TD2 février 2022

λ-calcul et logique informatique

leroux@lsv.fr

Exercice 1 — Un lemme de substitution
Soit $u, v, w \in \Lambda$ et x, y deux variables telles que $x, y \notin BV(u)$ et $FV(w) \cap BV(u) = FV(v) \cap BV(u) = \emptyset$ (tel que les substitutions ci-dessous soient valides).

1. Trouver u, v, w tel que $u[y := w][x := v] \neq u[x := v][y := w]$.
2. Trouver u, v, w tel que $u[y := w][x := v] \neq u[x := v][y := w[x := v]]$.
3. Montrer que si $y \notin FV(v)$, alors $u[y := w][x := v] = u[x := v][y := w[x := v]]$.

Exercice 2 — β-réduction et clôture reflexive et transitive
Ci-dessous, on donne la définition par récurrence de la clôture reflexive et transitive d'une relation binaire $\to$ quelconque.

$$\frac{}{u \to^* u}\ \text{refl} \qquad \frac{u \to v}{u \to^* v}\ \text{incl} \qquad \frac{u \to^* w \quad w \to^* v}{u \to^* v}\ \text{trans}$$

Dans cet exercice, la flèche simple $\to$ dénote dorénavant la β-réduction.

1. Montrer que pour tout $u, u' \in \Lambda$, si $u \to^* u'$ alors $\lambda x.u \to^* \lambda x.u'$.
2. Montrer que pour tout $u, u', v \in \Lambda$, si $u \to^* u'$ alors $uv \to^* u'v$.
3. Montrer que pour tout $u, v, v' \in \Lambda$, si $v \to^* v'$ alors $uv \to^* uv'$.

Exercice 3 — (Réduction parallèle)
Dans cet exercice, la flèche simple $\to$ dénote la β-réduction. On rappelle les régles de la β-réduction :

$$\frac{}{(\lambda x.u)v \to u[x := v]}\ \text{beta} \qquad \frac{u \to u'}{\lambda x.u \to \lambda x.u'}\ \text{abs}$$
$$\frac{u \to u'}{uv \to u'v}\ \text{appG} \qquad \frac{v \to v'}{uv \to uv'}\ \text{appD}$$

On définit la réduction parallèle comme suit, où y représent une variable abritraire :

$$\frac{u \Rightarrow u' \quad v \Rightarrow v'}{uv \Rightarrow u'v'}\ \text{appP} \qquad \frac{u \Rightarrow u'}{\lambda x.u \Rightarrow \lambda x.u'}\ \text{absP}$$
$$\frac{}{y \Rightarrow y}\ \text{reflVarP} \qquad \frac{u \Rightarrow u' \quad v \Rightarrow v'}{(\lambda x.u)v \Rightarrow u'[x := v']}\ \text{betaP}$$

1. Montrer que la règle supplémentaire suivante peut se déduire des règles originales.
$$\frac{}{u \Rightarrow u}\ \text{reflP}$$
2. Montrer que la règle supplémentaire suivante peut se déduire des règles originales.
$$\frac{u \Rightarrow u' \quad v \Rightarrow v'}{u[x := v] \Rightarrow u'[x := v']}\ \text{subP}$$
3. Montrer les inclusions suivantes en donnant des exemples illustrant qu'elles sont strictes :
   (a) $\to \subseteq \Rightarrow$
   (b) $\Rightarrow \subseteq \to^*$
4. Montrer que $\Rightarrow$ est fortement confluente.
5. En déduire que $\to$ est confluente.

Solutions exercice 1

1. Soient $u := y$, $w := v := xx$. Alors $u[y := w][x := v] = xx(xx) \neq xx = u[x := v][y := w]$.
2. Soient $u := x$, $v := y$, $w = z$. Alors $u[y := w][x := v] = y \neq z = u[x := v][y := w[x := v]]$.
3. On montre la propriété par récurrence structurelle sur u.
   — Cas (de base) où u et une variable. On procède par disjonction de cas.
      — Si $u = x$, alors $u[y := w][x := v] = x[x := v] = v$ et $u[x := v][y := w[x := v]] = v[y := w[x := v]] = v$, car $y \notin FV(v)$ par hypothèse.
      — Si $u = y$, alors $u[y := w][x := v] = w[x := v]$ et $u[x := v][y := w[x := v]] = y[y := w[x := v]] = w[x := v]$.
      — Si $u \notin \{x, y\}$, alors l'égalité est vraie car $u = u$.

   — Cas où $u = AB$. Alors

$$
\begin{aligned}
u[y := w][x := v] &= (AB)[y := w][x := v] \\
&= (A[y := w][x := v])(B[y := w][x := v]) \\
&\quad \text{par définition de la substitution} \\
&= (A[x := v][y := w[x := v]])(B[x := v][y := w[x := v]]) \\
&\quad \text{par hypothèse de récurrence (HR)} \\
&\quad \text{car les conditions sur les variables sont conservées.} \\
&= (AB)[x := v][y := w[x := v]] \\
&\quad \text{par définition de la substitution} \\
&= u[x := v][y := w[x := v]]
\end{aligned}
$$

   — Cas où $u = \lambda z.A$, avec z variable fraîche. Alors

$$
\begin{aligned}
u[y := w][x := v] &= (\lambda z.A)[y := w][x := v] \\
&= (\lambda z.A[y := w])[x := v] \text{ car } z \text{ est fraîche} \\
&= \lambda z.A[y := w][x := v] \text{ car } z \text{ est fraîche} \\
&= \lambda z.A[x := v][y := w[x := v]] \text{ par HR} \\
&= (\lambda z.A)[x := v][y := w[x := v]] \text{ car } z \text{ est fraîche} \\
&= u[x := v][y := w[x := v]]
\end{aligned}
$$

Solutions exercice 2

1. Montrons que $u \to^* u'$ implique $\lambda x.u \to^* \lambda x.u'$ par récurrence structurelle sur $u \to^* u'$.
   — Cas (de base) où $u \to^* u'$ vient de $u' = u$. Alors $\lambda x.u \to^* \lambda x.u'$, car $\lambda x.u = \lambda x.u'$.
   — Cas (de base) où $u \to^* u'$ vient de $u' \to u$. Alors $\lambda x.u \to^* \lambda x.u'$, car $\lambda x.u \to \lambda x.u'$ par la règle abs, puis par la règle incl.
   — Cas où $u \to^* u'$ vient de $u \to^* v$ et $v \to^* u'$. D'une part $\lambda x.u \to^* \lambda x.v$ par HR, et d'autre part $\lambda x.v \to^* \lambda x.u'$ également par HR. Donc $\lambda x.u \to^* \lambda x.u'$ par la règle trans (i.e. transitivité de $\to^*$).

2. Montrons que $u \to^* u'$ implique $uv \to^* u'v$ par récurrence sur $u \to^* u'$.
   — Cas (de base) où $u \to^* u'$ vient de $u' = u$. Alors $uv \to^* u'v$, car $uv = u'v$.
   — Cas (de base) où $u \to^* u'$ vient de $u' \to u$. Alors $uv \to^* u'v$, car $uv \to u'v$ par la règle appG.
   — Cas où $u \to^* u'$ vient de $u \to^* w$ et $w \to^* u'$. D'une part $uv \to^* wv$ par HR, et d'autre part $wv \to^* u'v$ également par HR. Donc $uv \to^* u'v$ par transitivité de $\to^*$.

3. D'une manière similaire on peut montrer que $v \to^* v'$ implique $uv \to^* uv'$.

Solutions exercice 3

1. À faire.
2. Montrons que pour tout $u, v, u', v' \in \Lambda$, si $u \Rightarrow u'$ et $v \Rightarrow v'$ alors $u[x := v] \Rightarrow u'[x := v']$, par récurrence structurelle sur u.
   - Cas (de base) où u est une variable. Alors $u' = u$, car $u \Rightarrow u'$ ne peut venir que de la règle refl. Si $u \neq x$ alors $u[x := v] = u'[x := v']$, et $u[x := v] \Rightarrow u'[x := v']$ par reflP. Si $u = x$, alors $u[x := v] = v$ et $u'[x := v'] = v'$, et $u[x := v] \Rightarrow u'[x := v']$ par hypothèse $v \Rightarrow v'$.
   - Cas où $u = \lambda y.A$ (où y est une variable fraîche, en particulier n'apparaissant pas dans v ou v'). Alors $u \Rightarrow u'$ ne peut venir que de la règle absP. Donc $u' = \lambda y.A'$ avec $A \Rightarrow A'$. Par hypothèse de récurrence (HR), on a $A[x := v] \Rightarrow A'[x := v']$, donc $\lambda y.A[x := v] \Rightarrow \lambda y.A'[x := v']$ par la règle absP. Or $\lambda y.A[x := v] = (\lambda y.A)[x := v]$ et par définition de la substitution, et de même $\lambda y.A'[x := v'] = (\lambda y.A')[x := v']$, donc $u[x := v] \Rightarrow u'[x := v']$.
   - Cas où $u = AB$. On procède par disjonction de cas.
     - Cas où $u \Rightarrow u'$ vient de appP. Alors $u = AB$ et $u' = A'B'$ avec $A \Rightarrow A'$ et $B \Rightarrow B'$. Par HR on a $A[x := v] \Rightarrow A'[x := v']$ et $B[x := v] \Rightarrow B'[x := v']$. Par la règle appP on trouve $A[x := v]B[x := v] \Rightarrow A'[x := v']B'[x := v']$. Or par définition de la substitution on a $A[x := v]B[x := v] = (AB)[x := v]$, donc $u[x := v] \Rightarrow u'[x := v']$.
     - Cas où $u \Rightarrow u'$ vient de betaP. Alors $u := (\lambda y.A)B$ et $u' := A'[y := B']$ avec $A \Rightarrow A'$ et $B \Rightarrow B'$. Alors

$$
\begin{aligned}
u[x := v] &= ((\lambda y.A)B)[x := v] \\
&= (\lambda y.A)[x := v]B[x := v] \\
&= (\lambda y.A[x := v])B[x := v] \\
&\Rightarrow (A'[x := v'])[y := B'[x := v']] \\
&\quad \text{car } A[x := v] \Rightarrow A'[x := v'] \text{ et } B[x := v] \Rightarrow B'[x := v'] \\
&\quad \text{par HR, et par la règle betaP} \\
&= A'[y := B'][x := v'] \\
&\quad \text{par l'exercice 1, en particulier car } y \text{ est fraîche} \\
&= u'[x := v']
\end{aligned}
$$

3. (a) Soit $M := (\lambda x.x)y$. On a $yMM \Rightarrow yyy$, mais $yMM \not\to yyy$. Cela montre que s'il y a inclusion, celle-ci est stricte.

Montrons que pour tout $u, v \in \Lambda$ si $u \to v$ alors $u \Rightarrow v$, par récurrence structurelle sur $u \to v$.

— Cas (de base) où $u \to v$ vient de la règle beta, i.e $u = (\lambda x.A)B$ et $v = A[x := B]$. Par reflP, on a $A \Rightarrow A$ et $B \Rightarrow B$, donc par betaP on trouve $u \Rightarrow v$.
— Cas où $u \to v$ vient de la règle abs, i.e. $u = \lambda x.A$ et $v = \lambda x.A'$ avec $A \to A'$. Par hypothèse de récurrence (HR) on a $A \Rightarrow A'$, donc par la règle absP, on a $u \Rightarrow v$.
— Cas où $u \to v$ vient de la règle appG, i.e. $u = AB$ et $v = A'B$ avec $A \to A'$. Par reflP, on a $B \Rightarrow B$, et par HR on a $A \Rightarrow A'$, donc par appP on a $u \Rightarrow v$.
— Le cas où $u \to v$ vient de la règle appD est traité de manière similaire.

(b) $(\lambda xy.z)xy \to^2 z$ mais $(\lambda xy.z)xy \not\Rightarrow z$. Cela montre que s'il y a inclusion, celle-ci est stricte.

Montrons que pour tout $u, v \in \Lambda$ si $u \Rightarrow v$ alors $u \to^* v$, par récurrence structurelle sur $u \Rightarrow v$.

— Cas (de base) où $u \Rightarrow v$ vient de reflVarP, i.e. $v = u$ est une variable y. Alors $u \to^* v$ par reflexivité de la clôture réflexive et transitive.
— Cas où $u \Rightarrow v$ vient de absP, i.e. $u = \lambda x.A$ et $v = \lambda.A'$ et $A \Rightarrow A'$. Par HR on a $A \to^* A'$. Par l'exercice 2.1 on a $\lambda x.A \to^* \lambda x.A'$.
— Cas où $u \Rightarrow v$ vient de appP, i.e. $u = AB$ et $v = A'B'$ avec $A \Rightarrow A'$ et $B \Rightarrow B'$. Par HR on a $A \to^* A'$ et $B \to^* B'$. Par l'exercice 2.2 on a $AB \to^* A'B$. Par l'exercice 2.3 on a $A'B \to^* A'B'$. Ainsi, $AB \to^* A'B'$ par la transitivité de $\to^*$.
— Cas où $u \Rightarrow v$ vient de betaP, i.e. $u = (\lambda x.A)B$ et $v = A'[x := B']$, avec $A \Rightarrow A'$ et $B \Rightarrow B'$. Par HR, on a $A \to^* A'$ et $B \to^* B'$. Par l'exercice 2.1 on a donc $\lambda x.A \to^* \lambda x.A'$. Par l'exercice 2.2 et 1.3 on a $(\lambda x.A)B \to^* (\lambda x.A')B'$. Par la règle beta on a $(\lambda x.A')B' \to A'[x := B']$, donc $(\lambda x.A)B \to^* A'[x := B']$ par définition de la clôture transitive.

4. Montrons que pour tout $u, u_1, u_2 \in \Lambda$, si $u \Rightarrow u_1$ et $u \Rightarrow u_2$ alors il existe $v \in \Lambda$ tel que $u_1 \Rightarrow v$ et $u_2 \Rightarrow v$, par récurrence structurelle sur u.

— Cas (de base) où u est une variable : donc $u_1 = u_2 = u$, et $v := u$

convient par la règle reflP.
- Cas où u est une λ-abstraction $\lambda x.A$. Alors u_i est nécessairement de la forme $\lambda x.A_i$ avec $A \Rightarrow A_i$, pour tout $i \in \{1, 2\}$, car seule la règle absP a pu s'appliquer pour réduire u. Par HR, il existe B tel que $A_1 \Rightarrow B$ et $A_2 \Rightarrow B$. Soit $v := \lambda x.B$. Par la règle absP, on a donc $u_1 \Rightarrow v$ et $u_2 \Rightarrow v$, donc v convient.
- Cas où u est une application AB. Procédons par disjonction de cas.
  - Si $u_1 = u$ (resp. $u_2 = u$) alors $v := u_2$ (resp. $v := u_1$) convient. Cela couvre les cas où au moins l'une des réductions parallèles vient de reflP.
  - Si $u_1 = A_1B_1$ et $u_2 = A_2B_2$ et les deux réductions viennent de appP. Alors $A \Rightarrow A_i$ et $B \Rightarrow B_i$ pour tout $i \in \{1, 2\}$. Par HR soient A' et B' tels que $A_i \Rightarrow A'$ et $B_i \Rightarrow B'$ pour tout $i \in \{1, 2\}$. Pour tout $i \in \{1, 2\}$, par appP on a $A_iB_i \Rightarrow A'B'$, donc $v := A'B'$ convient.
  - Si une des deux réductions parallèles (par exemple la première) vient de betaP et l'autre de appP, alors il existe $C, B, C_1, B_1, C_2, B_2 \in \Lambda$ tels que $A = \lambda x.C$ et $u_1 = C_1[x := B_1]$ et $u_2 = (\lambda x.C_2)B_2$, avec $C \Rightarrow C_i$ et $B \Rightarrow B_i$ pour tout $i \in \{1, 2\}$. Par HR, soient C' et B' tel que $C_i \Rightarrow C'$ et $B_i \Rightarrow B'$ pour tout $i \in \{1, 2\}$. Soit $v := C'[x := B']$. Par la question 3.1 on a $C_1[x := B_1] \Rightarrow C'[x := B']$ et par la règle betaP on a $(\lambda x.C_2)B_2 \Rightarrow C'[x := B']$, donc v convient.
  - Si les deux réductions parallèles viennent de betaP, alors on invoque deux fois la question 3.1.

5. On note que par un résultat du cours, toute relation fortement confluente, telle $\Rightarrow$, est confluente. On note que $\to^* \subseteq \Rightarrow^*$ car $\to \subseteq \Rightarrow$ comme prouvé à la question 3.2.a. On note finalement que

   Soient $u, u_1, u_2 \in \Lambda$ tels que $u \to^* u_1$ et $u \to^* u_2$. Alors $u \to^* u_1$ et $u \to^* u_2$