *Cas* d'un entier k. Immédiat.

∘ *Cas* d'une abstraction $\lambda x.e$. Le jugement considéré est $\Gamma \vdash \lambda x.e \rightsquigarrow \lambda x : \tau'.a : \tau' \to \tau$, et découle de la prémisse $\Gamma; x : \tau' \vdash e \rightsquigarrow a : \tau$. Par hypothèse d'induction, nous avons donc $e \approx [\![a]\!]$. La relation $\approx$ étant une congruence, ceci implique $\lambda x.e \approx \lambda x.[\![a]\!] = [\![\lambda x : \tau'.a]\!]$.

∘ *Cas* d'une application $e\, e'$. Le jugement considéré est $\Gamma \vdash e\, e' \rightsquigarrow a\, a' : \tau$, et découle de prémisses de la forme $\Gamma \vdash e \rightsquigarrow a : \tau' \to \tau$ et $\Gamma \vdash e' \rightsquigarrow a' : \tau'$. Par hypothèse d'induction, nous avons donc $e \approx [\![a]\!]$ et $e' \approx [\![a]\!]'$. La relation $\approx$ étant une congruence, ceci implique $e\, e' \approx [\![a]\!]\, [\![a']\!] = [\![a\, a']\!]$.

∘ *Cas* d'un liaison locale let $x = e$ in e'. Le jugement considéré est $\Gamma \vdash$ let $x = e$ in $e' \rightsquigarrow (\lambda x : \forall\bar{\alpha}.\tau.a')\, (\Lambda\bar{\alpha}.a) : \tau'$, et découle de prémisses de la forme $\Gamma \vdash e \rightsquigarrow a : \tau$ et $\Gamma; x : \forall\bar{\alpha}.\tau \vdash e' \rightsquigarrow a' : \tau'$. L'hypothèse d'induction donne donc $e = [\![a]\!]$ et $e' = [\![a']\!]$. Nous pouvons donc écrire let $x = e$ in $e' \approx (\lambda x.e')\, e \approx (\lambda x.[\![a']\!])\, [\![a]\!] = [\![(\lambda x : \forall\bar{\alpha}.\tau.a')\, (\Lambda\bar{\alpha}.a)]\!]$. □