Université Paris Diderot - Master 1 Linguistique Informatique - Sémantique computationnelle - 2013/2014

TD 6 - Lambda calcul

1 Lambda calcul non typé

1. **Convention de notation** Donnez les formes entièrement parenthésées des termes suivants. Si ce dont des rédex, les réduire autant que possible.

(1) a. $(\lambda x.\lambda y.fxy)xy$
b. $(\lambda x.\lambda y.xyy)\lambda y.\lambda a.y$

2. **Syntaxe des lambda-termes** Donnez l'arbre de décomposition syntaxique de :

(2) a. $\lambda f.\lambda g.\lambda x.(f)(g)x$
b. $\lambda f.\lambda g.\lambda x.((f)g)x$
c. aab, à comparer avec $(a)(a)b$
d. $(M)ab$, à comparer avec Mab
e. $(\lambda f.(f)g)x$

On rappelle que les étiquettes des noeuds d'un tel arbre peuvent être : LA (lambda abstraction), AF (application fonctionnelle) ou VAR (variable).
Quelle est la forme de l'arbre de décomposition d'une expression béta-réductible ?

3. β**-réduction :** Réduire autant que possible les termes suivants

(3) a. $(\lambda x.xx)\lambda x.x$
b. $((\lambda x.\lambda y.yx)f)\lambda x.x$
c. $(\lambda n.\lambda f.\lambda x.(f)((n)f)x)\lambda fx.fx$

4. β**-réduction :** Réduire autant que possible les expressions suivantes :

(4) a. $(\lambda x.\ (P)x)m$
b. $((\lambda x.\ \lambda Y.\ (Y)x)j)P$
c. $(\lambda x.\forall z.((\lambda y.((K)x)y)z \to ((R)z)x))j$

En fixant une interprétation des prédicats K, R et P, et des constantes m et j, proposez pour chacune de ces formules une phrase en français ayant les mêmes conditions de vérité.

5. β**-réduction :** Réduire autant que possible l'expression suivante en faisant attention à la portée des lambda abstracteurs [1].

(5) $((\lambda x.\lambda y.x)((\lambda x.\lambda y.y)\lambda x.\lambda y.x)\lambda x.\lambda y.y)\lambda x.\lambda y.y$

6. β**-réduction :** Identifiez tous les rédex de la forme suivante, et réduisez-la autant que possible. Si de nouveaux rédex apparaissent identifiez-les.

(6) $((\lambda S.\lambda V.(S)(V)\lambda Q.(Q)m)\lambda P.(P)j)\lambda O.\lambda y.(O)\lambda z.((kiss)y)z$

1. Indication : vous devez obtenir finalement une expression de la forme $\lambda a.\lambda b.b$

7. **Combinateurs booléens**

On représente les deux valeurs booléennes « vrai » et « faux » par les fonctions à deux arguments suivantes : $\mathsf{T} =_{\text{def}} \lambda x.\lambda y.x$
$\mathsf{F} =_{\text{def}} \lambda x.\lambda y.y$

Ces définitions, conventionnelles, s'expliquent par la forme très simple que reçoit alors la définition par cas :

`if` P `then` Q `else` $R =_{\text{def}} ((P)Q)R$.

En effet, si P se réduit en T (ie $P \equiv \mathsf{T}$), alors `if` P `then` Q `else` $R \equiv \mathsf{T}QR \equiv Q$. De même, si $P \equiv \mathsf{F}$, alors on obtient R.

(7) a. Utiliser cette définition par cas pour définir les opérateurs $\neg$, $\wedge$, $\vee$. Vous pouvez éventuellement vous reporter à la section **Aide**.

b. En utilisant les définitions de T et F calculer la valeur de vérité de :

1. $\neg P$ quand P est faux puis quand P est vrai.
2. $P \wedge Q$ quand P et Q sont vrais.
3. $P \vee Q$ quand P et Q sont faux puis quand P est vrai et Q est faux.
4. A quelle formule mettant en jeu un opérateur binaire liant deux propositions P et Q et quelles valeurs pour P et Q correspond l'expression de la question 5 ?

8. **Codage (de Church) des entiers**

On définit : $0 =_{\text{def}} \lambda f.\lambda x.x$
$1 =_{\text{def}} \lambda f.\lambda x.(f)x$
$n =_{\text{def}} \lambda f.\lambda x.\underbrace{(f)(f)...(f)}_{nfois}x$

On peut ainsi représenter tous les entiers.

(8) a. Définir l'opérateur successeur tel que $Succ(n) = n+1$. Chercher cette fonction en essayant de passer de 1 à 2. Si vous avez des difficultés, reportez-vous à la section **Aide**.
Vérifier en cherchant le successeur de 2.

b. Définir l'opérateur d'addition. Cette opération correspond à une sorte de « concaténation » des deux lambda-termes. Vous pouvez encore vous reporter à la section **Aide**.
Vérifier en effectuant l'opération $1 + 2$.

c. Vérifier que $\lambda n.\lambda m.\lambda f.(n)(m)f$ est l'opérateur de multiplication en effectuant l'opération $2 * 3$.

d. Question bonus : définir le prédicat `iszero` qui renvoie T si son argument est l'entier 0.

Aide

- **Question 7** : vous trouverez ci-dessous une aide au raisonnement, essayez de trouver les formules grâce à ce raisonnement. Ces formules sont une combinaison d'une ou plusieurs lambda abstraction (des λx) et des valeurs T et F. Si vous ne parvenez pas à trouver les formules à l'aide de ces raisonnements, le résultat est donné en bas de page, essayez de comprendre comment on y est parvenu à partir du raisonnement.

  On peut définir (relativement) aisément les combinateurs booléens en passant par la définition par cas :

  - Pour NOT : $\neg P$ = `if` P `then` F `else` T
    Le lambda terme recherché est une combinaison des valeurs booléennes T et F telle que si $P = T$, on récupère F, sinon on récupère T [2].
  - Pour AND : $P \wedge Q$ = `if` P `then` (`if` Q `then` T `else` F) `else` F [3]
  - Pour OR : $P \vee Q$ = `if` P `then` T `else` (`if` Q `then` T `else` F) [4]

- **Question 8** : vous trouverez ci-dessous des formules à trou, les X peuvent correspondre à des variables $(n, m...)$ et/ou à des parenthèses.

  - On cherche une certain fonction $Succ$ telle que :
    $(Succ)\lambda f.\lambda x.(f)x \equiv \lambda f.\lambda x.(f)(f)x$

    Partez de : $Succ = \lambda n.\lambda f.\lambda x.(f)\underbrace{X...X}_{\text{à trouver}}$
  - On cherche une fonction Add telle que :
    $((Add)\lambda f.\lambda x.(f)x)\lambda f.\lambda x.(f)(f)x \equiv \lambda f.\lambda x.(f)(f)(f)x$

    Partez de : $Add = \lambda n.\lambda m.\lambda f.\lambda x.((n)X)((m)X)X$.

2. $(NOT)P \equiv ((P)F)T)$, alors si $P \equiv$ T, alors NOT$P \equiv ((P)$F)T $\equiv$ ((T)F)T $\equiv$ F. D'où par λ-abstraction : NOT $=_{\text{def}} \lambda x.((x)$F)T.
3. $((AND)P)Q \equiv ((P)((Q)T)F)F$ d'où : AND $=_{\text{def}} \lambda x.\lambda y((x)((y)T)F)F$
4. $((OR)P)Q \equiv ((P)T)((Q)T)F$, et je vous laisse trouver la formule finale.

2 Typage

1. Si j est une constante de type e, M une constante de type $\langle e, t\rangle$, quel est le type des variables x, et Y pour que les expressions suivantes soient bien formées ? Réduire.

   (9) a. $(\lambda x.(M)x)j$
   b. $(\lambda Y.(Y)j)M$

2. Si l'on veut que la dénotation d'un nom propre soit de type e : par exemple, $[\![\text{léa}]\!] = l$, on peut charger la règle $SN \rightarrow NP$ pour garantir la « montée de type », de sorte que $[\![[_{SN}\text{léa}]]\!] = \lambda P.(P)l$. Il faut donc trouver un λ-terme (un combinateur) Ψ tel que $[\![\text{SN}]\!] = (\Psi)[\![\text{NP}]\!]$. Comment s'écrit Ψ ?

3. Comment représenter la contribution du verbe *être* dans la phrase « Jean est mortel » ?

4. On s'accorde généralement pour considérer que la contribution d'un adjectif épithète (intersectif) est un prédicat ($\langle e, t\rangle$). Si on considère que les adjectifs sont adjoints au N', comment définir les règles de composition pour que le N' le plus élevé soit bien du type $\langle e, t\rangle$?

5. La coordination en *et* peut s'appliquer à de multiples niveaux en français. Proposer une représentation de *et* qui fonctionne dans les cas suivants.

   (10) a. Paul est paresseux et menteur
   b. Jean et Marie dorment
   c. Paul regarde et admire Marie