

M2 LMFI – Cours Fondamental de logique

THÉORIE DE LA DÉMONSTRATION

Premières notions de lambda-calcul

(version préliminaire)

Alexis Saurin

Cours des 5 et 13 novembre 2021

Contents

1	Premières définitions	1
2	α , β et substitution	3
3	Graphes des réductions	8
4	Exercices du chapitre	11

1 Premières définitions

Le λ -calcul est un langage de fonctions: l'abstraction de la variable x dans un terme t , $\lambda x.t$ correspond à la notation usuelle en mathématiques $x \mapsto t$, la notation d'application d'un terme t à un argument u , $(t)u$, correspond à la notation mathématique usuelle $t(u)$.

L'interprétation fonctionnelle informelle des termes ci-dessus est la suivante: $\lambda x.x$ correspond à la fonction dont le résultat est l'argument qu'elle a reçu, c'est-à-dire la fonction identité, tout comme le second terme $\lambda y.y$: le nom de la variable n'importe pas. $\lambda x.y$, avec $x \neq y$ est la fonction qui, à un argument x , associe y : il s'agit d'une fonction constante retournant y , de même que le terme $\lambda z.y$.

$\lambda x.(x)x$ désigne la fonction prend un argument et l'applique à lui-même. $(x)\lambda x.\lambda x.x$ est le résultat de l'application d'une fonction x à la fonction qui prend deux arguments et retourne le second. $\lambda x.(y)z$ est une fonction constante qui retourne toujours $(y)z$. $\lambda x.(x)(x)y$ est la fonction qui, étant donné un argument x , itère deux fois l'application de cette fonction à l'argument y . $\lambda x.\lambda y.\lambda z.((z)x)y$ représente la fonction qui prenant trois arguments, applique son troisième argument à son premier puis applique cela à son deuxième argument. $\lambda x.\lambda y.x$ est la fonction qui prend deux arguments et retourne le premier; une autre manière de voir les choses est de l'interpréter comme la fonction qui prend un argument et construit la fonction constante qui retourne cet argument. $\lambda x.\lambda y.y$ désigne la fonction qui prend un argument et retourne la fonction identité, ou encore la fonction qui prend un premier argument, puis un second argument et retourne ce dernier. $(\lambda x.(x)x)\lambda y.(y)y$ représente le résultat de l'application de la fonction qui prend un argument et l'applique à lui-même, appliquée à elle-même...

Définition 1.1 (*Syntaxe du lambda-calcul*)

On se donne un ensemble dénombrable de variables $\mathcal{V}(\ni x, y, z, x', y', z', \dots)$.

L'ensemble Λ des λ -termes est un ensemble de mots construits sur l'alphabet $\mathcal{A}_\Lambda = \mathcal{V} \cup$

$\{\lambda, ., (,)\}$ et est défini inductivement par la grammaire suivante:

$$\Lambda \ni t, u, v, t', u', v' ::= x \mid (t)u \mid \lambda x.t$$

En d'autres termes, Λ est le plus petit ensemble contenu dans $\mathcal{A}_\Lambda^*$ tel que:

- $\mathcal{V} \subset \Lambda$;
- si $t \in \Lambda$ et $x \in \mathcal{V}$, alors $\lambda x.t \in \Lambda$;
- si $t, u \in \Lambda$, $(t)u \in \Lambda$.

Notation 1.2

Lorsqu'on voudra indiquer que deux λ -termes sont syntaxiquement égaux, on utilisera la notation $t \equiv u$ (mais dans la pratique, il arrivera souvent, lorsqu'il n'y a pas d'ambiguïté, que l'on utilise simplement le symbole usuel d'égalité).

Remarque 1.3

Il s'agit d'une grammaire non ambiguë dans le sens où elle satisfait une propriété de lecture unique, ce qui revient à dire que l'ensemble Λ est isomorphe à un ensemble d'arbres comme nous le verrons ci-dessous.

On trouve dans la littérature de petites variantes dans la syntaxe des λ -termes. Il s'agit principalement des variantes superficielles typiquement $(t u)$ pour l'application, mais également des différences plus profondes, comme la notation de "de Bruijn" que nous évoquerons plus bas.

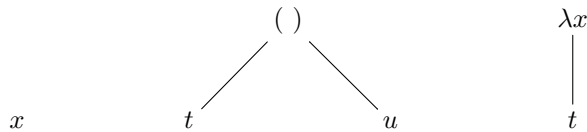
Notation 1.4

Pour simplifier l'écriture des λ -termes et minimiser le nombre de parenthèses, on adoptera les conventions de notation suivantes:

- L'application associe à gauche et on utilisera cette convention pour omettre certaines parenthèses superflues: on écrira $(t)uv$ pour $((t)u)v$ et plus généralement $(u)t_1 \dots t_n$ pour $((u)t_1) \dots t_n$. Autrement dit, $(t)(u)vw$ désigne $(t)((u)v)w$ et non pas $((t)(u)v)w$.
- La portée d'une λ -abstraction est la plus grande possible: elle ne s'arrête que lorsqu'une parenthèse qui clôt la λ -abstraction est rencontrée. Autrement dit, $(t)\lambda x.(u)vw$ désigne $(t)\lambda x.((u)v)w$ et non pas $((t)\lambda x.(u)v)w$ (et encore moins $(t)(\lambda x.(u)v)w \dots$);
- Enfin, pour condenser les notations, on va également s'autoriser deux simplifications de la notation pour les abstractions:
 - on notera une séquence d'abstractions sous une abstraction unique en regroupant les différentes variables abstraites entre le λ et le $.$: $\lambda x.\lambda y.\lambda z.t$ sera ainsi noté: $\lambda xyz.t$;
 - il arrivera que l'on note $\lambda \bar{x}.t$ (resp. $(t)\bar{u}$) pour $\lambda x_1 \dots \lambda x_n.t$ (resp. $((t)u_1) \dots u_n$) où $\bar{x}$ (resp. $\bar{u}$) désigne la séquence $x_1 \dots x_n$ (resp. $u_1 \dots u_n$).

Définition 1.5 (Notation d'arbre pour les lambda-termes)

La syntaxe des λ -termes définit des arbres. Il peut être utile de représenter les λ -termes sous cette forme arborescente. On adoptera dans ce cas les conditions suivantes pour représenter respectivement les variables (les feuilles des arbres), les applications (des noeuds internes binaires) et les λ -abstractions (des noeuds internes unaires):



Exemple 1.6

Voici quelques exemples de λ -termes:

$$\begin{array}{ll} \lambda x.x; & \lambda y.y; \\ \lambda x.y; & \lambda z.y \\ \lambda x.(x)x; & (x)\lambda x.\lambda x.x; \\ \lambda x.(y)z; & \lambda x.(x)(x)y; \\ \lambda x.\lambda y.\lambda z.((z)x)y; & \lambda x.\lambda y.x; \\ \lambda x.\lambda y.y; & (\lambda x.(x)x)\lambda y.(y)y. \end{array}$$

Exercice 1.1

Appliquer aux exemples ci-dessus les simplifications de parenthésage et de notation introduites plus haut et les représenter sous forme arborescente.

Définition 1.7 (Λ^0 , Variables libres et liées)

On désignera par $FV(t)$ et $BV(t)$ les ensembles (d'occurrences) de variables respectivement **libres** et **liées** dans le terme t . Ces ensembles sont définis de manière inductive comme suit:

$$\begin{array}{ll} FV(x) = \{x\} & BV(x) = \emptyset \\ FV((t)u) = FV(t) \cup FV(u) & BV((t)u) = BV(t) \cup BV(u) \\ FV(\lambda x.t) = FV(t) \setminus \{x\} & BV(\lambda x.t) = BV(t) \cup \{x\} \end{array}$$

Λ^0 désigne l'ensemble des λ -termes **clos**, ie. n'ayant pas de variable libre:

$$\Lambda^0 = \{t \in \Lambda, FV(t) = \emptyset\}.$$

Enfin, on appellera **clôture** d'un terme t tout terme $\lambda x_1 \dots \lambda x_n.t$ tel que $FV(t) = \{x_1, \dots, x_n\}$.

Définition 1.8 (Convention de Barendregt)

On dira qu'un terme satisfait la **convention de Barendregt** s'il est tel que:

- ses ensembles de variables libres et liées sont disjoints;
- tous les λ sont étiquetés par des variables distinctes.

Dans la suite, on va s'intéresser à des relations binaires sur les λ -termes. Deux catégories de relations vont particulièrement nous être utiles, les relations compatibles et les congruences:

Définition 1.9 (Relation compatible, congruence)

Une relation binaire $\mathcal{R}$ sur Λ est dite **compatible** si:

$$\forall t, u, v \in \Lambda, \forall x \in \mathcal{V}, \quad t \mathcal{R} u \Rightarrow (t)v \mathcal{R} (u)v, \quad (v)t \mathcal{R} (v)u, \quad \lambda x.t \mathcal{R} \lambda x.u$$

Une **congruence** est une relation d'équivalence compatible.

2 α , β et substitution

Étant données les interprétations informelles de l'abstraction et de l'application, un λ -terme de la forme $(\lambda x.t)u$ correspond à l'application de la fonction qui à x associe t à l'argument u . En d'autres termes, $(\lambda x.t)u$ devrait être égal au terme t dans lequel les occurrences libres de x ont été remplacées par u . Pour cela, on va introduire une notion de *substitution d'une variable x par un terme u dans le terme t* qui sera noté $t\{u/x\}$.

Avant de définir cette substitution, il est utile de remarquer le fait suivant: puisque les abstractions servent à construire des fonctions, la variable x abstraite dans $\lambda x.t$ doit être une *variable muette*, c'est-à-dire qu'en la remplaçant par une autre variable y (à condition que $y \notin FV(t)$)

et en remplaçant toutes les autres occurrences libres de x dans t par y , alors on obtient un terme qui est équivalent au premier. Ceci va être formalisé par l' α -équivalence. Une fois cette relation d'équivalence définie, on travaillera sur l'ensemble quotient $\Lambda/\equiv_\alpha$ des λ -termes modulo α -équivalence.

La définition de la substitution des occurrences libres de la variable libre x dans le terme t par le terme u va donc être définie en plusieurs étapes pour parvenir à une définition précise. On commence par deux définitions incorrectes pour motiver la définition finale. La seconde de ces définitions se révélera pourtant utile. On définira l' α -équivalence puis on reconsidérera les objets sur lesquels on travaillera. Finalement, on donnera la définition finale de la substitution qui nous servira dans la suite du cours.

Définition 2.1 (*Première tentative de substitution par u des occurrences de x dans t , $t(u/x)$*)

La substitution par u des occurrences libres de x dans t , notée $t(u/x)$, est définie par induction sur la structure de t de la manière suivante:

- $t \equiv x$. $t(u/x) = u$;
- $t \equiv y$ avec $y \neq x$. $t(u/x) = y$;
- $t \equiv (v)w$. $t(u/x) = (v')w'$ avec $v' = v(u/x)$ et $w' = w(u/x)$;
- $t \equiv \lambda y.v$. $t(u/x) = \lambda y.v'$ avec $v' = v(u/x)$;

Cette définition n'est pas satisfaisante. Ainsi, alors que le terme $(\lambda x.\lambda x.x)t$ est l'application de la fonction constante retournant l'identité à un argument t , $\lambda x.x(t/x)$ devrait donc être la fonction constante mais est en fait égal à $\lambda x.t$.

De même, alors que le terme $(\lambda y.\lambda x.y)x$ est l'application de la fonction qui construit une fonction constante à un argument x , $\lambda x.y(x/y)$ devrait donc être la fonction constante retournant x mais est en fait égal à $\lambda x.x$ qui est la fonction identité.

Définition 2.2 (*Deuxième tentative de substitution par u des occurrences de x dans t , $t[u/x]$*)

La substitution par u des occurrences libres de x dans t , notée $t[u/x]$, est définie par induction sur la structure de t de la manière suivante:

- $t \equiv x$. $t[u/x] = u$;
- $t \equiv y$ avec $y \neq x$. $t[u/x] = y$;
- $t \equiv (v)w$. $t[u/x] = (v')w'$ avec $v' = v[u/x]$ et $w' = w[u/x]$;
- $t \equiv \lambda x.v$. $t[u/x] = t$;
- $t \equiv \lambda y.v$. $t[u/x] = \lambda y.v'$ si $y \neq x$ avec $v' = v[u/x]$;

Cette notion de substitution n'est toujours pas satisfaisante: si elle évite le premier problème, elle présente toujours le second.

Pour autant, cette notion de substitution, *avec capture de variable*, s'avère utile pour définir l' α -équivalence:

La définition suivante va nous permettre de régler cette question:

Définition 2.3 (*Renommage de variable et α -équivalence*)

On appelle α la relation sur Λ^2 définie comme:

$$\alpha = \{(\lambda x.t, \lambda y.t[y/x]); t \in \Lambda, x \in \mathcal{V}, y \in \mathcal{V} \setminus (FV(t) \cup BV(t))\}.$$

L' α -équivalence, notée $\equiv_\alpha$, est définie comme la plus petite congruence contenant la relation α .

Remarque 2.4

On pourrait se dire que la définition précédente n'est pas plus satisfaisante que la seconde

définition de la substitution puisqu'elle utilise la substitution! Pourtant, à bien y regarder, la condition imposée à y de n'apparaître ni libre ni liée dans t nous assure que la clause problématique de la définition 2.2 qui entraîne la capture de variable ne se pose jamais.

Proposition 2.5 (Caractérisation alternative de l' α -équivalence)

La relation $=_\alpha$ définie ci-dessus coïncide avec la relation définie par $u \approx_\alpha v$ si:

- u est une variable et $u \equiv v$;
- $u = (u_1)u_2$ et $v = (v_1)v_2$ tel que $u_i \approx_\alpha v_i, i \in \{1, 2\}$;
- $u \equiv \lambda x.u'$ et $v \equiv \lambda y.v'$ tels que $u'[z/x] \approx_\alpha v'[z/y]$ pour toute variable z sauf pour un nombre fini de variables.

La congruence engendrée par α nous permet d'identifier deux λ -termes qui ne diffèrent que par le nom des variables liées. À partir de maintenant, on travaillera toujours à α -équivalence près, c'est-à-dire qu'on travaillera en fait sur l'ensemble $\Lambda / =_\alpha$ que l'on appellera simplement Λ par concision et l'égalité syntaxique $\equiv$ désignera à partir de maintenant l' α -équivalence.

Convention 2.6

À partir de maintenant, Λ désignera $\Lambda / =_\alpha$ et $\equiv$ désignera $=_\alpha$.

On dispose maintenant des outils nécessaires pour définir la bonne notion de substitution:

Définition 2.7 (Substitution par u des occurrences libres de x dans t , $t\{u/x\}$)

La substitution par u des occurrences libres de x dans t , notée $t\{u/x\}$, est définie par induction sur la structure de t de la manière suivante:

- $t \equiv x$. $t\{u/x\} = u$;
- $t \equiv y$ avec $y \neq x$. $t\{u/x\} = y$;
- $t \equiv (v)w$. $t\{u/x\} = (v')w'$ avec $v' = v\{u/x\}$ et $w' = w\{u/x\}$;
- $t \equiv \lambda x.v$. $t\{u/x\} = t$;
- $t \equiv \lambda y.v$, avec $y \neq x$ et avec $y \notin FV(u)$ ou $x \notin FV(v)$. $t\{u/x\} = \lambda y.v'$ avec $v' = v\{u/x\}$;
- $t \equiv \lambda y.v$, avec $y \neq x$ et avec $y \in FV(u)$ et $x \in FV(v)$. On considère $z \notin FV(v) \cup FV(u)$ et on pose $t\{u/x\} = \lambda z.v'$ avec $v' = v\{z/y\}\{u/x\}$.

Remarque 2.8

1. Pour que la définition précédente soit bien définie, il faut noter qu'on la fait non pas simplement par induction sur la structure du terme auquel s'appliquer la substitution, mais par récurrence sur la taille du terme (du nombre de noeuds de l'arbre associé) et que la substitution d'une variable par une variable ne modifie pas la taille d'un terme. En effet, la dernière clause de la définition applique deux substitutions ce qui n'est pas une définition inductive bien définie en général. Alternativement, on peut utiliser la deuxième tentative de définition de la substitution, ci-dessus (dite avec capture, qui ne s'appliquerait ici qu'aux variables et notée $u[y/x]$) et remplacer le dernier cas par $v' = v[z/y]\{u/x\}$, mais on doit toujours faire l'induction sur la taille du terme.
2. On peut en fait simplifier la définition de la substitution en adoptant la définition suivante pour le cas où t est une abstraction:
 - $t \equiv \lambda y.v$, on considère $z \notin FV(v) \cup FV(u) \cup \{x\}$ et on pose $t\{u/x\} = \lambda z.v'$ avec $v' = v\{z/y\}\{u/x\}$.
3. À ce point, il est naturel de se demander comment se fait le choix de la variable z dans la

dernière clause définissant la substitution. En effet, il y a une infinité de choix possibles: la substitution ne paraît donc pas bien définie à ce point. En fait, on se rappelle qu'on travaille maintenant sur les classe d' α -équivalence et on vérifie aisément que le résultat de la substitution ne dépend pas du représentant choisi pour la classe d'équivalence (de départ) ni de la variable choisie pour la dernière clause de la définition.

Proposition 2.9

La définition précédente est bien formée (c'est-à-dire que la classe d' α -équivalence ne dépend ni des représentants choisis, ni du choix de la variable z dans la dernière clause de la définition).

Il est en fait possible de travailler directement sur les classes d'équivalence car il existe une représentation canonique de ces classes, appelée syntaxe de de Bruijn:

Définition 2.10 (Syntaxe de de Bruijn)

Les λ -termes en notation de de Bruijn sont définis par la syntaxe suivante:

$$t, u, v ::= n \mid \lambda t \mid (t)u \quad n \geq 0$$

Remarque 2.11

L'idée de la notation de De Bruijn est de désigner les variables par le nombre de λ à rencontrer (en remontant dans l'arbre) pour atteindre le lieu: $\lambda\lambda 1$, $\lambda\lambda 2$ et $\lambda\lambda 3$ correspondent respectivement à $\lambda x.\lambda y.y$, $\lambda x.\lambda y.x$, $\lambda x.\lambda y.z$.

Les λ -termes en notation de De Bruijn sont ainsi directement les classes d' α -équivalence sur Λ .

Exercice 2.1

Définir la notion de terme clos pour la syntaxe de de Bruijn puis une bijection entre les λ -termes en notation de de Bruijn clos et (les classes d' α -équivalence) des λ -termes clos. Comment définir la substitution et la β -réduction avec la notation de De Bruijn?

La signification intuitive des constructions du lambda-calcul étant celle de construction et d'application de fonction, un λ -terme de la forme $(\lambda x.t)u$ représente donc l'application à l'argument u de la fonction qui a x associée t . Il semble donc naturel de vouloir relier ce terme à $t\{u/x\}$.

Définition 2.12 (Relation β)

On appelle β la relation sur Λ^2 définie comme:

$$\beta = \{((\lambda x.t)u, t\{u/x\}); t, u \in \Lambda, x \in \mathcal{V}\}.$$

Définition 2.13 (β -équivalence)

La β -équivalence, notée $=_\beta$, est la plus petite congruence contenant β .

Définition 2.14 (β -réduction)

La β -réduction, notée $\rightarrow_\beta$, est la plus petite relation compatible contenant la relation β . On notera respectivement $\rightarrow_\beta^*$, $\rightarrow_\beta^+$, $\rightarrow_\beta^{\leq 1}$ les: clôture réflexive et transitive, clôture transitive et clôture réflexive de la β -réduction (notation alternative pour la dernière: $\rightarrow_\beta^{0/1}$).

Remarque 2.15

Plus généralement, on utilisera les notations précédentes pour une relation binaire ρ sur un ensemble A quelconque: ρ^* , ρ^+ , $\rho^{\leq 1}$.

Voici maintenant quelques propriétés concernant les relations entre β -réduction et substitution:

Propriété 2.16

Soient $t, u, v, w \in \Lambda$ et $x \in \mathcal{V}$. On a:

1. Si $v \rightarrow_{\beta}^* w$ alors $t\{v/x\} \rightarrow_{\beta}^* t\{w/x\}$;
2. Si $t \rightarrow_{\beta}^* u$ alors $t\{v/x\} \rightarrow_{\beta}^* u\{v/x\}$;
3. Si $t \rightarrow_{\beta}^* u$ et $v \rightarrow_{\beta}^* w$ alors $t\{v/x\} \rightarrow_{\beta}^* u\{w/x\}$.

Démonstration : Les deux premières propriétés se montrent par induction sur t (et sur la longueur de la réduction pour la seconde propriété) tandis que la troisième se déduit immédiatement en combinant les deux premières. $\square$

Exercice 2.2

Rédiger la démonstration de la propriété 2.16.

On dit, d'après la propriété 2.16.2, que la β -réduction est substitutive.

Définition 2.17 (*Radical, forme normale*)

Un **radical** est un terme de la forme $(\lambda x.t)u$, tandis qu'une **forme normale** est un terme qui ne contient aucun radical (parmi ses sous-termes).

Définition 2.18 (*Réduction, première version*)

En première approximation, on considérera qu'une **réduction** est une famille de λ -termes $(t_i)_{i \in \mu \in \omega+1}$ telle que pour tout $i \in \mu$, si $i+1 \in \mu$ alors $t_i \rightarrow_{\beta} t_{i+1}$.

La définition précédente sera raffinée par la suite car on souhaitera distinguer, dans une suite de termes en relation par $\rightarrow_{\beta}$, les radicaux qui sont responsables de la réduction et, pour une même suite de terme, il peut y en avoir plusieurs, comme on le voit aisément en considérant les termes de la forme $(I)(I)(I)\dots(I)I$ où $I = \lambda x.x$. La section suivante sur les graphes de réduction, amorcera ce raffinement de la notion de séquences de réductions.

Définition 2.19 (*Terme normalisable, terme fortement normalisable*)

Un λ -terme est dit (**faiblement**) **normalisant** (ou **normalisable**) lorsqu'il existe une forme normale u telle que $t \rightarrow_{\beta}^* u$; on dira également que t **possède une forme normale**.

Un λ -terme t est dit **fortement normalisant** (ou **terminant**) si toutes ses réductions sont finies.

Un ensemble de λ -termes $L \subseteq \Lambda$ est dit **fortement** (resp. **faiblement**) **normalisant** si tous ses éléments sont **fortement** (resp. **faiblement**) **normalisants**.

Proposition 2.20

1. Λ n'est pas fortement normalisant.
2. Il existe des λ -termes qui sont faiblement normalisants sans être fortement normalisants.

Démonstration :

1. $\Omega = (\Delta)\Delta$, avec $\Delta = \lambda x.(x)x$, est un terme dont l'unique radical est lui-même et tel que $\Omega \rightarrow_{\beta} \Omega$.
2. $t = (\lambda x.y)\Omega$ peut se réduire soit sur y si on contracte le radical $(\lambda x.y)\Omega$, soit sur lui-même si on contracte le radical Ω . En conséquence, t est faiblement mais non fortement normalisant.

$\square$

Exercice 2.3

Montrer que la β -réduction est non-déterministe, c'est-à-dire qu'il existe des termes t, u, v tels

ue $t \longrightarrow_{\beta} u, v$ avec $u \neq v$.

Exercice 2.4 (B. Lercher)

Montrer que Ω est le seul λ -terme de la forme $(\lambda x.t)u$ tel que $(\lambda x.t)u = t \{u/x\}$.

3 Graphes des réductions

Définition 3.1 (Graphe des réductions d'un λ -terme)

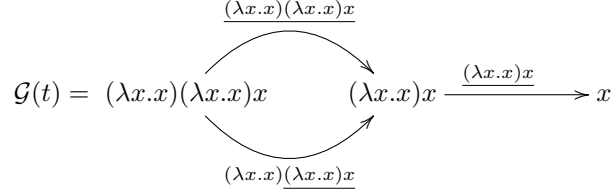
À tout lambda terme t , on associe un graphe orienté (qui peut contenir des arêtes multiples et des boucles), dont les noeuds sont étiquetés par des λ -termes et les arêtes par des radicaux, noté $\mathcal{G}(t) = (\mathcal{N}_t, \mathcal{A}_t)$ comme suit:

- $\mathcal{N}_t = \{u \in \Lambda; t \longrightarrow_{\beta}^* u\}$ c'est-à-dire que les noeuds sont les termes vers lesquels t peut se réduire;
- les arêtes ($a \in \mathcal{A}_t$) sont données par la relation de β -réduction: à chaque réduction d'un terme u vers un terme v par la réduction d'un(e occurrence de) redex ρ , correspond une arête de $\mathcal{G}(t)$, $a_{(u,v)} \in \mathcal{A}_t$, de source u , de cible v et étiquetée par ρ .

Remarque 3.2

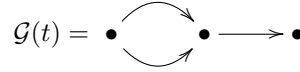
On remarquera qu'il arrive qu'un terme t puisse se réduire sur un terme u de différentes manières parce qu'il possède plusieurs radicaux: chacune de ces réductions donnera lieu à une arête différente dans le graphe des réductions.

Par exemple $t = (\lambda x.x)(\lambda x.x)x$ et on a alors:

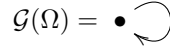


On verra plus tard dans le cours comment décrire de manière plus précise, à travers la notion d'occurrences, les radicaux présents dans un λ -terme ce qui lèvera la petite imprécision présente actuellement dans la définition des graphes de réductions pour la notion d'arête multiple.

Souvent, on n'écrira pas les termes explicitement, mais seulement des points représentant différents termes sur lesquels t peut se réduire:



Autre exemple, on a:



Définition 3.3 (Fonction d'oubli)

Étant donné un graphe de réduction $\mathcal{G}$, on définit $F(\mathcal{G})$ le graphe obtenu en oubliant les étiquettes des noeuds et des arêtes de G .

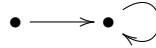
Définition 3.4 (Réalisateur)

On dira qu'un λ -terme t **réalise** un graphe avec multiarêtes et boucles, G , si $G = F(\mathcal{G}(t))$.
Un graphe est **réalisable** s'il existe un terme le réalisant.

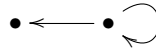
Exemple 3.5 (Sur les graphes de réduction)

Donner, lorsque c'est possible, un λ -terme dont le graphe de réduction correspond aux graphes présentés ci-dessous. Le cas échéant, argumenter de l'impossibilité de réaliser un graphe donné.

1. $\star$



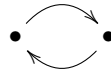
2. $\star$



3. $\star$



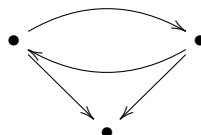
4. $\star$



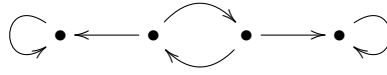
5.



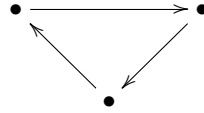
6. Ex



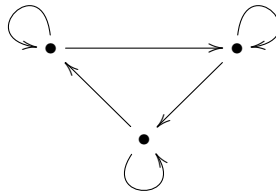
7. *Ex*



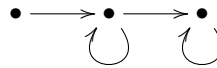
8. *Ex*



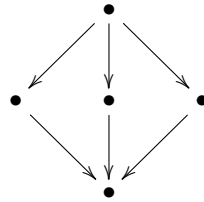
9.



10. $\star$



11.



Exercice 3.1

Donner un λ -terme réalisant les graphes de l'exemple précédent dont le numéro est suivi de la mention "*Ex*".

4 Exercices du chapitre

Exercice 4.1

Voici quelques exemples de λ -termes:

- $\lambda x.x$;
- $(x)\lambda x.\lambda x.x$;
- $\lambda x.(y)z$;
- $\lambda x.(x)(x)y$.
- $\lambda x.\lambda y.\lambda z.((z)x)y$;
- $\lambda x.\lambda y.x$;
- $\lambda x.\lambda y.y$;
- $(\lambda x.(x)x)\lambda y.(y)y$.

Appliquer aux exemples ci-dessus les simplification de parenthésage et notation introduite plus haut et les représenter sous forme arborescente.

Exercice 4.2

Les λ -termes en notation de de Bruijn sont définis par la syntaxe suivante:

$$t, u, v ::= n \mid \lambda t \mid (t)u \quad n \geq 0$$

Définir la notion de terme clos pour la syntaxe de de Bruijn puis une bijection entre les λ -termes en notation de de Bruijn clos et (les classes d' α -équivalence) des λ -termes clos.

Comment définir la substitution et la β -réduction avec la notation de De Bruijn?

Exercice 4.3

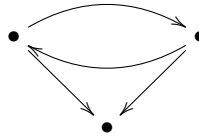
Démontrer les propriétés suivantes: Soient $t, u, v, w \in \Lambda$ et $x \in \mathcal{V}$. On a:

1. Si $v \rightarrow_{\beta}^* w$ alors $t\{v/x\} \rightarrow_{\beta}^* t\{w/x\}$;
2. Si $t \rightarrow_{\beta}^* u$ alors $t\{v/x\} \rightarrow_{\beta}^* u\{v/x\}$;
3. Si $t \rightarrow_{\beta}^* u$ et $v \rightarrow_{\beta}^* w$ alors $t\{v/x\} \rightarrow_{\beta}^* u\{w/x\}$.

Exercice 4.4 (*Sur les graphes de réduction*)

Donner, lorsque c'est possible, un λ -terme dont le graphe de réduction correspond aux graphes présentés ci-dessous.

1. Ex



2. Ex



3. Ex

