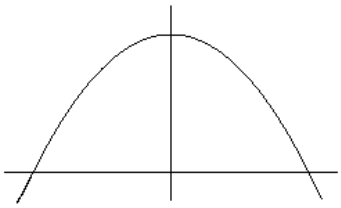
Gilles Dowek

Les démonstrations et les algorithmes

Introduction à la logique et à la calculabilité

L'auteur tient à remercier René Cori, René David, Maribel Fernández, Jean-Baptiste Joinet, Claude Kirchner, Jean-Louis Krivine, Daniel Lascar, Stéphane Lengrand, Michel Parigot, Laurence Rideau et Paul Rozière.

Introduction



Plusieurs méthodes permettent de déterminer l'aire de ce segment de parabole. Une première consiste à le découper en une infinité de petits triangles, puis à déterminer l'aire de chacun d'eux par une démonstration et à faire la somme des aires obtenues. C'est *grosso modo* la méthode qu'Archimède a employée pour démontrer que cette aire était égale à $4/3$. Depuis le XVII$^{\text{e}}$ siècle, toutefois, on peut utiliser une autre méthode, qui donne le même résultat, et qui consiste à déterminer la valeur de l'intégrale $\int_{-1}^{1}(1-x^2)dx$. Intégrer cette fonction polynomiale ne demande pas de construire une démonstration, mais simplement d'appliquer un algorithme.

Construire une démonstration, appliquer un algorithme, ces deux types de méthodes coexistent depuis longtemps au sein des mathématiques, mais les méthodes algorithmiques ont connu un nouvel essor depuis l'apparition des ordinateurs, qui permettent de les utiliser à une toute autre échelle que par le passé.

La coexistence de ces deux méthodes de résolution des problèmes mène à s'interroger sur leurs relations. Jusqu'à quel point peut-on remplacer la construction d'une démonstration par l'application d'un algorithme ? Ce livre

est consacré à un ensemble de résultats, tant négatifs que positifs, qui répondent partiellement à cette question.

Pour parvenir à ces résultats, nous devrons commencer par définir précisément ces notions de démonstration, dans la première partie, et d'algorithme, dans la deuxième. Définir la notion de démonstration permettra de comprendre comment démontrer des théorèmes d'indépendance, qui affirment qu'il n'existe pas de démonstration pour résoudre certains problèmes. Définir la notion d'algorithme permettra de comprendre comment démontrer des théorèmes d'indécidabilité, qui affirment qu'il n'existe pas d'algorithme pour résoudre certains problèmes. Cela nous permettra aussi de comprendre que les algorithmes peuvent se présenter sous de nombreuses formes : ensembles de règles de réécriture, termes du lambda-calcul, machines de Turing, mais que cette diversité masque une profonde unité : l'idée qu'un calcul est une suite de petits pas.

La troisième partie aborde enfin les liens entre les notions de démonstration et d'algorithme. Le résultat central de cette partie est le théorème de Church, qui montre que la démontrabilité dans la logique des prédicats est un problème indécidable, et dont un corollaire est le célèbre théorème de Gödel. Ce résultat négatif sera cependant nuancé par deux résultats positifs. Tout d'abord, s'il est indécidable, ce problème est semi-décidable, ce qui nous mènera à développer des algorithmes de recherche de démonstrations. Ensuite, ajouter des axiomes à la logique des prédicats permet, dans certains cas, de rendre la démontrabilité décidable, ce qui nous mènera à développer des algorithmes de décision pour des théories particulières.

Un dernier chapitre nous montrera un lien différent entre les démonstrations et les algorithmes : certaines démonstrations, que l'on appelle *constructives*, peuvent être utilisées comme des algorithmes.

Au fil des pages, se révélera donc la richesse des liens entre ces notions de démonstration et d'algorithme, mais aussi la complexité qui se cache derrière l'apparente évidence de la notion de vérité.

Première partie

Les démonstrations

1
La logique des prédicats

Quelles sont les conditions que doit vérifier une proposition pour être vraie ? Une réponse possible, qui définit un certain type de vérité, est qu'une proposition est vraie quand elle est démontrable. Dans ce chapitre, nous allons détailler cette réponse en donnant une définition de cette notion de démontrabilité. Pour cela, nous allons définir, dans un premier temps, l'ensemble des *propositions*, puis, dans un second temps, celui des *théorèmes*, ou *propositions démontrables*, qui en constitue un sous-ensemble.

Ces deux définitions sont des définitions d'ensembles. Commençons donc par nous interroger sur les outils qui permettent de définir des ensembles.

1.1 Les définitions inductives

L'outil le plus simple pour définir un ensemble est la notion de *définition explicite*. On peut, par exemple, définir explicitement l'ensemble des nombres pairs : $\{n \in \mathbb{N} \mid \exists p \in \mathbb{N}\ n = 2 \times p\}$. Cependant, ces définitions explicites ne suffisent pas à définir tous les ensembles dont on a besoin. Un deuxième outil utile pour définir des ensembles est la notion de *définition inductive*. Cet outil s'appuie sur un théorème simple : le théorème du point fixe.

1.1.1 Le théorème du point fixe

Définition 1.1 (Limite)

Soit $\leq$ une relation d'ordre, c'est-à-dire une relation réflexive, antisymétrique et transitive, sur un ensemble E et $u_0, u_1, u_2, \ldots$ une suite croissante, c'est-à-dire telle que $u_0 \leq u_1 \leq u_2 \leq \ldots$ L'élément l de E est appelé *limite* de la suite $u_0, u_1, u_2, \ldots$ si c'est la borne supérieure de l'ensemble $\{u_0, u_1, u_2, \ldots\}$, c'est-à-dire si c'est un majorant de cet ensemble

– pour tout i, $u_i \leq l$

et si c'est le plus petit

– si pour tout i, $u_i \leq l'$, alors $l \leq l'$.

Si elle existe, la limite d'une suite $(u_i)_i$ est unique et on la note $\lim_i\ u_i$.

Définition 1.2 (Relation d'ordre faiblement complète)

La relation d'ordre $\leq$ est dite *faiblement complète* si toutes les suites croissantes ont une limite.

La relation d'ordre ordinaire sur l'intervalle $[0, 1]$ de la droite réelle est un exemple de relation d'ordre faiblement complète. De plus, cette relation a un plus petit élément : 0. En revanche, la relation d'ordre ordinaire sur $\mathbb{R}^+$ n'est pas faiblement complète car la suite croissante $0, 1, 2, 3, \ldots$ n'a pas de limite.

Soit A un ensemble quelconque, la relation d'inclusion $\subseteq$ sur l'ensemble $\wp(A)$ des parties de A est un autre exemple de relation d'ordre faiblement complète. La limite d'une suite croissante $U_0, U_1, U_2, \ldots$ est l'ensemble $\bigcup_{i\in\mathbb{N}} U_i$. De plus, cette relation a un plus petit élément : $\varnothing$.

Définition 1.3 (Fonction croissante)

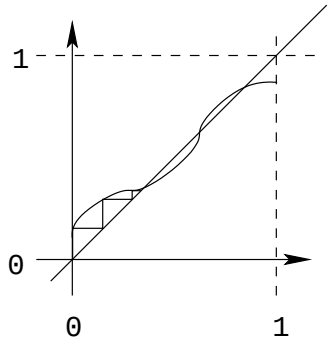
Soit $\leq$ une relation d'ordre définie sur un ensemble E et f une fonction de E dans E. La fonction f est *croissante* si $x \leq y \Rightarrow fx \leq fy$.

Définition 1.4 (Fonction continue)

Soit $\leq$ une relation d'ordre faiblement complète définie sur un ensemble E et f une fonction croissante de E dans E. La fonction f est *continue* si pour toute suite croissante $\lim_i\ (f\ u_i) = f\ (\lim_i\ u_i)$.

Proposition 1.1 (Le premier théorème du point fixe)

Soit $\leq$ une relation d'ordre faiblement complète sur un ensemble E qui a un plus petit élément m. Soit f une fonction de E dans E. Si f est continue, alors $p = \lim_i (f^i\ m)$ est le plus petit point fixe de f.



Démonstration. Tout d'abord, m étant le plus petit élément de E, $m \leq fm$. La fonction f étant croissante, $f^i m \leq f^{i+1} m$. La suite $f^i m$ est croissante, elle a donc bien une limite. La suite $f^{i+1} m$ a également p pour limite. Donc $p = \lim_i (f\ (f^i\ m)) = f\ (\lim_i (f^i\ m)) = f\ p$. De plus, p est le plus petit point fixe, car si q est un point fixe, $m \leq q$ et, la fonction f étant croissante, $f^i m \leq f^i q = q$. Donc $p = \lim_i (f^i\ m) \leq q$.

Le second théorème du point fixe donne l'existence d'un point fixe pour les fonctions croissantes, même si elles ne sont pas continues, en supposant une propriété un peu plus forte sur la relation d'ordre.

Définition 1.5 (Relation d'ordre fortement complète)

Une relation d'ordre $\leq$ sur un ensemble E est *fortement complète* si tout sous-ensemble A de E a une borne supérieure, $\sup A$.

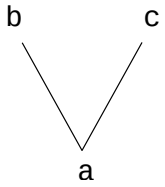
La relation d'ordre ordinaire sur l'intervalle $[0, 1]$ de la droite réelle est un exemple de relation d'ordre fortement complète. La relation d'ordre ordinaire sur $\mathbb{R}^+$ n'est pas fortement complète, car l'ensemble $\mathbb{R}^+$ tout entier n'a pas de borne supérieure.

Soit A un ensemble quelconque, la relation d'inclusion $\subseteq$ sur l'ensemble $\wp(A)$ des parties de A est un autre exemple de relation d'ordre fortement complète. La borne supérieure d'un ensemble B est l'ensemble $\bigcup_{C \in B} C$.

Exercice 1.1

Montrer que toute relation fortement complète est faiblement complète.

La relation d'ordre



est-elle faiblement complète ? Est-elle fortement complète ?

Proposition 1.2

Si la relation d'ordre $\leq$ sur l'ensemble E est fortement complète, alors tout sous-ensemble A de E a une borne inférieure, $\inf A$.

Démonstration. Soit A un sous-ensemble quelconque de E, soit B l'ensemble $\{y \in E \mid \forall x \in A\ y \leq x\}$ des minorants de A et l la borne supérieure de B. Par définition, l est un majorant de l'ensemble B

– $\forall y \in B\ y \leq l$

et c'est le plus petit

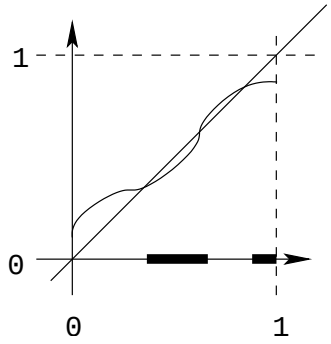
– $(\forall y \in B\ y \leq l') \Rightarrow l \leq l'$.

Il n'est pas difficile de démontrer que l est la borne inférieure de A. En effet, si x est un élément de A, c'est un majorant de B et comme l est le plus petit de ces majorants, $l \leq x$. Donc l est un minorant de A. Pour montrer que c'est le plus grand, il suffit de remarquer que si m est un minorant de A, c'est un élément de B et donc $m \leq l$.

La borne inférieure d'un sous-ensemble B de l'ensemble des parties d'un ensemble A est, bien entendu, l'ensemble $\bigcap_{C \in B} C$.

Proposition 1.3 (Le second théorème du point fixe)

Soit $\leq$ une relation d'ordre fortement complète sur un ensemble E. Soit f une fonction de E dans E. Si f est croissante, alors $p = \inf\{c \mid fc \leq c\}$ est le plus petit point fixe de f.



Démonstration. Soit C l'ensemble $\{c \mid fc \leq c\}$ et c un élément de C. On a $p \leq c$ car p est un minorant de C. La fonction f étant croissante, on en déduit $fp \leq fc$. Par ailleurs, $fc \leq c$ car c est un élément de C, donc par transitivité $fp \leq c$.

L'élément fp est inférieur à tous les éléments de C, il est donc inférieur à sa borne inférieure : $fp \leq p$.

La fonction f étant croissante, $f(fp) \leq fp$, donc fp est un élément de C et p étant un minorant de C, on en déduit $p \leq fp$. Par antisymétrie, $p = fp$.

Enfin, par définition, tous les points fixes de f appartiennent à C, ils sont donc plus grands que p.

1.1.2 Les définitions inductives

Voyons maintenant comment ce théorème du point fixe permet de définir des ensembles et des relations.

Définition 1.6 (Fermeture)

Soit E un ensemble, f une fonction partielle de E^n dans E et A un sous-ensemble de E. L'ensemble A est dit *fermé* par la fonction f si pour tous $x_1, \ldots, x_n$ dans A, tels que la fonction f soit définie en $x_1, \ldots, x_n$, $f\ x_1\ \ldots\ x_n$ est également un élément de A.

Par exemple, l'ensemble des nombres pairs est fermé par la fonction $n \mapsto n + 2$.

Définition 1.7 (Définition inductive)

Soit E un ensemble, une *définition inductive* sur E est une famille de fonctions

partielles f_1 de E^{n_1} dans E, f_2 de E^{n_2} dans E, ... Cette famille définit un sous-ensemble A de E : le plus petit sous-ensemble de E fermé par les fonctions f_1, f_2, ...

Par exemple, le sous-ensemble de $\mathbb{N}$ formé des nombres pairs est défini inductivement par l'entier 0, c'est-à-dire la fonction, de $\mathbb{N}^0$ dans $\mathbb{N}$, qui prend la valeur 0, et la fonction, de $\mathbb{N}$ dans $\mathbb{N}$, $n \mapsto n+2$. L'ensemble des nombres pairs n'est pas l'unique sous-ensemble de $\mathbb{N}$ qui contient 0 et qui est fermé par la fonction $n \mapsto n+2$, l'ensemble $\mathbb{N}$, par exemple, vérifie également ces propriétés, mais c'est le plus petit.

Le sous-ensemble de $\{a, b, c\}^*$ formé des mots de la forme $a^n b c^n$ est défini inductivement par le mot b et la fonction $m \mapsto amc$. D'une manière générale, une grammaire non contextuelle peut toujours se formuler comme une définition inductive.

Comme nous allons le voir, l'ensemble des théorèmes se définit comme le sous-ensemble de l'ensemble des propositions défini inductivement par les axiomes et les règles de déduction.

Les fonctions f_1, f_2, ... sont appelées des *règles*. Au lieu de noter une telle règle $x_1 \dots x_n \mapsto t$ on la note

$$\frac{x_1 \dots x_n}{t}$$

Par exemple, l'ensemble des nombres pairs est défini par les deux règles

$$\frac{}{0}$$

$$\frac{n}{n+2}$$

En notant P l'ensemble des nombres pairs, on écrit aussi parfois ces règles

$$\frac{}{0 \in P}$$

$$\frac{n \in P}{n+2 \in P}$$

Pour donner un sens à la définition 1.7, montrons qu'il existe toujours un plus petit sous-ensemble A fermé par les fonctions f_1, f_2, ...

Proposition 1.4

Soit E un ensemble et $f_1, f_2, \dots$ des règles sur l'ensemble E. Il existe un plus petit sous-ensemble A de E fermé par les fonctions f_1, f_2, ...

Démonstration. Soit F la fonction de $\wp(E)$ dans $\wp(E)$

$$FC = \{x \in E \mid \exists i \exists y_1 \dots y_{n_i} \in C \; x = f_i \; y_1 \; \dots \; y_{n_i}\}$$

Un sous-ensemble C de E est fermé par les fonctions f_1, f_2, ... si et seulement si $FC \subseteq C$.

La fonction F est trivialement croissante : si $C \subseteq C'$, alors $FC \subseteq FC'$. On définit l'ensemble A comme le plus petit point fixe de cette fonction : comme l'intersection de tous les ensembles C tels que $FC \subseteq C$, c'est-à-dire comme l'intersection de tous les ensembles fermés par les fonctions f_1, f_2, ...

D'après le second théorème du point fixe, cet ensemble est un point fixe de F, $FA = A$, et donc $FA \subseteq A$. Il est donc fermés par les fonctions f_1, f_2, ... Et par définition, il est plus petit que tous les ensembles C tels que $FC \subseteq C$, c'est donc le plus petit ensemble fermé par ces fonctions.

Le premier théorème du point fixe nous donne une autre caractérisation de cet ensemble.

Proposition 1.5

Soit E un ensemble et $f_1, f_2, \dots$ des règles sur l'ensemble E. Le plus petit sous-ensemble A de E fermé par les fonctions f_1, f_2, ... est l'ensemble $\bigcup_k (F^k \varnothing)$ où la fonction F est définie par

$$FC = \{x \in E \mid \exists i \exists y_1 \dots y_{n_i} \in C \; x = f_i \; y_1 \; \dots \; y_{n_i}\}$$

Démonstration. On a vu que la fonction F est croissante. Elle est, de plus, continue : si $C_0 \subseteq C_1 \subseteq C_2 \subseteq \dots$, alors $F(\bigcup_j C_j) = \bigcup_j (FC_j)$. En effet, si un élément x de E est dans $F(\bigcup_j C_j)$, alors il existe un entier i et des éléments $y_1, \dots, y_{n_i}$ de $\bigcup_j C_j$ tels que $x = f_i \; y_1 \; \dots \; y_{n_i}$. Chacun de ces éléments est dans l'un des C_j. Comme la suite des C_j est croissante, ils sont tous dans C_k, le plus grand de ces ensembles. L'élément x appartient donc à FC_k et donc à $\bigcup_j (FC_j)$. Réciproquement, si x appartient à $\bigcup_j (FC_j)$, il appartient à un certain FC_k, il existe donc un entier i et des éléments $y_1, \dots, y_{n_i}$ de C_k tels que $x = f_i \; y_1 \; \dots \; y_{n_i}$. Les éléments $y_1, \dots, y_{n_i}$ appartiennent à $\bigcup_j C_j$ et donc x à $F(\bigcup_j C_j)$.

On a vu que le plus petit sous-ensemble A de E fermé par les fonctions f_1, f_2, ... est le plus petit point fixe de la fonction F. D'après le premier théorème du point fixe, cet ensemble est $A = \bigcup_k (F^k \varnothing)$.

1.1.3 La récurrence structurelle

Les définitions inductives donnent un moyen de faire des démonstrations par récurrence. Si une propriété est *héréditaire*, c'est-à-dire qu'à chaque fois qu'elle est vérifiée par $y_1, \ldots, y_{n_i}$, elle est vérifiée par $f_i\ y_1\ \ldots\ y_{n_i}$, alors elle est vérifiée par tous les éléments de A.

Une manière de démontrer cela est d'utiliser le second théorème du point fixe et de remarquer que le sous-ensemble P de E des objets qui vérifient la propriété en question est fermé par les fonctions f_i et donc qu'il contient A. Une autre manière est d'utiliser la proposition 1.5 et de montrer, par récurrence sur k, que tous les objets de $F^k\varnothing$ vérifient la propriété en question.

1.1.4 Les dérivations

Un élément x appartient à l'ensemble A si et seulement s'il appartient à un certain ensemble $F^k\varnothing$, c'est-à-dire s'il existe une fonction f_i telle que $x = f_i\ y_1\ \ldots\ y_{n_i}$ où les $y_1, \ldots, y_{n_i}$ appartiennent à $F^{k-1}\varnothing$. Cette remarque permet de démontrer qu'un élément x de E appartient à A si et seulement s'il existe un arbre dont les nœuds sont étiquetés par des éléments de E, dont la racine est étiquetée par x, et tel que si un nœud est étiqueté par un élément y et ses enfants sont étiquetés par des éléments $z_1, \ldots, z_n$, alors il existe une règle f_i, telle que $y = f_i\ z_1\ \ldots\ z_n$. Un tel arbre s'appelle une *dérivation* de x.

Définition 1.8 (Dérivation)

Soit E un ensemble et $f_1, f_2, \ldots$ des règles sur l'ensemble E. Une *dérivation* dans $f_1, f_2, \ldots$ est un arbre dont les nœuds sont étiquetés par des éléments de E tel que si un nœud est étiqueté par un élément y et ses enfants par des éléments $z_1, \ldots, z_n$, alors il existe une règle f_i, telle que $y = f_i\ z_1\ \ldots\ z_n$.

Si la racine d'une dérivation est un élément x de E, alors cette dérivation est une *dérivation de* x.

On peut donc définir l'ensemble A comme l'ensemble des éléments de E qui ont une dérivation.

On utilise une écriture particulière pour les dérivations. Tout d'abord, on écrit la racine de l'arbre en bas et les feuilles en haut. Ensuite, on trace un trait au-dessus de chaque nœud de l'arbre et on écrit ses enfants au-dessus de ce trait.

Le nombre 8, par exemple, est dans l'ensemble des nombres pairs, en voici une dérivation

$$\dfrac{\dfrac{\dfrac{\dfrac{0}{2}}{4}}{6}}{8}$$

En notant P l'ensemble des nombres pairs, on écrit aussi parfois cette dérivation

$$\dfrac{\dfrac{\dfrac{\dfrac{0 \in P}{2 \in P}}{4 \in P}}{6 \in P}}{8 \in P}$$

Au lieu d'étiqueter les nœuds d'une dérivation par des éléments de E, on peut aussi l'étiqueter par des règles.

Définition 1.9 (Dérivation étiquetée par les règles)

Soit E un ensemble et $f_1, f_2, \ldots$ des règles sur l'ensemble E. Une *dérivation étiquetée par les règles* $f_1, f_2, \ldots$ est un arbre dont les nœuds sont étiquetés par $f_1, f_2, \ldots$ tel que le nombre d'enfants d'un nœud étiqueté par une fonction f soit le nombre d'arguments de f.

À chaque dérivation étiquetée par les règles, on peut associer, par récurrence structurelle, un élément de E : si la racine de la dérivation est étiquetée par la règle f_i et aux sous-arbres immédiats sont associés les éléments $z_1, \ldots, z_n$, alors on associe l'élément $f_i\ z_1\ \ldots\ z_n$ à la dérivation elle-même.

Quand un élément est associé à une dérivation, on dit que la dérivation est une *dérivation de cet élément.*

On peut donc définir l'ensemble A comme l'ensemble des éléments de E qui ont une dérivation étiquetée par les règles.

1.1.5 La fermeture réflexive-transitive d'une relation

Un exemple de définition inductive est celle de la fermeture réflexive-transitive d'une relation.

Définition 1.10 (Fermeture réflexive-transitive)

Soit R est une relation binaire sur un ensemble E, la *fermeture réflexive-transitive* de la relation R est la relation R^* inductivement définie par les règles
- $t\ R^*\ t$,
- si $t\ R\ t'$ et $t'\ R^*\ t''$, alors $t\ R^*\ t''$.

Si $t\ R^*\ t'$, une dérivation du couple (t, t') est une suite finie $t_0, \ldots, t_n$, telle que $t_0 = t$, $t_n = t'$ et pour tout $i \leq n-1$, $t_i\ R\ t_{i+1}$.

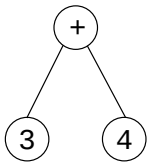
Si on voit R comme un graphe orienté, les dérivations sont les chemins de ce graphe et la relation R^* relie donc deux sommets, quand il y a un chemin qui mène de l'un à l'autre.

1.2 Les langages

1.2.1 Les langages sans variables

Maintenant que nous avons introduit la notion de définition inductive, nous allons l'utiliser pour définir la notion de *langage*. Nous allons dans un premier temps définir une notion très générale qui comprendra aussi bien les langages de programmation que les langages logiques. Puis nous définirons, dans un second temps, les langages de la logique des prédicats.

Nous cherchons, par ailleurs, à définir une notion de langage qui s'affranchit des conventions syntaxiques superficielles, par exemple de savoir si on écrit 3+4, $+(3, 4)$, ou encore 3 4 +. Cette expression sera plus abstraitement exprimée par un arbre



Chaque nœud de cet arbre est étiqueté par un symbole. Le nombre d'enfants d'un nœud de l'arbre dépend du symbole qui l'étiquette — 2 enfants si ce symbole est +, 0 si c'est 3 ou 4, ...

Un langage est donc un ensemble de symboles munis d'un entier appelé *arité*, ou plus simplement *nombre d'arguments*, de ce symbole. Les symboles sans arguments sont appelés des *constantes*.

L'ensemble des *expressions* de ce langage est l'ensemble d'arbres défini inductivement par la règle suivante.

– Si f est un symbole d'arité n et t_1, ..., t_n sont des expressions alors $f(t_1, \ldots, t_n)$, c'est-à-dire l'arbre dont la racine est étiquetée par f et dont les sous-arbres immédiats sont t_1, ..., t_n, est une expression.

1.2.2 Les variables

Imaginons que nous voulions définir un langage qui contient des expressions comme *impair*(3) ou *impair*(3) $\Rightarrow$ *pair*(3+1). Nous voudrons alors probablement aussi pouvoir exprimer que pour tout entier, si cet entier est impair, alors son successeur est pair.

Pour former de telles expressions, les langues naturelles, comme le français, utilisent des pronoms indéfinis, par exemple *tous* et *quelques*. Mais ce mécanisme est ambigu quand plusieurs expressions sont remplacées par de tels pronoms. Ainsi, la phrase « Il existe un nombre entier supérieur à tout nombre entier » peut signifier ou bien que pour chaque nombre entier, il existe un nombre entier qui lui est supérieur, ce qui est vrai, ou bien qu'il existe un nombre qui est supérieur à tous les nombres entiers, ce qui est faux. On utilise donc un mécanisme plus complexe, qui consiste à utiliser dans un premier temps une variable, dont on indique ensuite la signification et la portée par un quantificateur $\forall$, *pour tout*, ou $\exists$, *il existe*, qui lie cette variable. Ainsi, on distingue les propositions $\forall x \exists y\ (y \geq x)$ et $\exists y \forall x\ (y \geq x)$.

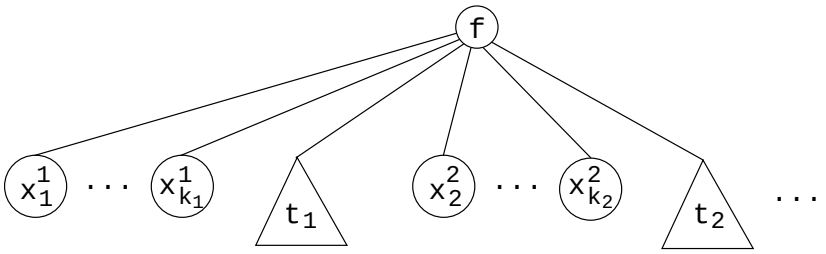
Les quantificateurs sont donc des symboles qui lient une variable dans leur argument. D'autres exemples de symboles lieurs sont les symboles $\mapsto$, ∂/∂, $\int d$, $\sum$, $\prod$, ... Nous devons donc étendre la notion de langage définie ci-avant de manière à prendre en compte le fait que chaque symbole du langage peut lier des variables.

L'arité d'un symbole f ne sera désormais plus un entier n, mais une suite finie d'entiers $(k_1, \ldots, k_n)$ qui indique que le symbole f lie k_1 variables dans son premier argument, k_2 variables dans le deuxième, ..., k_n variables dans le n-ième.

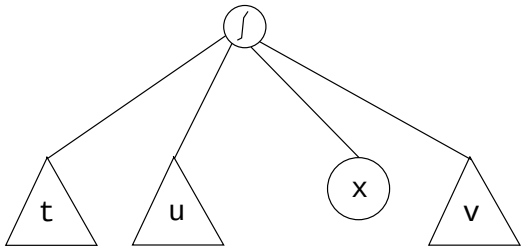
Ainsi, quand on s'est donné un langage, c'est-à-dire un ensemble de symboles munis d'une arité, et un ensemble infini dont les éléments sont appelés *variables*, on définit les expressions inductivement par les règles suivantes.

– Les variables sont des expressions.
– Si f est un symbole d'arité $(k_1, \ldots, k_n)$, t_1, ..., t_n sont des expressions et x_1^1, ..., $x_{k_1}^1$, ..., x_1^n, ..., $x_{k_n}^n$ sont des variables, alors $f(x_1^1 \ldots x_{k_1}^1 t_1, \ldots, x_1^n \ldots x_{k_n}^n t_n)$ est une expression.

La notation $f(x_1^1 \ldots x_{k_1}^1 t_1, \ldots, x_1^n \ldots x_{k_n}^n t_n)$ désigne l'arbre



Par exemple, l'expression $\int_t^u v\ dx$ désigne l'arbre



1.2.3 Les langages à plusieurs sortes d'expressions

On aura besoin, dans ce livre, d'utiliser des langages un peu plus généraux appelés *langages à plusieurs sortes d'expressions*. Si on se donne des constantes 0 et 1, un symbole binaire $+$, des symboles unaires *pair* et *impair* et un symbole binaire $\Rightarrow$, aucun de ces symboles ne liant de variables, on peut former les expressions 1, $1+1$, $pair(1+1)$ et $impair(1) \Rightarrow pair(1+1)$, mais on peut malheureusement former également les expressions $impair(pair(1))$ ou $1 \Rightarrow (1 + pair(1))$. Pour les exclure, il faut distinguer deux sortes d'expressions : les *termes*, qui expriment des entiers et les *propositions* qui expriment des faits concernant ces entiers. Ainsi, le symbole *pair* prend en argument un terme pour former une proposition et le symbole $\Rightarrow$ prend en argument deux propositions pour former une proposition.

Pour cela, on introduit un ensemble à deux éléments $\{Terme, Prop\}$ dont les éléments sont appelés des *sortes d'expressions* et on associe au symbole *pair* l'arité $(Terme, Prop)$ qui indique que dans une expression de la forme $pair(t)$, l'expression t doit être de sorte $Terme$ alors que l'expression $pair(t)$ elle-même est de sorte $Prop$.

On introduit, plus généralement, un ensemble $\mathcal{S}$ de sortes. L'arité d'un symbole f est alors une suite finie de sortes $(s_1, \ldots, s_n, s')$ qui indique que le symbole f a n arguments, que le premier est de sorte s_1, ..., le n-ième de sorte s_n et que l'expression formée est elle-même de la sorte s'.

Quand on a, de plus, des variables liées, l'arité d'un symbole f est une suite finie $((s_1^1, \ldots, s_{k_1}^1, s'^1), \ldots, (s_1^n, \ldots, s_{k_n}^n, s'^n), s'')$ qui indique que le symbole f a n arguments, que le premier est de sorte s'^1 et qu'il lie k_1 variables de sortes $s_1^1, \ldots, s_{k_1}^1, \ldots$, et que l'expression formée est elle-même de la sorte s''.

Les expressions se définissent alors ainsi.

Définition 1.11 (Expression d'un langage)

Soit $\mathcal{L}$ un langage, c'est-à-dire un ensemble de symboles, chacun muni d'une arité, et une famille d'ensembles infinis et disjoints indexée par les sortes dont les éléments sont appelés *variables*. L'ensemble des expressions de $\mathcal{L}$ est inductivement par les règles suivantes.

- Les variables de sorte s sont des expressions de sorte s.
- Si f est un symbole d'arité $((s_1^1, \ldots, s_{k_1}^1, s'^1), \ldots, (s_1^n, \ldots, s_{k_n}^n, s'^n), s'')$, $x_1^1, \ldots, x_{k_1}^1, \ldots, x_1^n, \ldots, x_{k_n}^n$ sont des variables de sortes $s_1^1, \ldots, s_{k_1}^1, \ldots$, $s_1^n, \ldots, s_{k_n}^n$ et $t_1, \ldots, t_n$ sont des expressions de sortes $s'^1, \ldots, s'^n$ alors $f(x_1^1 \ldots x_{k_1}^1 t_1, \ldots, x_1^n \ldots x_{k_n}^n t_n)$ est une expression de sorte s''.

Définition 1.12 (Variable d'une expression)

L'ensemble des *variables* d'une expression est défini par récurrence structurelle de la manière suivante

- $Var(x) = \{x\}$,
- $Var(f(x_1^1 \ldots x_{k_1}^1 t_1, \ldots, x_1^n \ldots x_{k_n}^n t_n))$
$= Var(t_1) \cup \{x_1^1, \ldots, x_{k_1}^1\} \cup \cdots \cup Var(t_n) \cup \{x_n^n, \ldots, x_{k_n}^n\}$.

Définition 1.13 (Variable libre d'une expression)

L'ensemble des *variables libres* d'une expression est défini par récurrence structurelle de la manière suivante

- $VL(x) = \{x\}$,
- $VL(f(x_1^1 \ldots x_{k_1}^1 t_1, \ldots, x_1^n \ldots x_{k_n}^n t_n))$
$= (VL(t_1) \setminus \{x_1^1, \ldots, x_{k_1}^1\}) \cup \cdots \cup (VL(t_n) \setminus \{x_n^n, \ldots, x_{k_n}^n\})$.

Par exemple, $Var(\forall x\ (x = x)) = \{x\}$, mais $VL(\forall x\ (x = x)) = \varnothing$.

Une expression sans variables libres est dite *close*.

Définition 1.14 (Hauteur d'une expression)

La *hauteur* d'une expression est définie par récurrence structurelle de la manière

suivante
- $Hauteur(x) = 0$,
- $Hauteur(f(x_1^1 \dots x_{k_1}^1 t_1, \dots, x_1^n \dots x_{k_n}^n t_n)) = 1 + max(Hauteur(t_1), \dots, Hauteur(t_n))$.

1.2.4 La substitution

La première opération que l'on est amené à définir avec la notion de variable est celle de substitution : le rôle des variables est, en effet, non seulement d'être liées, mais également d'être substituées. Par exemple, de la proposition $\forall x\ (impair(x) \Rightarrow pair(x+1))$, on veut pouvoir déduire la proposition $impair(3) \Rightarrow pair(3+1)$ obtenue en substituant la variable x par l'expression 3.

Définition 1.15 (Substitution)

Une *substitution* est une fonction de domaine fini qui à des variables $x_1, \dots, x_n$ associe des expressions de même sorte, c'est-à-dire un ensemble fini de couples dont la première composante est une variable et la seconde une expression tel que chaque variable apparaisse dans un couple au plus, ou encore une liste d'associations : $\theta = t_1/x_1, \dots, t_n/x_n$.

Quand on applique une substitution à une expression, on veut remplacer toutes les occurrences des variables $x_1, \dots, x_n$ par les expressions $t_1, \dots, t_n$.

Bien entendu, ce remplacement ne concerne que les variables libres. Par exemple, si on substitue la variable x par l'expression 2 dans l'expression $x+3$, on veut obtenir l'expression $2+3$. En revanche, si on substitue la variable x par l'expression 2 dans l'expression $\forall x\ (x = x)$, on veut obtenir l'expression $\forall x\ (x = x)$ et non l'expression $\forall x\ (2 = 2)$.

La première tentative pour définir l'application d'une substitution à une expression mène à la définition suivante.

Définition 1.16 (Application d'une substitution — avec capture)

Soit $\theta = t_1/x_1, \dots, t_n/x_n$ une substitution et t une expression. On définit l'expression $\langle\theta\rangle t$ par récurrence sur la structure de t de la manière suivante
- $\langle\theta\rangle x_i = t_i$,
- $\langle\theta\rangle x = x$ si x n'est pas dans le domaine de θ,
- $\langle\theta\rangle f(y_1^1 \dots y_{k_1}^1\ u_1, \dots, y_1^p \dots y_{k_p}^p\ u_p)$

$$= f(y_1^1 \dots y_{k_1}^1 \ \langle \theta_{|(\mathcal{V}\setminus\{y_1^1,\dots,y_{k_1}^1\})} \rangle u_1, \dots, y_1^p \dots y_{k_p}^p \ \langle \theta_{|(\mathcal{V}\setminus\{y_1^p,\dots,y_{k_p}^p\})} \rangle u_p)$$

où la notation $\theta_{|\mathcal{V}\setminus\{y_1,\dots,y_k\}}$ désigne la substitution θ restreinte à l'ensemble $\mathcal{V} \setminus \{y_1, \dots, y_k\}$, c'est-à-dire dans laquelle on a supprimé les couples dont la première composante est l'une des variables $y_1, \dots, y_k$.

Cette définition pose néanmoins un problème, car elle autorise les *captures de variables*. Par exemple, l'expression $\exists x \ (x + 1 = y)$ exprime que y est le successeur d'un certain nombre. Si on substitue y par 4 dans cette expression, on obtient l'expression $\exists x \ (x+1 = 4)$, qui exprime que 4 est le successeur d'un certain nombre. Si on substitue y par z, on obtient l'expression $\exists x \ (x+1 = z)$, qui exprime que z est le successeur d'un certain nombre. Mais si on substitue y par x, on obtient l'expression $\exists x \ (x + 1 = x)$, qui exprime qu'il existe un nombre qui est son propre successeur, et non, comme on s'y attendrait, que x est le successeur d'un certain nombre.

Pour éviter ce problème, il faut se rappeler que les variables liées sont muettes : leur nom n'importe pas. Autrement dit, dans l'expression $\exists x \ (x+1 = y)$, on peut remplacer la variable liée x par n'importe quelle autre variable, sauf, bien entendu, y. Ainsi, quand on substitue dans une expression u des variables $x_1, \dots, x_n$ par des expressions $t_1, \dots, t_n$, on peut changer le nom des variables liées dans u en prenant des noms qui n'apparaissent ni parmi $x_1, \dots, x_n$, ni parmi les variables de $t_1, \dots, t_n$, ni parmi les variables de u, afin d'éviter ces problèmes.

On commence donc par définir, en utilisant la notion de substitution avec capture définie ci-avant, une relation d'équivalence sur les expressions, par récurrence sur leur hauteur : la relation d'*équivalence alphabétique*, qui est le changement de nom des variables liées.

Définition 1.17 (Équivalence alphabétique)

La relation d'*équivalence alphabétique*, ou *alpha-équivalence*, est inductivement définie par les règles

- $x \sim x$,
- $f(y_1^1 \dots y_{k_1}^1 \ t_1, \dots, y_1^n \dots y_{k_n}^n \ t_n) \sim f(y_1'^1 \dots y_{k_1}'^1 \ t_1', \dots, y_1'^n \dots y_{k_n}'^n \ t_n')$ si pour tout i, et pour toute suite de variables distinctes $z_1, \dots, z_{k_i}$ qui n'apparaissent pas dans t_i ni dans t_i', $\langle z_1/y_1^i, \dots, z_{k_i}/y_{k_i}^i \rangle t_i \sim \langle z_1/y_1'^i, \dots, z_{k_i}/y_{k_i}'^i \rangle t_i'$.

Par exemple, les expressions $\forall x \ (x = x)$ et $\forall y \ (y = y)$ sont α-équivalentes.

Désormais on ne considère plus les expressions que *à α-équivalence près*, c'est-à-dire que l'on considère implicitement des classes d'α-équivalence d'ex-

pressions.

On peut maintenant définir l'opération de substitution par récurrence sur la hauteur des expressions.

Définition 1.18 (Application d'une substitution)

Soit $\theta = t_1/x_1, \ldots, t_n/x_n$ une substitution et t une expression. On définit l'expression θt par récurrence sur la hauteur de t de la manière suivante

- $\theta x_i = t_i$,
- $\theta x = x$ si x n'est pas dans le domaine de θ,
- $\theta f(y_1^1 \ldots y_{k_1}^1 u_1, \ldots, y_1^p \ldots y_{k_p}^p u_p) =$
  $f(z_1^1 \ldots z_{k_1}^1 \theta\langle z_1^1/y_1^1, \ldots, z_{k_1}^1/y_{k_1}^1\rangle u_1, \ldots, z_1^p \ldots z_{k_p}^p \theta\langle z_1^p/y_1^p, \ldots, z_{k_p}^p/y_{k_p}^p\rangle u_p)$
  où $z_1^1, \ldots, z_{k_1}^1, \ldots, z_1^p, \ldots, z_{k_p}^p$ sont des variables qui n'apparaissent pas dans $f(y_1^1 \ldots y_{k_1}^1\ u_1, \ldots, y_1^p \ldots y_{k_p}^p\ u_p)$ ni dans θ.

Par exemple, quand on substitue la variable y par l'expression $2 \times x$ dans l'expression $\exists x\ (x+1 = y)$, on obtient l'expression $\exists z\ (z+1 = 2 \times x)$. Le choix de la variable z est arbitraire, on aurait pu tout aussi bien choisir v ou w, ce qui aurait donné la même expression, à α-équivalence près.

Définition 1.19 (Composition de deux substitutions)

La *composition* de deux substitutions $\theta = t_1/x_1, \ldots, t_n/x_n$ et $\sigma = u_1/y_1, \ldots, u_p/y_p$ est la substitution

$$\theta \circ \sigma = \{\theta(\sigma z)/z | z \in \{x_1, \ldots, x_n, y_1, \ldots, y_p\}\}$$

On démontre, par récurrence sur la hauteur de t, que pour toute expression t

$$(\theta \circ \sigma)t = \theta(\sigma t)$$

1.2.5 L'articulation

Dans les définitions ci-avant, nous n'avons donné aucune restriction sur le nombre de symboles d'un langage. Il faut cependant prendre en compte le fait que, *in fine*, les expressions d'un langage doivent s'écrire avec un alphabet fini. Si chaque symbole du langage est exprimé par une lettre de cet alphabet, cela impose que l'ensemble des symboles du langage soit fini. Toutefois, il est également possible d'exprimer ces symboles par des mots formés sur un alphabet fini, ou plus généralement par des arbres étiquetés par les éléments d'un ensemble fini. Ainsi, en géométrie, certains symboles, comme π, sont des lettres,

mais d'autres, comme « médiatrice », des mots. Rien n'empêche d'itérer ce processus et de représenter les symboles d'un langage par des arbres étiquetés par des arbres eux-mêmes étiquetés par les éléments d'un ensemble fini, ce qui mène à la définition suivante.

Définition 1.20 (Ensemble d'arbres articulé)

- Un ensemble d'arbres est *simplement articulé*, ou 1-*articulé*, si les nœuds des arbres de cet ensemble sont étiquetés par les éléments d'un ensemble fini.
- Un ensemble d'arbre est $(n+1)$-*articulé*, si les nœuds des arbres de cet ensemble sont étiquetés par les éléments d'un ensemble d'arbres n-articulé.

Un ensemble d'arbres est *articulé* s'il est n-articulé pour un certain entier n.

Par exemple, l'ensemble des expressions sans variables d'un langage contenant un nombre fini de symboles est un ensemble d'arbres simplement articulé. En revanche, un ensemble de variables étant toujours infini, l'ensemble des expressions d'un langage est toujours un ensemble d'arbres au moins doublement articulé. Les variables $x, x', x'', x''', x'''', \ldots$ forment un ensemble infini et peuvent s'exprimer par des arbres dont les nœuds sont étiquetés par les symboles x et $'$.

Quand un langage est articulé, l'ensemble de ses symboles est fini ou dénombrable. Il est toutefois parfois nécessaire de considérer des langages qui comprennent un nombre non dénombrable de symboles et qui sont, de ce fait, non articulés. Nous verrons un exemple à la section 2.4. Il faut cependant avoir conscience que cette notion de langage généralise quelque peu la notion courante, car les expressions de ces langages ne peuvent plus s'écrire avec un alphabet fini.

Soit E un ensemble et $f_1, f_2, \ldots$ des règles sur l'ensemble E. L'ensemble des dérivations dans $f_1, f_2, \ldots$ n'est pas toujours un ensemble d'arbres articulé. Toutefois, si E est un ensemble d'arbres articulé, alors l'ensemble des dérivations dans $f_1, f_2, \ldots$ est un ensemble d'arbres articulé. De même, si chaque règle $f_1, f_2, \ldots$ est associée à un élément d'un ensemble articulé, alors l'ensemble des dérivations étiquetées par les règles $f_1, f_2, \ldots$ est un ensemble articulé.

1.3 Les langages de la logique des prédicats

La notion de langage introduite à la section précédente est très générale. Nous allons à présent nous concentrer sur le cas particulier des langages de la logique des prédicats. Dans ces langages, la plupart des symboles ne lient pas de variables. Les seules exceptions sont les quantificateurs $\forall$ et $\exists$. Par ailleurs, ces langages sont organisés autour d'une opposition entre les *termes*, qui expriment des choses, et les *propositions*, qui expriment des faits à propos de ces choses, les termes pouvant eux-mêmes avoir plusieurs sortes. Ainsi, un langage est défini par un ensemble non vide $\mathcal{S}$, dont les éléments sont appelés *sortes de termes*, un ensemble $\mathcal{F}$, dont les éléments sont appelés *symboles de fonction* et permettent de former des termes à partir d'autres termes, et un ensemble $\mathcal{P}$, dont les éléments sont appelés *symboles de prédicat* et permettent de former des propositions à partir de termes.

Les sortes du langage sont les sortes de termes plus une sorte $Prop$ pour les propositions. Comme les symboles de fonction ne lient pas de variables, leur arité est de la forme $(s_1, \ldots, s_n, s')$ où $s_1, \ldots, s_n$ et s' sont des sortes de termes. Si un symbole f a une telle arité et si $t_1, \ldots, t_n$ sont des termes de sorte $s_1, \ldots, s_n$, alors l'expression $f(t_1, \ldots, t_n)$ est un terme de sorte s'. De même, comme les symboles de prédicat ne lient pas de variables, leur arité est de la forme $(s_1, \ldots, s_n, Prop)$, où $s_1, \ldots, s_n$ sont des sortes de termes. Cette arité est plus simplement notée $(s_1, \ldots, s_n)$. Si un symbole P a une telle arité et si $t_1, \ldots, t_n$ sont des termes de sorte $s_1, \ldots, s_n$, alors l'expression $P(t_1, \ldots, t_n)$ est une proposition. À ces symboles, qui varient d'un langage à l'autre, s'ajoutent des symboles communs à tous les langages de la logique des prédicats : $\top$, *vrai*, et $\bot$, *faux*, d'arité $(Prop)$, $\neg$, *non*, d'arité $(Prop, Prop)$, $\wedge$, *et*, $\vee$, *ou*, et $\Rightarrow$, *implique*, d'arité $(Prop, Prop, Prop)$ et enfin, pour chaque élément de $\mathcal{S}$, deux quantificateurs $\forall_s$, *pour tout*, et $\exists_s$, *il existe*, d'arité $((s, Prop), Prop)$. Comme il n'y a pas de symbole permettant de lier une variable de sorte $Prop$, il n'est pas nécessaire d'introduire de telles variables.

Cela mène aux définitions suivantes.

Définition 1.21 (Langage de la logique des prédicats)

Un *langage* $\mathcal{L}$ est un triplet $(\mathcal{S}, \mathcal{F}, \mathcal{P})$ où $\mathcal{S}$ est un ensemble non vide dont les éléments sont appelés *sortes de termes* et $\mathcal{F}$ et $\mathcal{P}$ des ensembles dont les éléments sont respectivement appelés *symboles de fonction* et *symboles de prédicat*. À chaque symbole de fonction, on associe une arité qui est un $(n+1)$-uplet d'éléments de $\mathcal{S}$ et à chaque symbole de prédicat une arité qui est un n-uplet d'éléments de $\mathcal{S}$.

Définition 1.22 (Terme)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage et $(\mathcal{V}_s)_{s\in\mathcal{S}}$ une famille d'ensembles infinis et disjoints indexée par les sortes de termes dont les éléments sont appelés *variables*. Les *termes* de sorte s du langage $\mathcal{L}$, pour la famille d'ensembles de variables $(\mathcal{V}_s)_{s\in\mathcal{S}}$, sont inductivement définis par les règles suivantes.

- Les variables de sorte s sont des termes de sorte s.
- Si f est un symbole d'arité $(s_1, \ldots, s_n, s')$ et t_1, …, t_n des termes de sortes s_1, …, s_n, alors $f(t_1, \ldots, t_n)$ est un terme de sorte s'.

Définition 1.23 (Proposition)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage et $(\mathcal{V}_s)_{s\in\mathcal{S}}$ une famille d'ensembles infinis et disjoints indexée par les sortes de termes dont les éléments sont appelés *variables*. Les *propositions* du langage $\mathcal{L}$, pour la famille d'ensembles de variables $(\mathcal{V}_s)_{s\in\mathcal{S}}$, sont inductivement définies par les règles suivantes.

- Si P est un symbole de prédicat d'arité $(s_1, \ldots, s_n)$ et $t_1, \ldots, t_n$ sont des termes de sorte s_1, …, s_n, alors l'expression $P(t_1, \ldots, t_n)$ est une proposition.
- $\top$ et $\bot$ sont des propositions.
- Si A est une proposition, alors $\neg A$ est une proposition.
- Si A et B sont des propositions, alors $A \wedge B$, $A \vee B$ et $A \Rightarrow B$ sont des propositions.
- Si A est une proposition et x une variable de sorte s, alors $\forall_s x\ A$ et $\exists_s x\ A$ sont des propositions.

On utilise la notation $A \Leftrightarrow B$ pour la proposition $(A \Rightarrow B) \wedge (B \Rightarrow A)$. Une proposition de la forme $P(t_1, \ldots, t_n)$ est appelée une *proposition atomique*.

Quand $\mathcal{S}$ est un singleton, on dit que le langage a une seule sorte de termes, et l'arité d'un symbole de fonction ou de prédicat se réduit alors à un entier : le nombre d'arguments de ce symbole.

Exercice 1.2

Soit $\mathcal{L}$ le langage à une sorte de termes formé des symboles $\mathbb{C}$, $\mathbb{N}$, 0, $=$, $\hat{}$, $\in$ et $\#$ où le symbole $\hat{}$ est la puissance et $\#$ le cardinal.

1. Écrire la proposition

   Tout nombre complexe non nul a n racines n-ièmes

   comme une proposition du langage $\mathcal{L}$.

2. Quels symboles sont des symboles de fonction, quels symboles sont des symboles de prédicat ?
3. Quelle est l'arité de chaque symbole ?

1.4 Les démonstrations

Nous voulons maintenant distinguer une proposition, comme $\exists x\ (x = 0+1)$, qui est démontrable, d'une proposition qui ne l'est pas, comme $\exists x\ (0 = x+1)$.

Une manière de le faire est de se donner un ensemble de règles et de définir inductivement, à l'aide de ces règles, un sous-ensemble de l'ensemble des propositions : l'ensemble des *théorèmes* ou *propositions démontrables*.

Exercice 1.3

Soit le langage à une sorte de termes formé des symboles de fonction 0, d'arité nulle, et S, *successeur*, d'arité 1, et du symbole de prédicat $\leq$, d'arité 2. On se donne les règles suivantes

$$\frac{\forall x\ A}{(t/x)A}$$

$$\frac{A \Rightarrow B\ A}{B}$$

$$\frac{A\ B}{A \wedge B}$$

$$\frac{}{\forall x \forall y \forall z\ ((x \leq y \wedge y \leq z) \Rightarrow x \leq z)}$$

$$\frac{}{\forall x\ (x \leq S(x))}$$

montrer que la proposition

$$0 \leq S(S(0))$$

est démontrable.

Une telle définition de la notion de démonstration est possible, on parle alors de *démonstration à la Frege et Hilbert*, mais elle est difficile à utiliser. En effet, en posant ainsi des règles qui permettent de démontrer des propositions, on se contraint à garder les mêmes hypothèses tout au long de la démonstration. On ne peut donc pas traduire une forme de raisonnement pourtant courante : nous voulons démontrer $A \Rightarrow B$, supposons A et démontrons B sous cette hypothèse. Cette remarque mène à introduire une notion de couple formé d'un ensemble fini d'hypothèses et d'une conclusion. Un tel couple est appelé un *séquent*.

Définition 1.24 (Séquent)

Un *séquent* est un couple $\Gamma \vdash A$, où Γ est un ensemble fini de propositions et A une proposition.

Définition 1.25 (Les règles de la déduction naturelle)

$$\frac{}{\Gamma \vdash A}\ \text{axiome } A \in \Gamma$$

$$\frac{}{\Gamma \vdash \top}\ \top\text{-intro}$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \bot\text{-élim}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\text{-intro}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge\text{-élim}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge\text{-élim}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{-intro}$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\text{-intro}$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \vee\text{-élim}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \Rightarrow\text{-intro}$$

$$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \Rightarrow\text{-élim}$$

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg\text{-intro}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash \neg A}{\Gamma \vdash \bot}\ \neg\text{-élim}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x\ A}\ \forall\text{-intro } x \text{ non libre dans } \Gamma$$

$$\frac{\Gamma \vdash \forall x\ A}{\Gamma \vdash (t/x)A}\ \forall\text{-élim}$$

$$\frac{\Gamma \vdash (t/x)A}{\Gamma \vdash \exists x\ A}\ \exists\text{-intro}$$

$$\frac{\Gamma \vdash \exists x\ A \quad \Gamma, A \vdash B}{\Gamma \vdash B}\ \exists\text{-élim } x \text{ non libre dans } \Gamma, B$$

$$\frac{}{\Gamma \vdash A \vee \neg A}\ \text{tiers exclu}$$

Les règles $\top$-intro, $\wedge$-intro, $\vee$-intro, $\Rightarrow$-intro, $\neg$-intro, $\forall$-intro et $\exists$-intro sont appelées des *règles d'introduction* et les règles $\bot$-élim, $\wedge$-élim, $\vee$-élim, $\Rightarrow$-élim, $\neg$-élim, $\forall$-élim et $\exists$-élim des *règles d'élimination*. Les règles de la déduction naturelle sont donc classées en quatre groupes : les règles d'introduction, les règles d'élimination, la règle *axiome* et la règle *tiers exclu*.

Définition 1.26 (Séquent démontrable)

L'*ensemble des séquents démontrables* est inductivement défini par les règles de la déduction naturelle.

Définition 1.27 (Démonstration)

Une *démonstration* d'un séquent $\Gamma \vdash A$ est une dérivation de ce séquent, c'est-à-dire un arbre dont les nœuds sont étiquetés par des séquents dont la racine est étiquetée par $\Gamma \vdash A$, et tel que si un nœud est étiqueté par un séquent $\Delta \vdash B$, alors ses enfants sont étiquetés par des séquents $\Sigma_1 \vdash C_1, \ldots, \Sigma_n \vdash C_n$ tels qu'il existe une règle de déduction naturelle, qui permet de déduire $\Delta \vdash B$ de $\Sigma_1 \vdash C_1, \ldots, \Sigma_n \vdash C_n$.

Un séquent $\Gamma \vdash A$ est donc démontrable s'il existe une démonstration de ce séquent.

Exercice 1.4

On considère un langage à trois sortes de termes : *point*, *droite* et *scalaire* formé de deux symboles de prédicat $=$ d'arité (*scalaire*, *scalaire*) et $\in$ d'arité (*point*, *droite*) et de deux symboles de fonction d, *distance*, d'arité (*point*, *point*, *scalaire*) et m, *médiatrice*, d'arité (*point*, *point*, *droite*). Soient Γ l'ensemble contenant les propositions

$$\forall x \forall y \forall z\ (x \in m(y,z) \Leftrightarrow d(x,y) = d(x,z))$$

et

$$\forall x \forall y \forall z\ ((x = y \wedge y = z) \Rightarrow x = z)$$

et A la proposition qui exprime que si deux médiatrices du triangle xyz sont concourantes, alors ses trois médiatrices le sont

$$\forall w \forall x \forall y \forall z \ ((w \in m(x,y) \wedge w \in m(y,z)) \Rightarrow w \in m(x,z))$$

Donner une démonstration du séquent $\Gamma \vdash A$.

La proposition suivante montre que, dans un séquent, on peut ajouter des hypothèses inutiles.

Proposition 1.6 (L'affaiblissement)

Si le séquent $\Gamma \vdash A$ est démontrable, alors le séquent $\Gamma, B \vdash A$ est démontrable.

Démonstration. Par récurrence sur la structure d'une démonstration de $\Gamma \vdash A$.

Proposition 1.7 (La double négation)

Les trois propositions sont équivalentes.

1. Le séquent $\Gamma \vdash A$ est démontrable.
2. Le séquent $\Gamma, \neg A \vdash \bot$ est démontrable.
3. Le séquent $\Gamma \vdash \neg\neg A$ est démontrable.

Démonstration.

(1.) $\Rightarrow$ (2.) Si le séquent $\Gamma \vdash A$ est démontrable, alors, d'après la proposition 1.6, le séquent $\Gamma, \neg A \vdash A$ également. Le séquent $\Gamma, \neg A \vdash \neg A$ est démontrable avec la règle *axiome* et donc le séquent $\Gamma, \neg A \vdash \bot$ avec la règle $\neg$-élim.

(2.) $\Rightarrow$ (3.) Si le séquent $\Gamma, \neg A \vdash \bot$ est démontrable, alors le séquent $\Gamma \vdash \neg\neg A$ est démontrable avec la règle $\neg$-intro.

(3.) $\Rightarrow$ (2.) Si le séquent $\Gamma \vdash \neg\neg A$ est démontrable, alors, d'après la proposition 1.6, le séquent $\Gamma, \neg A \vdash \neg\neg A$ également. Le séquent $\Gamma, \neg A \vdash \neg A$ est démontrable avec la règle *axiome* et donc le séquent $\Gamma, \neg A \vdash \bot$ avec la règle $\neg$-élim.

(2.) $\Rightarrow$ (1.) Si le séquent $\Gamma, \neg A \vdash \bot$ a une démonstration π, alors le séquent $\Gamma \vdash A$ a la démonstration

$$\dfrac{\dfrac{}{\Gamma \vdash A \vee \neg A}\ \text{tiers exclu} \quad \dfrac{}{\Gamma, A \vdash A}\ \text{axiome} \quad \dfrac{\dfrac{\pi}{\Gamma, \neg A \vdash \bot}}{\Gamma, \neg A \vdash A}\ \bot\text{-élim}}{\Gamma \vdash A}\ \vee\text{-élim}$$

Proposition 1.8

Le séquent $\vdash \neg\exists x \neg A \Rightarrow \forall x A$ est démontrable.

Démonstration. Ce séquent a la démonstration

$$\dfrac{\dfrac{\dfrac{}{\vdash \neg\exists x\neg A, A \vee \neg A}\,\text{tiers exclu} \quad \dfrac{\dfrac{\dfrac{}{\neg\exists x\neg A, \neg A \vdash \neg A}\,\text{axiome}}{\neg\exists x\neg A, \neg A \vdash \exists x\ \neg A}\,\exists\text{-intro}}{\neg\exists x\neg A, \neg A \vdash \neg\exists x\ \neg A}\,\neg\text{-élim} \quad \dfrac{}{\neg\exists x\neg A, A \vdash A}\,\text{axiome}}{\dfrac{\neg\exists x\neg A, \neg A \vdash \bot}{\neg\exists x\neg A, \neg A \vdash A}\,\bot\text{-élim}}\,\vee\text{-élim}}{\dfrac{\neg\exists x\neg A \vdash A}{\neg\exists x\neg A \vdash \forall x\ A}\,\forall\text{-intro}}{\vdash \neg\exists x\neg A \Rightarrow \forall x\ A}\,\Rightarrow\text{-intro}$$

Définition 1.28 (Théorie)

Une *théorie* est un ensemble fini ou infini de propositions closes dont les éléments sont appelés *axiomes*.

Quand une théorie $\mathcal{T}$ est finie, on dit qu'une proposition A est un *théorème de cette théorie*, ou encore qu'elle est *démontrable dans cette théorie*, si le séquent $\mathcal{T} \vdash A$ est démontrable. Cependant, dans le cas général, le couple $\mathcal{T} \vdash A$ n'est pas un séquent. On doit donc donner une définition un peu plus générale.

Définition 1.29 (Théorème)

Une proposition A est un *théorème de la théorie* $\mathcal{T}$, ou une *proposition démontrable dans cette théorie*, s'il existe un sous-ensemble fini Γ de $\mathcal{T}$, tel que le séquent $\Gamma \vdash A$ soit démontrable.

Définition 1.30 (Cohérence, contradiction)

Une théorie $\mathcal{T}$ est *cohérente* s'il existe une proposition qui n'est pas démontrable dans $\mathcal{T}$. Elle est dite *contradictoire* sinon.

Proposition 1.9

Une théorie est contradictoire si et seulement si elle démontre la proposition $\bot$.

Démonstration. Si la théorie est contradictoire, elle démontre toutes les propositions, donc, en particulier, la proposition $\bot$. Réciproquement, si une théorie démontre la proposition $\bot$, alors il existe un sous-ensemble fini Γ de $\mathcal{T}$ tel que le séquent $\Gamma \vdash \bot$ ait une démonstration π. Soit A une proposition quelconque. Le séquent $\Gamma \vdash A$ a la démonstration

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \bot}}{\Gamma \vdash A}\,\bot\text{-élim}$$

et la proposition A est donc démontrable dans la théorie $\mathcal{T}$.

Proposition 1.10

Une théorie $\mathcal{T}$ est contradictoire si et seulement s'il existe une proposition A telle que la théorie démontre A et $\neg A$.

Démonstration. Si la théorie est contradictoire elle démontre toutes les propositions, donc, en particulier, les propositions $\top$ et $\neg\top$.

Réciproquement, si une théorie démontre les propositions A et $\neg A$, alors il existe deux sous-ensembles finis Γ et Γ' tels que les séquents $\Gamma \vdash A$ et $\Gamma' \vdash \neg A$ soient démontrables. D'après la proposition 1.6, les séquents $\Gamma, \Gamma' \vdash A$ et $\Gamma, \Gamma' \vdash \neg A$ ont des démonstrations π_1 et π_2. Le séquent $\Gamma, \Gamma' \vdash \bot$ a alors la démonstration

$$\dfrac{\dfrac{\pi_2}{\Gamma, \Gamma' \vdash \neg A} \quad \dfrac{\pi_1}{\Gamma, \Gamma' \vdash A}}{\Gamma, \Gamma' \vdash \bot}\,\neg\text{-élim}$$

La proposition $\bot$ est donc démontrable dans la théorie $\mathcal{T}$ et, d'après la proposition 1.9, la théorie $\mathcal{T}$ est contradictoire.

Exercice 1.5

Montrer que si le séquent $\Gamma \vdash A \Leftrightarrow A'$ est démontrable, alors les séquents $\Gamma \vdash (A \wedge B) \Leftrightarrow (A' \wedge B)$, $\Gamma \vdash (B \wedge A) \Leftrightarrow (B \wedge A')$, $\Gamma \vdash (A \vee B) \Leftrightarrow (A' \vee B)$, $\Gamma \vdash (B \vee A) \Leftrightarrow (B \vee A')$, $\Gamma \vdash (A \Rightarrow B) \Leftrightarrow (A' \Rightarrow B)$, $\Gamma \vdash (B \Rightarrow A) \Leftrightarrow (B \Rightarrow A')$, $\Gamma \vdash (\neg A) \Leftrightarrow (\neg A')$, $\Gamma \vdash (\forall x\ A) \Leftrightarrow (\forall x\ A')$ et $\Gamma \vdash (\exists x\ A) \Leftrightarrow (\exists x\ A')$ sont démontrables.

Exercice 1.6

Une théorie à plusieurs sortes de termes peut se *relativiser* en une théorie à une seule sorte de termes. À chaque symbole de fonction f d'arité $(s_1, \ldots, s_n, s')$,

on associe un symbole de fonction f' d'arité n et à chaque symbole de prédicat P d'arité $(s_1, \ldots, s_n)$, on associe un symbole de prédicat P' d'arité n. Pour chaque sorte s, on introduit un symbole de prédicat unaire S_s. On traduit alors les termes et les propositions de la manière suivante

- $|x| = x$,
- $|f(t_1, \ldots, t_n)| = f'(|t_1|, \ldots, |t_n|)$,
- $|P(t_1, \ldots, t_n)| = P'(|t_1|, \ldots, |t_n|)$,
- $|\top| = \top$,
- $|\bot| = \bot$,
- $|\neg A| = \neg|A|$,
- $|A \wedge B| = |A| \wedge |B|$, $|A \vee B| = |A| \vee |B|$, $|A \Rightarrow B| = |A| \Rightarrow |B|$,
- $|\forall_s x\ A| = \forall x\ (S_s(x) \Rightarrow |A|)$, $|\exists_s x\ A| = \exists x\ (S_s(x) \wedge |A|)$.

On traduit une théorie en traduisant chaque axiome et en ajoutant pour chaque sorte s l'axiome

$$\exists x\ S_s(x)$$

et pour chaque symbole de fonction f d'arité $(s_1, \ldots, s_n, s')$ l'axiome

$$\forall x_1 \ldots \forall x_n\ ((S_{s_1}(x_1) \wedge \ldots \wedge S_{s_n}(x_n)) \Rightarrow (S_{s'}(f'(x_1, \ldots, x_n))))$$

Soit $\mathcal{T}'$ la théorie formée pour chaque variable de sorte s de l'axiome $S_s(x)$. Montrer que si le terme t a la sorte s, alors la proposition $S_s(|t|)$ est démontrable dans la théorie $|\mathcal{T}|, \mathcal{T}'$.

Montrer que si la proposition A est démontrable dans la théorie $\mathcal{T}$, alors la proposition $|A|$ est démontrable dans la théorie $|\mathcal{T}|, \mathcal{T}'$.

Montrer que si la proposition close A est démontrable dans la théorie $\mathcal{T}$, alors la proposition $|A|$ est démontrable dans la théorie $|\mathcal{T}|$.

1.5 Des exemples de théories

Définition 1.31 (Les axiomes de l'égalité)

Soit un langage contenant des prédicats $=_s$ de sorte (s, s) pour certaines sortes s. Les *axiomes de l'égalité* pour ce langage sont les suivants. Pour chaque sorte s ayant un symbole d'égalité, l'axiome d'identité

$$\forall_s x\ (x =_s x)$$

Pour chaque symbole de fonction f d'arité $(s_1, \ldots, s_n, s')$ telle que la sorte s' ait un symbole d'égalité et chaque entier i tel que la sorte s_i ait un symbole d'égalité de l'axiome

$$\forall x_1 \ldots \forall x_i \forall x_i' \ldots \forall x_n\ (x_i =_{s_i} x_i' \Rightarrow f(x_1, \ldots, x_i, \ldots, x_n) =_{s'} f(x_1, \ldots, x_i', \ldots, x_n))$$

Pour chaque symbole de prédicat P d'arité $(s_1, \ldots, s_n)$ et chaque entier i tel que la sorte s_i ait un symbole d'égalité de l'axiome

$$\forall x_1 \ldots \forall x_i \forall x_i' \forall x_n \; (x_i =_{s_i} x_i' \Rightarrow (P(x_1, \ldots, x_i, \ldots, x_n) \Rightarrow P(x_1, \ldots, x_i', \ldots, x_n)))$$

Exercice 1.7

Donner une démonstration, dans la théorie de l'égalité, des propositions

$$\forall_s x \forall_s y \forall_s z \; (x =_s y \Rightarrow (y =_s z \Rightarrow x =_s z))$$

$$\forall_s x \forall_s y \; (x =_s y \Rightarrow y =_s x)$$

Définition 1.32 (La théorie des classes)

Soit un langage à deux sortes de termes : ι pour les objets et κ pour les classes d'objets, contenant un nombre arbitraire de symboles de fonction d'arité $(\iota, \ldots, \iota, \iota)$ et de symboles de prédicat d'arité $(\iota, \ldots, \iota)$, ainsi qu'un symbole de prédicat ϵ d'arité (ι, κ).

La *théorie des classes* pour ce langage contient, pour chaque proposition A ne contenant pas le symbole ϵ et dont les variables libres sont parmi $x_1, \ldots, x_n, y$, l'axiome

$$\forall x_1 \ldots \forall x_n \exists c \forall y \; (y \; \epsilon \; c \Leftrightarrow A)$$

Cet ensemble d'axiomes s'appelle le *schéma de compréhension*.

Définition 1.33 (L'arithmétique)

Le langage de l'arithmétique contient deux sortes de termes ι et κ, une constante 0 de sorte ι, des symboles de fonction S, *successeur*, d'arité (ι, ι), $+$ et $\times$ d'arité (ι, ι, ι) et des symboles de prédicat ϵ d'arité (ι, κ) et $=$ d'arité (ι, ι). Aux axiomes de l'égalité et au schéma de compréhension, on ajoute les axiomes du successeur

$$\forall x \forall y \; (S(x) = S(y) \Rightarrow x = y)$$

$$\forall x \; \neg(0 = S(x))$$

l'axiome de récurrence

$$\forall c \; (0 \; \epsilon \; c \Rightarrow \forall x \; (x \; \epsilon \; c \Rightarrow S(x) \; \epsilon \; c) \Rightarrow \forall y \; y \; \epsilon \; c)$$

et les axiomes de l'addition et de la multiplication

$$\forall y \; (0 + y = y)$$

$$\forall x \forall y\ (S(x) + y = S(x + y))$$
$$\forall y\ (0 \times y = 0)$$
$$\forall x \forall y\ (S(x) \times y = (x \times y) + y)$$

Exercice 1.8 (Le schéma de récurrence)

Cet exercice demande d'avoir fait l'exercice 1.5.

Montrer que pour chaque proposition A de l'arithmétique, ne contenant pas le symbole ϵ et dont les variables libres sont parmi $x_1, \ldots, x_n, y$, la proposition

$$\forall x_1 \ldots \forall x_n\ ((0/y)A \Rightarrow \forall m\ ((m/y)A \Rightarrow (S(m)/y)A) \Rightarrow \forall n\ (n/y)A)$$

est démontrable dans l'arithmétique.

Définition 1.34 (La théorie naïve des ensembles)

Le langage de la théorie naïve des ensembles contient une sorte et un symbole de prédicat binaire $\in$. Elle contient pour chaque proposition A dont les variables libres sont parmi $x_1, \ldots, x_n, y$, un axiome de la forme

$$\forall x_1 \ldots \forall x_n \exists a \forall y\ (y \in a \Leftrightarrow A)$$

Exercice 1.9 (Le paradoxe de Russell)

Montrer que le séquent

$$\forall y\ (y \in a \Leftrightarrow \neg y \in y) \vdash \bot$$

est démontrable. En déduire que la théorie naïve des ensembles est contradictoire. Pourquoi ce paradoxe ne s'applique-t-il pas à la théorie des classes ?

Définition 1.35 (La théorie des classes binaires)

Soit un langage à deux sortes de termes ι pour les objets et σ pour les classes binaires, contenant un nombre arbitraire de symboles de fonction d'arité $(\iota, \ldots, \iota, \iota)$ et de symboles de prédicat d'arité $(\iota, \ldots, \iota)$ ainsi qu'un symbole de prédicat ϵ_2 d'arité (ι, ι, σ).

La théorie des classes binaires pour ce langage contient, pour chaque proposition A ne contenant pas le symbole ϵ_2 et dont les variables libres sont parmi $x_1, \ldots, x_n, y, z$ un axiome de la forme

$$\forall x_1 \ldots \forall x_n \exists r \forall y \forall z\ (y, z\ \epsilon_2\ r \Leftrightarrow A)$$

Cet ensemble d'axiomes s'appelle le *schéma de compréhension binaire*.

Définition 1.36 (ZF : La théorie des ensembles de Zermelo-Fraenkel)

Le langage de la théorie des ensembles de Zermelo-Fraenkel contient deux sortes de termes ι et σ, un symbole de prédicat ϵ_2 d'arité (ι, ι, σ), un symbole de prédicat $=$ d'arité (ι, ι) et un symbole de prédicat $\in$ d'arité (ι, ι) pour l'appartenance d'un ensemble à un autre. Outre les axiomes de l'égalité et le schéma de compréhension binaire, la théorie des ensembles de Zermelo-Fraenkel contient les axiomes suivants.
L'*axiome d'extensionalité*, qui énonce que deux ensembles sont égaux quand ils ont les mêmes éléments

$$\forall x \forall y \ ((\forall z \ (z \in x \Leftrightarrow z \in y)) \Rightarrow x = y)$$

L'*axiome de la réunion*, qui énonce que, quand on a construit un ensemble x qui contient des éléments $v_0, v_1, \ldots$, on peut construire la réunion des ensembles $v_0, v_1, \ldots$

$$\forall x \exists z \forall w \ (w \in z \Leftrightarrow (\exists v \ (w \in v \wedge v \in x)))$$

L'*axiome des parties*, qui énonce que quand on a construit un ensemble x on peut construire un ensemble qui contient les parties de x

$$\forall x \exists z \forall w \ (w \in z \Leftrightarrow (\forall v \ (v \in w \Rightarrow v \in x)))$$

L'*axiome de l'infini*, qui énonce que l'on peut construire un ensemble infini. Soit *Vide* la proposition $\forall y \ (\neg(y \in x))$. On écrit $\mathit{Vide}[t]$ la proposition $(t/x)\mathit{Vide}$. Soit *Succ* la proposition $\forall z \ (z \in y \Leftrightarrow (z \in x \vee z = x))$. On écrit $\mathit{Succ}[t, u]$ la proposition $(t/x, u/y)\mathit{Succ}$. Intuitivement, cela signifie que u est l'ensemble $t \cup \{t\}$. L'axiome de l'infini est

$$\exists I \ (\forall x \ (\mathit{Vide}[x] \Rightarrow x \in I) \wedge \forall x \forall y \ ((x \in I \wedge \mathit{Succ}[x, y]) \Rightarrow y \in I))$$

L'*axiome de remplacement* qui énonce que quand on a construit un ensemble a et une classe binaire fonctionnelle r, on peut construire l'ensemble des objets reliés à un élément de a par la classe binaire r. Soit *fonctionnelle* la proposition $\forall y \forall z \forall z' \ ((y, z \ \epsilon_2 \ r \wedge y, z' \ \epsilon_2 \ r) \Rightarrow z = z')$. On écrit $\mathit{fonctionnelle}[t]$ la proposition $(t/r)\mathit{fonctionnelle}$. L'axiome de remplacement est

$$\forall r \ (\mathit{fonctionnelle}[r] \Rightarrow \forall a \exists b \forall z \ (z \in b \Leftrightarrow \exists y \ (y \in a \wedge y, z \ \epsilon_2 \ r)))$$

Exercice 1.10 (Le schéma de remplacement)

Cet exercice demande d'avoir fait l'exercice 1.5.

Soit A une proposition ne contenant pas le symbole ϵ_2 et dont les variables libres sont parmi $x_1, \ldots, x_n, y, z$. On écrit $A[t, u]$ la proposition $(t/y, u/z)A$. Montrer que la proposition

$$\forall x_1 \ldots \forall x_m \ ((\forall y \forall z \forall z' \ ((A[y,z] \wedge A[y,z']) \Rightarrow z = z')) \Rightarrow \\ \forall a \exists b \forall z \ (z \in b \Leftrightarrow \exists y \ (y \in a \wedge A[y,z])))$$

est démontrable dans ZF.

Exercice 1.11 (Le schéma de séparation)

Cet exercice demande d'avoir fait les exercices 1.5 et 1.10.

Soit A une proposition ne contenant pas le symbole ϵ_2 et dont les variables libres sont parmi $x_1, \ldots, x_n, y$. On écrit $A[t]$ la proposition $(t/y)A$. Montrer que la proposition

$$\forall x_1 \ldots \forall x_n \forall a \exists b \forall y \ (y \in b \Leftrightarrow (y \in a \wedge A[y]))$$

est démontrable dans ZF.

Exercice 1.12 (Le théorème de l'ensemble vide)

Cet exercice demande d'avoir fait l'exercice 1.11.

Montrer que la proposition

$$\exists b \ \mathit{Vide}[b]$$

est démontrable dans ZF.

Exercice 1.13 (Le théorème de la paire)

Cet exercice demande d'avoir fait l'exercice 1.10.

Soit Un la proposition $\forall y \ (y \in x \Leftrightarrow \mathit{Vide}[y])$. On écrit $\mathit{Un}[t]$ la proposition $(t/x)\mathit{Un}$. Intuitivement, cela signifie que $t = \{\varnothing\}$. Soit Deux la proposition $\forall y \ (y \in x \Leftrightarrow (\mathit{Vide}[y] \vee \mathit{Un}[y]))$. On écrit $\mathit{Deux}[t]$ la proposition $(t/x)\mathit{Deux}$. Intuitivement, cela signifie que $t = \{\varnothing, \{\varnothing\}\}$.

Montrer que les propositions $\exists x \ \mathit{Vide}[x]$, $\exists x \ \mathit{Un}[x]$, $\exists x \ \mathit{Deux}[x]$ et $\forall x \neg(\mathit{Vide}[x] \wedge \mathit{Un}[x])$ sont démontrables dans ZF.

Montrer que la proposition

$$\forall x \forall y \exists z \forall w \ (w \in z \Leftrightarrow (w = x \vee w = y))$$

est démontrable dans ZF.

Exercice 1.14 (Les couples)

Cet exercice demande d'avoir fait l'exercice 1.13.

En théorie des ensembles, le couple (a, b) est l'ensemble qui contient les éléments $\{a\}$ et $\{a, b\}$. Écrire une proposition utilisant uniquement les symboles $=$ et $\in$ qui exprime que le couple formé des éléments x et y est égal à z. Écrire une proposition qui exprime que le couple formé des éléments x et y est un élément de z.

Exercice 1.15 (La réunion de deux ensembles)

Cet exercice demande d'avoir fait l'exercice 1.13.

Montrer que la proposition

$$\forall x \forall y \exists z \forall w \ (w \in z \Leftrightarrow (w \in x \vee w \in y))$$

est démontrable dans ZF.

Exercice 1.16

Cet exercice demande d'avoir fait les exercices 1.12 et 1.15.

Montrer que les propositions suivantes sont démontrables.

$$\exists x \ \mathit{Vide}[x]$$

$$\forall x \exists y \ \mathit{Succ}[x, y]$$

$$\forall x \forall y \ ((\mathit{Vide}[x] \wedge \mathit{Vide}[y]) \Rightarrow x = y)$$

$$\forall x \forall y \forall y' \ ((\mathit{Succ}[x, y] \wedge \mathit{Succ}[x, y']) \Rightarrow y = y')$$

$$\forall x \forall y \ \neg(\mathit{Succ}[x, y] \wedge \mathit{Vide}[y])$$

Exercice 1.17 (Les entiers de Von Neumann)

Cet exercice demande d'avoir fait les exercices 1.11 et 1.16.

En théorie des ensembles, les entiers sont les ensembles suivants $0 = \varnothing, 1 = \{0\}, 2 = \{0, 1\}, 3 = \{0, 1, 2\}, \ldots$ Un ensemble est donc un entier s'il appartient à tous les ensembles qui contiennent 0 et qui sont clos par successeur. Écrire une proposition N contenant une variable libre x et utilisant uniquement les symboles $=$ et $\in$ qui exprime que x est un entier. On écrit $N[t]$ la proposition $(t/x)N$. On peut remarquer que tous les entiers appartiennent à l'ensemble I dont l'axiome de l'infini énonce l'existence. Démontrer la proposition

$$\exists \mathbb{N} \ (x \in \mathbb{N} \Leftrightarrow N[x])$$

Écrire une proposition qui exprime le principe de récurrence : si un ensemble contient 0 et est clos par successeur, alors il contient tous les entiers. Montrer que cette proposition est démontrable dans ZF.

Exercice 1.18

Cet exercice demande d'avoir fait les exercices 1.11, 1.13, 1.14, 1.16 et 1.17.

Soit *succ* la classe binaire fonctionnelle définie par compréhension par $x, y\ \epsilon_2\ succ \Leftrightarrow Succ[x, y]$. L'axiome de l'infini exprime l'existence d'un ensemble qui contient 0 et qui est clos par la classe binaire fonctionnelle *succ*. On veut montrer, dans cet exercice, qu'une conséquence de cet axiome est que, si a est un ensemble quelconque et r une classe binaire fonctionnelle quelconque, alors il existe un ensemble qui contient a et qui est clos par r.

On veut donc montrer la proposition

$$\forall a \forall r\ (fonctionnelle[r] \Rightarrow \exists E\ (a \in E \wedge \forall y \forall y'\ ((y \in E \wedge y, y'\ \epsilon_2\ r) \Rightarrow y' \in E)))$$

Pour cela, on pose l'hypothèse $fonctionnelle[r]$ et on cherche à montrer la proposition $\exists E\ (a \in E \wedge \forall y \forall y'\ ((y \in E \wedge y, y'\ \epsilon_2\ r) \Rightarrow y' \in E)))$.

Soit A la proposition

$n \in \mathbb{N} \wedge \forall g\ (((\forall p\ (Vide[p] \Rightarrow (p, a) \in g))$
$\wedge \forall p \forall p' \forall y \forall y'\ ((p \in n \wedge (p, y) \in g \wedge Succ[p, p'] \wedge y, y'\ \epsilon_2\ r) \Rightarrow (p', y') \in g))$
$\Rightarrow (n, x) \in g)$

On écrit $A[t, u]$ la proposition $(t/n, u/x)A$

1. Démontrer les propositions

$$\forall n\ (Vide[n] \Rightarrow A[n, a])$$

$$\forall n \forall n' \forall x \forall x'\ ((A[n, x] \wedge Succ[n, n'] \wedge x, x'\ \epsilon_2\ r) \Rightarrow A[n', x'])$$

2. On veut maintenant démontrer la proposition

$$\forall n \forall x \forall y\ ((A[n, x] \wedge A[n, y]) \Rightarrow x = y)$$

Dans un premier temps, on suppose

$$\forall p \forall x \forall y\ ((p \in n \wedge A[p, x] \wedge A[p, y]) \Rightarrow x = y)$$

et on démontre

$$\forall x \forall y\ ((A[n, x] \wedge A[n, y]) \Rightarrow x = y)$$

Soit r' la classe binaire définie en compréhension par
$p, c\ \epsilon_2\ r' \Leftrightarrow \exists y\ (c = (p, y)$
$\wedge ((Vide[p] \wedge y = a) \vee \exists m \exists w\ (m \in n \wedge A[m, w] \wedge Succ[m, p] \wedge w, y\ \epsilon_2\ r)))$

Montrer que la classe binaire r' est fonctionnelle. Soit G l'image du successeur de n par la classe binaire r', construite avec l'axiome de remplacement. Démontrer la proposition $\forall p \forall x\ ((p, x) \in G \Rightarrow A[p, x])$.

Démontrer les propositions

$$\forall p\ (Vide[p] \Rightarrow (p, a) \in G)$$

$$\forall p \forall p' \forall x \forall x'\ ((p \in n \wedge (p, x) \in G \wedge Succ[p, p'] \wedge x, x'\ \epsilon_2\ r) \Rightarrow (p', x') \in G))$$

En déduire la proposition

$$\forall x\ (A[n, x] \Rightarrow (n, x) \in G)$$

Démontrer la proposition

$$\forall n \forall x \forall y\ (((n, x) \in G \wedge (n, y) \in G) \Rightarrow x = y)$$

Démontrer la proposition

$$\forall n \forall x \forall y\ ((A[n, x] \wedge A[n, y]) \Rightarrow x = y)$$

Soit C le sous-ensemble de $\mathbb{N}$ contenant les n tels que $\forall p \forall x \forall y\ ((p \in n \wedge A[p, x] \wedge A[p, y]) \Rightarrow x = y)$. Montrer que l'ensemble C contient 0 et qu'il est clos par successeur. Montrer qu'il contient tous les entiers. En déduire

$$\forall n \forall x \forall y\ ((A[n, x] \wedge A[n, y]) \Rightarrow x = y)$$

3. Soit s la classe binaire définie en compréhension par la proposition A. Montrer que la classe binaire s est fonctionnelle. Soit E l'image de $\mathbb{N}$ par s, construite avec l'axiome de remplacement. Montrer les propositions

$$a \in E$$

$$\forall y \forall y'\ ((y \in E \wedge y, y'\ \epsilon_2\ r) \Rightarrow y' \in E)$$

1.6 Variations sur le tiers exclu

Le principe du tiers exclu, que nous avons exprimé par la règle

$$\frac{}{\Gamma \vdash A \vee \neg A}\ \text{tiers exclu}$$

peut s'exprimer de nombreuses manières alternatives.

1.6.1 La double négation

Une première est de remplacer cette règle par la règle

$$\frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ \text{double négation}$$

ce qui donne un système équivalent.

Proposition 1.11

Les séquents démontrables en déduction naturelle et dans le système dans lequel la règle *tiers exclu* est remplacée par la règle *double négation* sont les mêmes.

Démonstration. Dans un sens, on doit montrer que si le séquent $\Gamma \vdash \neg\neg A$ est démontrable en déduction naturelle, alors c'est également le cas du séquent $\Gamma \vdash A$, ce qui est une conséquence de la proposition 1.7. Dans l'autre, on doit montrer que tous les séquents de la forme $\Gamma \vdash A \vee \neg A$ sont démontrables dans le système dans lequel la règle *tiers exclu* est remplacé par la règle *double négation*. Un tel séquent a la démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \neg(A \vee \neg A), A \vdash \neg(A \vee \neg A)}\,\text{axiome} \quad \dfrac{}{\Gamma, \neg(A \vee \neg A), A \vdash A}\,\text{axiome}}{\Gamma, \neg(A \vee \neg A), A \vdash A \vee \neg A}\,\vee\text{-intro}}{\Gamma, \neg(A \vee \neg A), A \vdash \bot}\,\neg\text{-élim}}{\Gamma, \neg(A \vee \neg A) \vdash \neg A}\,\neg\text{-intro} \quad \dfrac{}{\Gamma, \neg(A \vee \neg A) \vdash \neg(A \vee \neg A)}\,\text{axiome}}{\Gamma, \neg(A \vee \neg A) \vdash A \vee \neg A}\,\vee\text{-intro}}{\Gamma, \neg(A \vee \neg A) \vdash \bot}\,\neg\text{-élim}}{\Gamma \vdash \neg\neg(A \vee \neg A)}\,\neg\text{-intro}}{\Gamma \vdash A \vee \neg A}\,\text{double négation}$$

1.6.2 Les séquents à plusieurs conclusions

De manière plus surprenante, il est aussi possible d'exprimer ce principe du tiers exclu, non par une règle spéciale, mais en changeant la forme des séquents et en considérant des séquents qui ont, non seulement plusieurs hypothèses, mais aussi plusieurs conclusions.

Cela peut se comprendre en analysant la démonstration ci-avant. Dans le cas où le contexte Γ est vide et où la proposition A est juste un symbole de prédicat d'arité nulle, on cherche à démontrer le séquent $\vdash A \vee \neg A$. Si on utilise la règle $\vee$-intro, on est ramené au séquent $\vdash A$ ou au séquent $\vdash \neg A$ et aucun de ces séquents n'est démontrable. La règle $\vee$-intro remplace donc la conclusion

$A \vee \neg A$ du séquent à démontrer par la conclusion A ou la conclusion $\neg A$, mais, ce faisant, elle détruit la proposition $A \vee \neg A$. L'utilisation de la règle *double négation* et de la règle $\neg$-intro peut se comprendre comme un moyen de protéger cette proposition, en en mettant une copie en mémoire, sous la forme de l'hypothèse $\neg(A \vee \neg A)$. On peut ensuite utiliser cette hypothèse autant de fois que l'on veut en la faisant réapparaître dans la conclusion du séquent avec les règles $\neg$-élim et *axiome*. Dans cette démonstration, on utilise cette proposition deux fois, de manière à utiliser deux fois la règle $\vee$-intro et obtenir la première fois $\neg A$ et A la seconde. La conclusion $\neg A$ se transforme en l'hypothèse A et, comme on a l'hypothèse et la conclusion A, on peut conclure avec la règle *axiome*.

Une alternative est de laisser la proposition $A \vee \neg A$ comme une conclusion du séquent, mais cela demande de considérer des séquents dans lesquels il y a, non seulement plusieurs hypothèses, mais aussi plusieurs conclusions et une règle qui permet de dupliquer une conclusion

$$\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \text{contraction}$$

Cela demande en outre que les séquents soient définis, non comme des couples d'ensembles finis, mais comme des couples de multiensembles finis.

Intuitivement, une proposition dans les conclusions d'un séquent joue le même rôle que sa négation dans les hypothèses de ce séquent. Ainsi, si la virgule qui sépare deux hypothèses dans un séquent peut être considérée comme une sorte de *et*, celle qui sépare deux conclusions doit, quant à elle, être considérée comme une sorte de *ou*.

La démonstration ci-avant peut alors se réécrire

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma, A \vdash \bot, A}\ \text{axiome}}{\Gamma, A \vdash \bot, A \vee \neg A}\ \vee\text{-intro}}{\Gamma \vdash \neg A, A \vee \neg A}\ \neg\text{-intro}}{\Gamma \vdash A \vee \neg A, A \vee \neg A}\ \vee\text{-intro}}{\Gamma \vdash A \vee \neg A}\ \text{contraction}$$

Cela mène à la définition alternative de la déduction naturelle.

Définition 1.37 (Les règles du système D')

$$\frac{}{\Gamma \vdash A, \Delta}\ \text{axiome } A \in \Gamma$$

$$\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \text{contraction}$$

$$\frac{}{\Gamma \vdash \top, \Delta}\ \top\text{-intro}$$

$$\frac{\Gamma \vdash \bot, \Delta}{\Gamma \vdash A, \Delta}\ \bot\text{-élim}$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \wedge\text{-intro}$$

$$\frac{\Gamma \vdash A \wedge B, \Delta}{\Gamma \vdash A, \Delta}\ \wedge\text{-élim}$$

$$\frac{\Gamma \vdash A \wedge B, \Delta}{\Gamma \vdash B, \Delta}\ \wedge\text{-élim}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee\text{-intro}$$

$$\frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee\text{-intro}$$

$$\frac{\Gamma \vdash A \vee B, \Delta \quad \Gamma, A \vdash C, \Delta \quad \Gamma, B \vdash C, \Delta}{\Gamma \vdash C, \Delta}\ \vee\text{-élim}$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}\ \Rightarrow\text{-intro}$$

$$\frac{\Gamma \vdash A \Rightarrow B, \Delta \quad \Gamma \vdash A, \Delta}{\Gamma \vdash B, \Delta}\ \Rightarrow\text{-élim}$$

$$\frac{\Gamma, A \vdash \bot, \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg\text{-intro}$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash \neg A, \Delta}{\Gamma \vdash \bot, \Delta}\ \neg\text{-élim}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x\ A, \Delta}\ \forall\text{-intro } x \text{ non libre dans } \Gamma, \Delta$$

$$\frac{\Gamma \vdash \forall x\ A, \Delta}{\Gamma \vdash (t/x)A, \Delta}\ \forall\text{-élim}$$

$$\frac{\Gamma \vdash (t/x)A, \Delta}{\Gamma \vdash \exists x\ A, \Delta}\ \exists\text{-intro}$$

$$\frac{\Gamma \vdash \exists x\ A, \Delta \quad \Gamma, A \vdash B, \Delta}{\Gamma \vdash B, \Delta}\ \exists\text{-élim } x \text{ non libre dans } \Gamma, \Delta, B$$

Proposition 1.12

Un séquent $\Gamma \vdash A$ est démontrable en déduction naturelle si et seulement s'il est démontrable dans le système D'.

Démonstration. Si le séquent $\Gamma \vdash A$ a une démonstration en déduction naturelle, on montre par récurrence sur la structure de cette démonstration qu'il est démontrable dans le système D'. Toutes les règles de la déduction naturelles sont des règles du système D', sauf la règle *tiers exclu*. Mais, comme on l'a vu, les séquents de la forme $\Gamma \vdash A \vee \neg A$ sont démontrables dans le système D'.

Réciproquement, en utilisant la proposition 1.7, il suffit de montrer que si le séquent $\Gamma \vdash A$ est démontrable dans le système D', alors le séquent $\Gamma, \neg A \vdash \bot$ est démontrable en déduction naturelle. On montre, plus généralement, que si le séquent $\Gamma \vdash \Delta$ est démontrable dans le système D', alors le séquent $\Gamma, \neg\Delta \vdash \bot$ est démontrable en déduction naturelle, où $\neg\Delta$ est l'ensemble formé des négations des propositions de Δ. Cela se montre par récurrence sur la structure de la démonstration de $\Gamma \vdash \Delta$ dans le système D'. Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A, A, \Delta'}}{\Gamma \vdash A, \Delta'}\,\text{contraction}$$

alors les ensembles $\Gamma, \neg A, \neg A, \neg\Delta$ et $\Gamma, \neg A, \neg\Delta$ sont identiques et donc, par hypothèse de récurrence, le séquent $\Gamma, \neg A, \neg\Delta \vdash \bot$ est démontrable en déduction naturelle. Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma_1 \vdash A_1, \Delta'} \quad \ldots \quad \dfrac{\pi_n}{\Gamma_n \vdash A_n, \Delta'}}{\Gamma \vdash B, \Delta'}\, r$$

où r est une règle du système D', distincte de la règle *contraction*, alors par hypothèse de récurrence, les séquents $\Gamma_1, \neg A_1, \neg\Delta' \vdash \bot, \ldots, \Gamma_n, \neg A_n, \neg\Delta' \vdash \bot$ sont démontrables en déduction naturelle. D'après la proposition 1.7, les séquents $\Gamma_1, \neg\Delta' \vdash A_1, \ldots, \Gamma_n, \neg\Delta' \vdash A_n$ le sont également. En utilisant la règle de déduction naturelle homologue de la règle r de D', on construit une démonstration en déduction naturelle du séquent $\Gamma, \neg\Delta' \vdash B$ et donc, en utilisant la proposition 1.7 à nouveau, une démonstration du séquent $\Gamma, \neg B, \neg\Delta' \vdash \bot$.

La proposition suivante qui est l'analogue, pour le système D' de la proposition 1.6.

Proposition 1.13 (L'affaiblissement)

Si le séquent $\Gamma \vdash \Delta$ est démontrable dans le système D' alors c'est également le cas des séquents $\Gamma, A \vdash \Delta$ et $\Gamma \vdash A, \Delta$.

Démonstration. Par récurrence sur la structure de la démonstration de $\Gamma \vdash \Delta$.

Quand on définit une notion de vérité, comme on l'a fait dans ce chapitre, on ne peut éviter de s'interroger sur les outils qu'il est légitime d'utiliser dans cette définition. D'un côté, on ne peut pas faire comme si on ne connaissait rien. Si on ne disposait d'aucun langage, d'aucun concept, d'aucune notion préalable de vérité, comment pourrait-on donner une définition ? D'un autre côté, si on disposait déjà de la notion de vérité mathématique toute entière, on pourrait donner la définition triviale : la proposition « Tout espace vectoriel admet une base » est vraie si tout espace vectoriel admet une base.

Dans ce chapitre, nous avons donné une définition intermédiaire entre ces deux extrêmes : la proposition « Tout espace vectoriel admet une base » est vraie s'il existe une démonstration π de cette proposition.

Pour énoncer cette définition, il est certes nécessaire de disposer de notions mathématiques comme celles de nombre entier, d'ensemble fini ou d'arbre. Cependant, une démonstration est constituée d'un nombre fini de symboles et le fait qu'une suite de symboles soit une démonstration d'une proposition ou non est quelque chose que chacun peut vérifier : il suffit de vérifier que chaque étape est bien l'application d'une règle de la déduction naturelle. La proposition « La suite de symboles π est une démonstration de la proposition "Tout espace vectoriel admet une base" » n'utilise donc que des objets finis et des relations vérifiables entre ces objets. On dit qu'une telle proposition est *combinatoire*. La notion préalable de vérité dont il est nécessaire de disposer pour pouvoir comprendre la définition que nous avons donnée ici se limite donc à la notion élémentaire de vérité des propositions combinatoires.

2
Les modèles

Après avoir donné une définition de la notion de démonstration, au chapitre précédent, nous allons, dans ce chapitre, étudier certaines propriétés des démonstrations. En particulier nous allons introduire des outils qui permettent de démontrer des *résultats d'indépendance* de la forme : il n'existe pas de démonstration de la proposition A dans la théorie $\mathcal{T}$.

S'il était nécessaire de se restreindre à des propositions combinatoires pour définir la notion de démonstration, aucune restriction n'est, en revanche, nécessaire pour étudier les démonstrations : pour démontrer des résultats d'indépendance, nous pouvons utiliser tous les outils mathématiques que nous voulons.

2.1 La notion de modèle

Définition 2.1 (Modèle)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage. Un *modèle* de ce langage est une structure $\mathcal{M} = ((\mathcal{M}_s)_{s\in\mathcal{S}}, \mathcal{B}, \mathcal{B}^+, (\hat{f})_{f\in\mathcal{F}}, (\hat{P})_{P\in\mathcal{P}}, \hat{\top}, \hat{\bot}, \hat{\neg}, \hat{\wedge}, \hat{\vee}, \hat{\Rightarrow}, \hat{\forall}, \hat{\exists})$ formée,

- pour chaque sorte s de $\mathcal{S}$, d'un ensemble non vide $\mathcal{M}_s$,
- d'un ensemble non vide $\mathcal{B}$, d'un sous-ensemble $\mathcal{B}^+$ de $\mathcal{B}$,
- pour chaque symbole de fonction f de $\mathcal{F}$ d'arité $(s_1, \ldots, s_n, s')$ d'une fonction $\hat{f}$ de $\mathcal{M}_{s_1} \times \ldots \times \mathcal{M}_{s_n}$ dans $\mathcal{M}_{s'}$,
- pour chaque symbole de prédicat P de $\mathcal{P}$ d'arité $(s_1, \ldots, s_n)$ d'une fonction $\hat{P}$ de $\mathcal{M}_{s_1} \times \ldots \times \mathcal{M}_{s_n}$ dans $\mathcal{B}$,

– de deux éléments $\hat{\top}$ et $\hat{\bot}$ de $\mathcal{B}$, d'une fonction $\hat{\neg}$ de $\mathcal{B}$ dans $\mathcal{B}$, de fonctions $\hat{\wedge}$, $\hat{\vee}$ et $\hat{\Rightarrow}$ de $\mathcal{B} \times \mathcal{B}$ dans $\mathcal{B}$ et de deux fonctions $\hat{\forall}$ et $\hat{\exists}$ de $\wp^+(\mathcal{B})$ dans $\mathcal{B}$ où $\wp^+(\mathcal{B})$ est l'ensemble des parties non vides de $\mathcal{B}$.

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage et $\mathcal{M}$ un modèle de ce langage. On veut définir une fonction $[\![\ ]\!]$ qui associe, à chaque terme terme t de sorte s, un élément $[\![t]\!]$ de $\mathcal{M}_s$ et, à chaque proposition A, un élément $[\![A]\!]$ de $\mathcal{B}$. On veut, en outre, que cette fonction soit un morphisme, c'est-à-dire que $[\![f(t_1, \ldots, t_n)]\!] = \hat{f}([\![t_1]\!], \ldots, [\![t_n]\!])$, $[\![P(t_1, \ldots, t_n)]\!] = \hat{P}([\![t_1]\!], \ldots, [\![t_n]\!])$, $[\![A \wedge B]\!] = \hat{\wedge}([\![A]\!], [\![B]\!])$, ... On sait qu'un morphisme d'espaces vectoriels est complètement défini par son image sur une base de l'espace de départ. De même, un morphisme entre un langage et un modèle est complètement défini par son image sur les variables. Cela mène à la définition suivante.

Définition 2.2 (Valuation)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage, $\mathcal{M}$ un modèle de ce langage et $(\mathcal{V}_s)_{s \in \mathcal{S}}$ une famille d'ensembles de variables. On appelle *valuation* une fonction de domaine fini qui associe aux variables $x_1, \ldots, x_n$ de sortes $s_1, \ldots, s_n$ des éléments $a_1, \ldots, a_n$ de $\mathcal{M}_{s_1}, \ldots, \mathcal{M}_{s_n}$.

La valuation qui lie l'élément a_1 à la variable x_1, ..., a_n à la variable x_n s'écrit $x_1 = a_1, \ldots, x_n = a_n$. Si ϕ est une valuation, x une variable et a un élément de $\mathcal{M}$ on note $(\phi, x = a)$ la valuation qui prend la même valeur que ϕ partout sauf en x où elle vaut a.

Une valuation se prolonge naturellement en un morphisme $[\![\ ]\!]_\phi$ entre les termes et propositions du langage $\mathcal{L}$ dont les variables libres sont dans le domaine de ϕ et le modèle $\mathcal{M}$, en définissant $[\![x]\!]_\phi$ comme $\phi(x)$, $[\![f(t_1, \ldots, t_n)]\!]_\phi$ comme $\hat{f}([\![t_1]\!]_\phi, \ldots, [\![t_n]\!]_\phi)$, ... En fait, les quantificateurs et les variables liées compliquent un tout petit peu la définition. En effet, les variables libres de la proposition A sont toutes celles de $\forall x\ A$ plus, potentiellement, x. Pour définir $[\![\forall x\ A]\!]_\phi$, on doit donc commencer par considérer toutes les valeurs $[\![A]\!]_{\phi, x=a}$ obtenues en associant à x un élément quelconque a de $\mathcal{M}_s$, on obtient alors un sous-ensemble non vide de $\mathcal{B}$ auquel on applique la fonction $\hat{\forall}$, qui doit donc être une fonction de l'ensemble des parties non vides de $\mathcal{B}$ dans $\mathcal{B}$.

Définition 2.3 (Dénotation)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage, $\mathcal{M}$ un modèle de ce langage, $(\mathcal{V}_s)_{s \in \mathcal{S}}$ une famille d'ensembles de variables, ϕ une valuation et t un terme dont les variables libres

sont dans le domaine de ϕ, la *dénotation* du terme t dans le modèle $\mathcal{M}$ pour la valuation ϕ est l'élément $[\![t]\!]_\phi$ de $\mathcal{M}_s$ ainsi défini par récurrence sur la structure de t

- $[\![x]\!]_\phi = \phi(x)$,
- $[\![f(t_1, \ldots, t_n)]\!]_\phi = \hat{f}([\![t_1]\!]_\phi, \ldots, [\![t_n]\!]_\phi)$.

Soit A une proposition dont les variables libres sont dans le domaine de ϕ, la *dénotation* de la proposition A dans le modèle $\mathcal{M}$ pour la valuation ϕ est l'élément $[\![A]\!]_\phi$ de $\mathcal{B}$ ainsi défini par récurrence sur la structure de A

- $[\![P(t_1, \ldots, t_n)]\!]_\phi = \hat{P}([\![t_1]\!]_\phi, \ldots, [\![t_n]\!]_\phi)$,
- $[\![\top]\!]_\phi = \hat{\top}$,
- $[\![\bot]\!]_\phi = \hat{\bot}$,
- $[\![\neg A]\!]_\phi = \hat{\neg}([\![A]\!]_\phi)$,
- $[\![A \wedge B]\!]_\phi = \hat{\wedge}([\![A]\!]_\phi, [\![B]\!]_\phi)$,
- $[\![A \vee B]\!]_\phi = \hat{\vee}([\![A]\!]_\phi, [\![B]\!]_\phi)$,
- $[\![A \Rightarrow B]\!]_\phi = \hat{\Rightarrow}([\![A]\!]_\phi, [\![B]\!]_\phi)$,
- $[\![\forall x\ A]\!]_\phi = \hat{\forall}(\{[\![A]\!]_{\phi, x=a} \mid a \in \mathcal{M}_s\})$,
- $[\![\exists x\ A]\!]_\phi = \hat{\exists}(\{[\![A]\!]_{\phi, x=a} \mid a \in \mathcal{M}_s\})$.

Proposition 2.1 (Substitution)

$$[\![(u/x)t]\!]_\phi = [\![t]\!]_{\phi, x=[\![u]\!]_\phi}$$
$$[\![(u/x)A]\!]_\phi = [\![A]\!]_{\phi, x=[\![u]\!]_\phi}$$

Démonstration. Par récurrence sur la structure de t et sur celle de A.

Définition 2.4 (Validité)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage, $\mathcal{M}$ un modèle de ce langage, $(\mathcal{V}_s)_{s \in \mathcal{S}}$ une famille d'ensembles de variables. Une proposition close est *valide* dans le modèle $\mathcal{M}$ si $[\![A]\!]_\emptyset$ appartient à l'ensemble $\mathcal{B}^+$. On dit aussi dans ce cas que $\mathcal{M}$ *est un modèle* de A.

Une proposition A qui contient les variables libres $x_1, \ldots, x_n$ est *valide* dans le modèle $\mathcal{M}$ si la proposition close $\forall x_1 \ldots \forall x_n\ A$ l'est, c'est-à-dire si pour toute valuation ϕ, dont le domaine contient les variables $x_1, \ldots, x_n$, $[\![A]\!]_\phi$ appartient à l'ensemble $\mathcal{B}^+$.

Un séquent $A_1, \ldots, A_n \vdash B_1, \ldots, B_p$ est *valide* dans le modèle $\mathcal{M}$ si la proposition $(A_1 \wedge \ldots \wedge A_n) \Rightarrow (B_1 \vee \ldots \vee B_p)$ l'est.

Une théorie $\mathcal{T}$ est *valide* dans un modèle si tous ses axiomes le sont.

Définition 2.5 (Modèle bivalué)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage. Un modèle *bivalué* de $\mathcal{L}$ est un modèle dans lequel $\mathcal{B} = \{0,1\}$, $\mathcal{B}^+ = \{1\}$, $\hat{\top} = 1$, $\hat{\bot} = 0$ et $\hat{\neg}$, $\hat{\wedge}$, $\hat{\vee}$, $\hat{\Rightarrow}$ $\hat{\forall}$ et $\hat{\exists}$ sont les fonctions

$\hat{\neg}$	0	1
	1	0

$\hat{\wedge}$	0	1
0	0	0
1	0	1

$\hat{\vee}$	0	1
0	0	1
1	1	1

$\hat{\Rightarrow}$	0	1
0	1	1
1	0	1

$\hat{\forall}$	$\{0\}$	$\{0,1\}$	$\{1\}$
	0	0	1

$\hat{\exists}$	$\{0\}$	$\{0,1\}$	$\{1\}$
	0	1	1

Dans la suite de ce livre, tous les modèles seront bivalués.

Exercice 2.1

On considère le langage à une sorte de termes formé d'un symbole de fonction binaire $+$ et d'un symbole de prédicat binaire $=$. Soit $\mathcal{M}_1$ le modèle formé de l'ensemble $\mathbb{N}$, de l'addition sur $\mathbb{N}$ et de la fonction caractéristique de l'égalité sur $\mathbb{N}$, c'est-à-dire de la fonction $\hat{=}$ de $\mathbb{N}^2$ dans $\{0,1\}$ telle que $\hat{=}(n,p) = 1$ si $n = p$ et $\hat{=}(n,p) = 0$ sinon. La proposition $\forall x \forall y \exists z\ (x + z = y)$ est-elle valide dans ce modèle ?

Même question pour le modèle $\mathcal{M}_2$ formé de l'ensemble $\mathbb{Z}$ de l'addition et de la fonction caractéristique de l'égalité sur $\mathbb{Z}$.

La proposition $\forall x \forall y\ (x+y = y+x)$ est-elle valide dans $\mathcal{M}_1$? Et dans $\mathcal{M}_2$? Donner un exemple de modèle dans lequel cette proposition n'est pas valide.

2.2 Le théorème de correction

Un intérêt de la notion de modèle est que la validité dans un modèle est un invariant de la démontrabilité : tous les séquents démontrables sont donc valides dans tous les modèles. De ce fait, si une proposition est démontrable dans une théorie, alors elle est valide dans tous les modèles de cette théorie. Cela donne une méthode pour montrer qu'une proposition n'est pas démontrable dans une certaine théorie : il suffit de montrer qu'il existe un modèle de cette théorie dans lequel elle n'est pas valide. Ce principe est exprimé par la deuxième forme du théorème de correction ci-après.

Proposition 2.2

Si un séquent $A_1, \ldots, A_n \vdash B_1, \ldots, B_p$ est démontrable en déduction naturelle, alors il est valide dans tous les modèles.

Démonstration. Par récurrence sur la structure des démonstrations.

De cette proposition, on peut déduire le théorème de correction qui peut se formuler sous trois formes équivalentes.

Théorème 2.1 (Correction)

Soit $\mathcal{T}$ une théorie et A une proposition.

1. Si A est démontrable dans $\mathcal{T}$, alors A est valide dans tous les modèles de $\mathcal{T}$.
2. S'il existe un modèle de $\mathcal{T}$ qui n'est pas un modèle de A, alors A n'est pas démontrable dans $\mathcal{T}$.
3. Si $\mathcal{T}$ a un modèle, alors $\mathcal{T}$ est cohérente.

Démonstration. Soit $\mathcal{M}$ un modèle de la théorie $\mathcal{T}$ et A une proposition démontrable dans $\mathcal{T}$. Il existe un sous-ensemble fini $H_1, \ldots, H_n$ de $\mathcal{T}$, tel que le séquent $H_1, \ldots, H_n \vdash A$ soit démontrable. D'après la proposition 2.2, ce séquent est valide dans $\mathcal{M}$, c'est-à-dire que la proposition $(H_1 \wedge \ldots \wedge H_n) \Rightarrow A$ est valide dans ce modèle. Les propositions $H_1, \ldots H_n$ étant valides dans $\mathcal{M}$ on en déduit que A également est valide dans $\mathcal{M}$, ce qui montre la proposition (1.) La proposition (2.) est une conséquence triviale de (1.) et la proposition (3.) est une conséquence de (2.), en prenant $A = \bot$.

Exercice 2.2

Soit la théorie formée de l'axiome $P(c) \vee Q(c)$. Montrer que la proposition $P(c)$ n'est pas démontrable dans cette théorie. Montrer que la proposition $\neg P(c)$ n'est pas démontrable non plus. Qu'en est-il de la proposition $Q(c)$?

On peut utiliser le théorème de correction pour démontrer que l'axiome de l'infini n'est pas démontrable à partir des autres axiomes ZF.

Définition 2.6 (L'ensemble des ensembles héréditairement finis)

Soit V_n la suite d'ensembles définie par récurrence par $V_0 = \varnothing$ et $V_{i+1} = \wp(V_i)$. Soit $V_\omega = \cup_i V_i$.

Proposition 2.3

Soit le modèle $\mathcal{M} = (\mathcal{M}_\iota, \mathcal{M}_\sigma, \hat{\epsilon}_2, \hat{=}, \hat{\in})$ où $\mathcal{M}_\iota = V_\omega$, $\mathcal{M}_\sigma = \wp(\mathcal{M}_\iota \times \mathcal{M}_\iota)$, $\hat{\epsilon}_2$ la fonction de $\mathcal{M}_\iota \times \mathcal{M}_\iota \times \mathcal{M}_\sigma$ dans $\{0,1\}$ telle que $\hat{\epsilon}_2(a,b,c) = 1$ si (a,b) appartient à c et $\hat{\epsilon}_2(a,b,c) = 0$ sinon, $\hat{=}$ la fonction de $\mathcal{M}_\iota \times \mathcal{M}_\iota$ dans $\{0,1\}$ telle que $\hat{=}(a,b) = 1$ si $a = b$ et $\hat{=}(a,b) = 0$ sinon, $\hat{\in}$ la fonction de $\mathcal{M}_\iota \times \mathcal{M}_\iota$ dans $\{0,1\}$ telle que $\hat{\in}(a,b) = 1$ si a appartient à b et $\hat{\in}(a,b) = 0$ sinon.

Le modèle $\mathcal{M}$ est un modèle de tous les axiomes de ZF sauf l'axiome de l'infini.

Démonstration. On montre, par exemple, que c'est un modèle de l'axiome de la réunion. Remarquons tout d'abord que la réunion d'une famille de parties de V_j étant une partie de V_j, la réunion d'une famille d'éléments de V_{j+1} est un élément de V_{j+1}. On montre ensuite que, si c est un élément de V_ω, la réunion $\bigcup_{b\in c} b$ des éléments de c appartient également à V_ω. Comme $c \in V_\omega$, il existe, par définition de V_ω, un entier i non nul tel que $c \in V_i$. Si $i = 1$, $c = \varnothing$ et la réunion des éléments de c est également l'ensemble vide, c'est donc un élément de V_ω. Sinon, il existe un entier j tel que $i = j + 2$. On a $c \in V_{j+2}$, donc $c \subseteq V_{j+1}$ et les éléments de c appartiennent à V_{j+1}. C'est donc aussi le cas de la réunion des éléments de c, qui appartient donc à V_ω. On a donc

$$[\![\forall w\ (w \in z \Leftrightarrow (\exists v\ (w \in v \wedge v \in x)))]\!]_{x=c, z=\bigcup_{b\in c} b} = 1$$

et donc

$$[\![\forall x \exists z \forall w\ (w \in z \Leftrightarrow (\exists v\ (w \in v \wedge v \in x)))]\!] = 1$$

On montre de même que l'axiome d'extensionalité, l'axiome des parties et l'axiome de remplacement sont valides dans ce modèle.

Les axiomes de l'égalité et le schéma de compréhension binaire sont trivialement valides dans ce modèle.

On montre enfin, par l'absurde, que l'axiome de l'infini n'est pas valide dans ce modèle. On commence par montrer, par récurrence sur i, que tous les éléments de V_i sont des ensembles finis. On en déduit que tous les éléments de V_ω sont des ensembles finis.

Si l'axiome de l'infini était valide dans V_ω, il existerait un ensemble a dans V_ω qui contient l'ensemble vide et qui contient l'ensemble $b \cup \{b\}$ chaque fois qu'il contient un ensemble b. Cet ensemble contiendrait donc tous les éléments de la suite définie par récurrence par : $e_0 = \varnothing, e_1 = \{e_0\}, e_2 = \{e_0, e_1\}, e_3 = \{e_0, e_1, e_2\}, \ldots, e_{i+1} = e_i \cup \{e_i\}, \ldots$ Ces éléments étant tous distincts, l'ensemble a serait infini.

Proposition 2.4

L'axiome de l'infini n'est pas démontrable à partir des autres axiomes de ZF.

Démonstration. Tous les axiomes de ZF, sauf l'axiome de l'infini, sont valides dans le modèle $\mathcal{M}$ de la proposition 2.3.

2.3 Le théorème de complétude

Nous avons vu que, d'après le théorème de correction, si une proposition A est démontrable dans une théorie $\mathcal{T}$, alors elle est valide dans tous les modèles de cette théorie. Le théorème de complétude, démontré par K. Gödel en 1930, mais qui n'est pas le célèbre théorème de Gödel, est la réciproque de ce théorème.

2.3.1 Les trois formes du théorème de complétude

Comme le théorème de correction, le théorème de complétude peut se formuler sous trois formes équivalentes.

Théorème 2.2 (Complétude)

Soit $\mathcal{T}$ une théorie et A une proposition.

1. Si A est valide dans tous les modèles de $\mathcal{T}$, alors A est démontrable dans $\mathcal{T}$.
2. Si A n'est pas démontrable dans $\mathcal{T}$, alors il existe un modèle de $\mathcal{T}$ qui n'est pas un modèle de A.
3. Si $\mathcal{T}$ est cohérente, alors $\mathcal{T}$ a un modèle.

Les formes (1.) et (2.) sont trivialement équivalentes. La forme (3.) est une conséquence de (2.) en prenant $A = \bot$. Montrons que la forme (2.) est une conséquence de (3.). Considérons une théorie $\mathcal{T}$ et une proposition A qui n'est pas démontrable dans cette théorie. D'après la proposition 1.7, la proposition $\bot$ n'est pas démontrable dans la théorie $\mathcal{T}, \neg A$. D'après (3.), la théorie $\mathcal{T}, \neg A$ a donc un modèle. Ce modèle est un modèle de $\mathcal{T}$, mais pas un modèle de A.

2.3.2 La démonstration du théorème de complétude

Nous allons démontrer le théorème de complétude dans sa forme (3.) et nous restreindre au cas d'un langage fini ou dénombrable.

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un tel langage et $\mathcal{T}$ une théorie cohérente dans ce langage. Nous devons construire un modèle de cette théorie. L'idée est de définir le domaine $\mathcal{M}_s$ comme l'ensemble des termes clos de sorte s, de définir la fonction $\hat{f}$ comme la fonction qui aux termes clos $t_1, \ldots, t_n$ associe le terme $f(t_1, \ldots, t_n)$ et de définir la fonction $\hat{P}$ comme la fonction qui à $t_1, \ldots, t_n$ associe 1 ou 0 selon que la proposition $P(t_1, \ldots, t_n)$ est démontrable ou non.

Cependant, même en supposant la théorie $\mathcal{T}$ cohérente, le modèle ainsi construit n'est pas toujours un modèle de cette théorie. Si la théorie $\mathcal{T}$ est constituée, par exemple, d'un seul axiome $P(c) \vee Q(c)$, ni la proposition $P(c)$ ni la proposition $Q(c)$ ne sont démontrables — voir l'exercice 2.2. Donc, en suivant la construction ci-avant, nous serons amenés à poser $\hat{P}(c) = 0$ et $\hat{Q}(c) = 0$. Ainsi, la proposition $P(c) \vee Q(c)$ ne sera pas valide dans ce modèle.

Pour que cette construction fonctionne, il est donc nécessaire, dans un premier temps, de compléter la théorie : quand une proposition A est indéterminée, c'est-à-dire que ni A ni $\neg A$ ne sont démontrables, il faut faire un choix et ajouter l'axiome A ou l'axiome $\neg A$. Si, dans cet exemple, on ajoute l'axiome $P(c)$ alors, en construisant le modèle, on sera amené à poser $\hat{P}(c) = 1$ et, de ce fait, la proposition $P(c) \vee Q(c)$ sera valide dans le modèle. Si, en revanche, on pose l'axiome $\neg P(c)$, alors la proposition $Q(c)$ devient démontrable, en construisant le modèle, on sera amené à poser $\hat{Q}(c) = 1$ et la proposition $P(c) \vee Q(c)$ sera donc encore valide.

Toutefois, compléter ainsi la théorie n'est pas suffisant. Par exemple, pour la théorie $\neg P(c), \exists x\ P(x)$, la construction ci-avant nous mène à poser $\mathcal{M} = \{c\}$ et $\hat{P}(c) = 0$. La proposition $\exists x\ P(x)$ n'est donc pas valide dans ce modèle. Le problème est ici qu'il n'y a pas de terme clos témoin du fait qu'il existe un objet qui vérifie la propriété P. Il est donc nécessaire, avant de construire le modèle ci-avant d'ajouter une constante d et l'axiome $P(d)$. Cette constante d s'appelle le *témoin de Henkin* de la proposition $\exists x\ P(x)$.

La démonstration du théorème de complétude demande donc de montrer d'abord la proposition suivante.

Proposition 2.5

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage et $\mathcal{T}$ une théorie cohérente dans ce langage. Il existe un langage $\mathcal{L}'$ tel que $\mathcal{L} \subseteq \mathcal{L}'$ et une théorie $\mathcal{U}$ dans le langage $\mathcal{L}'$ telle que $\mathcal{T} \subseteq \mathcal{U}$ et qui vérifie les propriétés suivantes.

1. La théorie $\mathcal{U}$ est cohérente.
2. Pour toute proposition close A dans le langage $\mathcal{L}'$, la proposition A est démontrable dans $\mathcal{U}$ ou la proposition $\neg A$ est démontrable dans $\mathcal{U}$.
3. Si la proposition $\exists x\ A$ est démontrable dans $\mathcal{U}$, alors il existe une constante c telle que $(c/x)A$ soit démontrable dans $\mathcal{U}$.

La démonstration de cette proposition est similaire à celle du théorème de la base incomplète : on examine les propositions une par une de manière à sélectionner certaines d'entre elles. Quand on examine la proposition A, on se demande si A ou $\neg A$ sont démontrables à partir des axiomes de $\mathcal{T}$ et des propositions déjà sélectionnées. Si A est démontrable, on la sélectionne. Si $\neg A$ est démontrable, on la sélectionne. Si ni A ni $\neg A$ ne sont démontrables, alors on sélectionne arbitrairement A. Si en outre A a la forme $\exists x\ B$, alors on sélectionne également la proposition $(c/x)B$ où c est une nouvelle constante que l'on ajoute au langage.

Démonstration. Soit $\mathcal{H} = \{c_i^s\}$ un ensemble dénombrable contenant une infinité de constantes $c_0^s, c_1^s, c_2^s, \ldots$ de chaque sorte s. Soit $\mathcal{L}'$ le langage $(\mathcal{S}, \mathcal{F} \uplus \mathcal{H}, \mathcal{P})$.

Le langage $\mathcal{L}'$ et les ensembles $\mathcal{V}_s$ étant dénombrables, l'ensemble des propositions de ce langage est dénombrable. Soit $A_0, A_1, A_2, \ldots$ une énumération de cet ensemble. On définit la famille de théories $\mathcal{U}_n$ ainsi. On pose $\mathcal{U}_0 = \mathcal{T}$. Si A_n est démontrable dans la théorie $\mathcal{U}_n$, alors on pose $B = A_n$, si $\neg A_n$ est démontrable dans la théorie $\mathcal{U}_n$, alors on pose $B = \neg A_n$ et si ni A_n ni $\neg A_n$ ne sont démontrables dans la théorie $\mathcal{U}_n$, alors on pose arbitrairement $B = A_n$. Si B n'a pas la forme $\exists x\ C$, alors on pose $\mathcal{U}_{n+1} = \mathcal{U}_n \cup \{B\}$ et si B a la forme $\exists x\ C$, alors on pose $\mathcal{U}_{n+1} = \mathcal{U}_n \cup \{B, (c_i^s/x)C\}$, où s est la sorte de x et i est le plus petit entier tel que la constante c_i^s n'apparaisse pas dans $\mathcal{U}_n$ ni dans B. Une telle constante existe toujours, car seules un nombre fini de constantes de $\mathcal{H}$ apparaissent dans chaque $\mathcal{U}_i$. Enfin, on pose $\mathcal{U} = \bigcup_i \mathcal{U}_i$.

On montre par récurrence sur i que toutes les théories $\mathcal{U}_i$ sont cohérentes. On en déduit que la théorie $\mathcal{U}$, elle-même, est cohérente. En effet, s'il existait une démonstration de $\bot$ dans $\mathcal{U}$, il existerait un sous-ensemble fini $B_1, \ldots, B_n$ de $\mathcal{U}$ tel que le séquent $B_1, \ldots, B_n \vdash \bot$ soit démontrable. Chaque proposition B_j appartiendrait à un ensembles $\mathcal{U}_{i_j}$ et toutes appartiendraient à $\mathcal{U}_k$ où k est le plus grand des i_j. La théorie $\mathcal{U}_k$ serait donc contradictoire, ce qui n'est pas le cas.

Soit A une proposition close quelconque. Il existe un indice i tel que $A_i = A$ et l'une des propositions A ou $\neg A$ est un élément de $\mathcal{U}_{i+1}$. De ce fait, la théorie $\mathcal{U}$ contient l'axiome A ou l'axiome $\neg A$ et démontre donc l'une de ces propositions.

Enfin, si la proposition $\exists x\ A$ est démontrable dans $\mathcal{U}$, alors il existe un indice i tel que $A_i = \exists x\ A$. Comme la théorie $\mathcal{U}_i$ est cohérente et démontre la proposition A_i, elle ne démontre pas la proposition $\neg A_i$. De ce fait, $\mathcal{U}_{i+1} = \mathcal{U}_i \cup \{\exists x\ A, (c/x)A\}$ pour une certaine constante c. La théorie $\mathcal{U}$ contient donc l'axiome $(c/x)A$ et démontre donc cette proposition.

Proposition 2.6

Soit $\mathcal{U}$ une théorie qui vérifie les propriétés suivantes.

1. La théorie $\mathcal{U}$ est cohérente.
2. Pour toute proposition close A, la proposition A ou la proposition $\neg A$ est démontrable dans $\mathcal{U}$.
3. Si la proposition $\exists x\ A$ est démontrable dans $\mathcal{U}$, alors il existe un terme clos t telle que la proposition $(t/x)A$ soit démontrable dans $\mathcal{U}$.

Alors

- La proposition $\neg A$ est démontrable si et seulement si la proposition A n'est pas démontrable dans $\mathcal{U}$.
- La proposition $A \wedge B$ est démontrable dans $\mathcal{U}$ si et seulement si la proposition A est démontrable dans $\mathcal{U}$ et la proposition B est démontrable dans $\mathcal{U}$.
- La proposition $A \vee B$ est démontrable dans $\mathcal{U}$ si et seulement si la proposition A est démontrable dans $\mathcal{U}$ ou la proposition B est démontrable dans $\mathcal{U}$.
- La proposition $A \Rightarrow B$ est démontrable dans $\mathcal{U}$ si et seulement si si la proposition A est démontrable dans $\mathcal{U}$, alors la proposition B est démontrable dans $\mathcal{U}$.
- La proposition $\forall x\ A$ est démontrable dans $\mathcal{U}$ si et seulement pour tout terme clos t, la proposition $(t/x)A$ est démontrable dans $\mathcal{U}$.
- La proposition $\exists x\ A$ est démontrable dans $\mathcal{U}$ si et seulement s'il existe un terme clos t, tel que la proposition $(t/x)A$ soit démontrable dans $\mathcal{U}$.

Démonstration.

- Si la proposition A est démontrable dans $\mathcal{U}$, la théorie $\mathcal{U}$ étant cohérente, la proposition $\neg A$ n'est pas démontrable dans $\mathcal{U}$. Réciproquement, d'après la deuxième condition, si la proposition $\neg A$ n'est pas démontrable dans $\mathcal{U}$, la proposition A est démontrable dans $\mathcal{U}$.
- Si les propositions A et B sont démontrables dans $\mathcal{U}$ alors la proposition $A \wedge B$ l'est également en utilisant de la règle $\wedge$-intro. Réciproquement, si la proposition $A \wedge B$ est démontrable dans $\mathcal{U}$, les propositions A et B le sont également, en utilisant de la règle $\wedge$-élim.
- Si la proposition A ou la proposition B est démontrable dans $\mathcal{U}$, alors la proposition $A \vee B$ l'est également, en utilisant la règle $\vee$-intro. Réciproquement, si la proposition $A \vee B$ est démontrable dans $\mathcal{U}$, alors, d'après la deuxième condition, la proposition A ou la proposition $\neg A$ est démontrable dans $\mathcal{U}$. Dans le premier cas, la proposition A est démontrable dans $\mathcal{U}$, dans le second, comme les proposition $A \vee B$ et $\neg A$ sont démontrables, la proposition B est démontrable dans $\mathcal{U}$ avec les règles *axiome*, $\neg$-élim, $\bot$-élim et $\vee$-élim.
- Supposons que si la proposition A est démontrable dans $\mathcal{U}$ alors la proposition B est démontrable dans $\mathcal{U}$. Dans ce cas, d'après la deuxième

condition, ou bien la proposition A est démontrable dans $\mathcal{U}$ ou bien la proposition $\neg A$ est démontrable dans $\mathcal{U}$. Dans le premier cas, la proposition B est démontrable dans $\mathcal{U}$ et donc la proposition $A \Rightarrow B$ est démontrable avec la règle $\Rightarrow$-intro. Dans le second, la proposition $\neg A$ est démontrable et donc la proposition $A \Rightarrow B$ est démontrable avec les règles $\Rightarrow$-intro, $\bot$-élim et $\neg$-élim. Réciproquement, si $A \Rightarrow B$ est démontrable dans $\mathcal{U}$, alors, si A est démontrable dans $\mathcal{U}$, alors B est démontrable dans $\mathcal{U}$ avec la règle $\Rightarrow$-élim.

- Supposons que pour tout terme clos t, la proposition $(t/x)A$ soit démontrable dans $\mathcal{U}$. Si la proposition $\exists x\ \neg A$ était démontrable dans $\mathcal{U}$, alors, d'après la troisième condition, il existerait un terme clos t telle que $\neg(t/x)A$ soit démontrable. La théorie $\mathcal{U}$ serait alors contradictoire. La proposition $\exists x\ \neg A$ n'est donc pas démontrable dans $\mathcal{U}$ et donc la proposition $\neg\exists x\ \neg A$ l'est. La proposition $\forall x\ A$ est donc démontrable d'après la proposition 1.8. Réciproquement, si la proposition $\forall x\ A$ est démontrable dans la théorie $\mathcal{U}$, toutes les propositions $(t/x)A$ sont démontrables avec la règle $\forall$-élim.
- S'il existe un terme clos t, tel que la proposition $(t/x)A$ soit démontrable dans $\mathcal{U}$ alors, la proposition $\exists x\ A$ est démontrable avec la règle $\exists$-intro. Réciproquement, si la proposition $\exists x\ A$ est démontrable dans $\mathcal{U}$ alors, d'après la troisième condition, il existe un terme clos t telle que $(t/x)A$ soit démontrable dans $\mathcal{U}$.

On peut enfin démontrer le théorème de complétude.

Démonstration. Soit $\mathcal{T}$ une théorie cohérente et $\mathcal{U}$ la théorie construite à la proposition 2.5. On définit le domaine $\mathcal{M}_s$ comme l'ensemble des termes clos de sorte s du langage $\mathcal{L}'$, on définit la fonction $\hat{f}$ comme la fonction qui aux termes clos $t_1, \ldots, t_n$ associe le terme $f(t_1, \ldots, t_n)$ et la fonction $\hat{P}$ comme la fonction qui à $t_1, \ldots, t_n$ associe 1 ou 0 selon que la proposition $P(t_1, \ldots, t_n)$ est démontrable dans $\mathcal{U}$ ou non.

Soit A une proposition close. On démontre par récurrence sur la structure de la proposition A que A est démontrable dans $\mathcal{U}$ si et seulement si A est valide dans ce modèle. Si A est une proposition atomique, l'équivalence est une simple conséquence de la définition des fonctions $\hat{P}$. Si A est de la forme $B \wedge C$, la proposition A est démontrable dans $\mathcal{U}$ si et seulement si les propositions B et C le sont également — proposition 2.6 — si et seulement si les propositions B et C sont valides dans $\mathcal{M}$ — hypothèse de récurrence — si et seulement si la proposition A est valide dans $\mathcal{M}$. On procède de même dans les autres cas.

Dans cette démonstration, on s'est restreint au cas des langages finis ou dénombrables. Le théorème de complétude s'étend aux langages non dénombrables et l'esprit de la démonstration est identique. La seule différence est

dans la démonstration de la proposition 2.5. On doit tout d'abord ajouter un ensemble de constantes de chaque sorte, non pas dénombrable, mais de même cardinal que le langage. Ensuite, au lieu d'énumérer les propositions, il est nécessaire de bien les ordonner, en utilisant l'axiome du choix. Enfin, la famille d'ensembles $(\mathcal{U}_i)_i$ n'est plus indexée par les entiers, mais par un ordinal plus grand.

2.3.3 Les modèles égalitaires

Définition 2.7 (Modèle égalitaire)

Soit $\mathcal{L}$ un langage qui contient des prédicats $=_s$ de sorte (s, s) pour certaines sortes s et $\mathcal{T}$ une théorie qui contient au moins les axiomes de l'égalité pour ces sortes. On appelle *modèle égalitaire* de la théorie $\mathcal{T}$ un modèle dans lequel les fonctions $\hat{=}_s$ sont les fonctions définies sur $\mathcal{M}_s$ par $\hat{=}_s(x, y) = 1$ si $x = y$ et $\hat{=}_s(x, y) = 0$ sinon.

Proposition 2.7 (Complétude pour les modèles égalitaires)

Soit une théorie $\mathcal{T}$ contenant au moins les axiomes de l'égalité, alors si $\mathcal{T}$ est cohérente, elle a un modèle égalitaire.

Démonstration. La théorie étant cohérente, elle a un modèle $\mathcal{M}$. Sur les sortes s munies d'un prédicat d'égalité, on définit la relation R_s qui relie a et b quand $\hat{=}(a, b) = 1$. Cette relation est une relation d'équivalence. On pose $\mathcal{M}'_s = \mathcal{M}_s/R_s$. Le modèle $\mathcal{M}$ étant un modèle des axiomes de l'égalité, toutes fonctions $\hat{f}$ et $\hat{P}$ passent au quotient. On définit ainsi un modèle égalitaire $\mathcal{M}'$ qui valide les mêmes propositions que $\mathcal{M}$. C'est donc un modèle de $\mathcal{T}$.

2.3.4 Les démonstrations de cohérence relative

Une application importante du théorème de complétude est qu'il permet de faire des démonstrations de *cohérence relative*. Le modèle V_ω construit à la section 2.2 est un modèle de la théorie ZF^f, c'est-à-dire de la théorie formée des mêmes axiomes que ZF mais dans laquelle l'axiome de l'infini a été remplacé par sa négation. La construction de ce modèle peut se formaliser dans ZF, c'est-à-dire que la proposition « Il existe un modèle de ZF^f » est démontrable dans ZF. D'après le théorème de correction, on peut en déduire que la proposition « La théorie ZF^f est cohérente » est démontrable dans ZF.

Cette situation est cependant exceptionnelle. Si, au lieu de cet exemple élémentaire de ZF^f, on considère les exemples plus intéressants de ZFC ou de $ZF\neg C$ obtenus en ajoutant aux axiomes de ZF respectivement l'axiome du choix et sa négation, alors, une conséquence du second théorème d'incomplétude de Gödel — qui ne sera pas abordé dans ce livre, mais qui montre que sous des conditions assez générales, la cohérence d'une théorie ne peut pas être démontrée dans cette même théorie — est qu'il est impossible de démontrer, dans ZF, la cohérence de ZF et *a fortiori* celle de ZFC ou $ZF\neg C$.

En revanche, il est possible de démontrer dans ZF des théorèmes de cohérence relative. Ainsi, K. Gödel a démontré que si la théorie ZF est cohérente, alors la théorie ZFC également et A. Fraenkel et A. Mostowski que si la théorie ZF est cohérente, alors la théorie $ZF\neg C$ également.

Autrement dit, pour démontrer la cohérence des théories ZFC et $ZF\neg C$, on ajoute l'axiome « La théorie ZF est cohérente » aux mathématiques ordinaires, formalisables dans ZF. Ensuite, ces démonstrations utilisent le théorème de complétude, pour déduire, de la cohérence de ZF, l'existence d'un modèle de ZF, puis elles construisent un modèle de ZFC ou $ZF\neg C$ en utilisant ce modèle.

Exercice 2.3

Dans cet exercice, inspiré de la démonstration de Fraenkel et Mostowski de la cohérence relative de $ZF\neg C$, on montre que si ZF est cohérente, alors ZF^+ est cohérente, où ZF^+ est la théorie obtenue en ajoutant à ZF l'axiome $\exists x\ (x \in x)$. Autrement dit, si ZF est cohérente, alors elle ne démontre pas la proposition $\neg\exists x\ (x \in x)$.

1. Montrer que les propositions suivantes sont équivalentes.
   (a) Si ZF est cohérente alors ZF^+ est cohérente.
   (b) Si ZF a un modèle alors ZF^+ a un modèle.
   (c) Si ZF a un modèle égalitaire alors ZF^+ a un modèle égalitaire.
   On se propose de démontrer la proposition (c). Soit $\mathcal{M} = (\mathcal{M}, \mathcal{C}, \hat{\epsilon}_2, \hat{\in})$ un modèle égalitaire de ZF.
2. Montrer qu'il existe dans $\mathcal{M}$ un élément 0 tel que l'on n'ait $a\ \hat{\in}\ 0$ pour aucun élément a de $\mathcal{M}$. Montrer qu'il existe dans $\mathcal{M}$ un élément 1 tel que pour tout élément a de $\mathcal{M}$, on ait $a\ \hat{\in}\ 1$ si et seulement si $a = 0$.
   Soit f la bijection de $\mathcal{M}$ dans $\mathcal{M}$ définie par $f(0) = 1$, $f(1) = 0$ et $f(a) = a$ si a est distinct de 0 et 1. Soit $\mathcal{M}'$ le modèle $(\mathcal{M}, \mathcal{C}, \hat{\epsilon}_2, \hat{\in}')$ où $\hat{\in}'$ est la relation définie par $a\ \hat{\in}'\ b$ si et seulement si $a\ \hat{\in}\ f(b)$. On veut montrer que $\mathcal{M}'$ est un modèle égalitaire de ZF^+.
3. Montrer qu'il existe une proposition *Zéro* telle que $[\![\mathit{Zéro}]\!]^{\mathcal{M}}_{x=a} = 1$ si et seulement si $a = 0$. Montrer qu'il existe une proposition *Un* telle que

$[\![Un]\!]^{\mathcal{M}}_{x=a} = 1$ si et seulement si $a = 1$. Montrer qu'il existe une proposition F telle que $[\![F]\!]^{\mathcal{M}}_{x=a,y=b} = 1$ si et seulement si $b = f(a)$. Montrer qu'il existe une proposition E telle que $[\![E]\!]^{\mathcal{M}}_{x=a,y=b} = 1$ si et seulement si $a \hat{\in}' b$. Montrer que $\mathcal{M}'$ est un modèle du schéma de compréhension des classes binaires.

4. Montrer que l'axiome d'extensionalité est valide dans $\mathcal{M}'$.
5. Soit a un élément de $\mathcal{M}$. On pose $a_1 = f(a)$. Montrer qu'il existe un élément a_2 de $\mathcal{M}$ tel que $x \hat{\in} a_2$ si et seulement s'il existe un y tel que $x = f(y)$ et $y \hat{\in} a_1$. Montrer qu'il existe un élément a_3 de $\mathcal{M}$ tel que $x \hat{\in} a_3$ si et seulement s'il existe un z tel que $x \hat{\in} z$ et $z \hat{\in} a_2$. Soit $a_4 = f^{-1}(a_3)$. Montrer que $x \hat{\in}' a_4$ si et seulement s'il existe un y tel que $x \hat{\in}' y$ et $y \hat{\in}' a$. Montrer que l'axiome de la réunion est valide dans $\mathcal{M}'$.
6. Soit a un élément de $\mathcal{M}$. On pose $a_1 = f(a)$. Montrer qu'il existe un élément a_2 de $\mathcal{M}$ tel que $x \hat{\in} a_2$ si et seulement si pour tout z, $z \hat{\in} x$ implique $z \hat{\in} a_1$. Montrer qu'il existe un élément a_3 de $\mathcal{M}$ tel que $x \hat{\in} a_3$ si et seulement $f(x) \hat{\in} a_2$. Soit $a_4 = f^{-1}(a_3)$. Montrer que $x \hat{\in}' a_4$ si et seulement si pour tout z, $z \hat{\in}' x$ implique $z \hat{\in}' a$. Montrer que l'axiome de l'ensemble des parties est valide dans $\mathcal{M}'$.
7. Soit a un élément de $\mathcal{M}$ et r un élément de $\mathcal{C}$ qui est une classe binaire fonctionnelle, c'est-à-dire tel que si $a, b \ \hat{\epsilon}_2 \ r$ et $a, b' \ \hat{\epsilon}_2 \ r$ alors $b = b'$. On pose $a_1 = f(a)$. Montrer qu'il existe un élément a_2 de $\mathcal{M}$ tel que $x \hat{\in} a_2$ si et seulement s'il existe un y tel que $y \hat{\in} a_1$ et $y, x \ \hat{\epsilon}_2 \ r$. Soit $a_3 = f^{-1}(a_2)$. Montrer que $x \hat{\in}' a_3$ si et seulement s'il existe un y tel que $y \hat{\in}' a$ et $y, x \ \hat{\epsilon}_2 \ r$. Montrer que l'axiome de remplacement est valide dans $\mathcal{M}'$.
8. Dans cette question on admettra le résultat suivant, démontré à l'exercice 1.18 : *Si a est un élément de $\mathcal{M}$ et r un élément de $\mathcal{C}$ qui est une classe binaire fonctionnelle, alors il existe un élément E de $\mathcal{M}$ tel que $a \hat{\in} E$ et si $x \hat{\in} E$ et $x, x' \ \hat{\epsilon}_2 \ r$ alors $x' \hat{\in} E$.*

   Montrer que il n'existe pas d'objet a tel que $a \hat{\in}' 1$.

   Soit a un élément de $\mathcal{M}$. Soit $S(a)$ l'élément de $\mathcal{M}$ tel que $x \hat{\in} S(a)$ si et seulement si $x \hat{\in} a$ ou $x = a$ et $S'(a)$ l'élément de $\mathcal{M}$ tel que $x \hat{\in}' S'(a)$ si et seulement si $x \hat{\in}' a$ ou $x = a$. Montrer que si a n'est ni 0 ni 1, alors $S'(a) = S(a)$. Quel est l'objet $S'(0)$? Et l'objet $S'(1)$? Montrer que la classe binaire r telle que $a, b \ \hat{\epsilon}_2 \ r$ si $b = S'(a)$ appartient à $\mathcal{C}$ et qu'elle est fonctionnelle.

   Montrer qu'il existe un ensemble I' qui contient 1 et tel que si $a \hat{\in} I'$ alors $S'(a) \hat{\in} I'$.

   Montrer que l'axiome de l'infini est valide dans $\mathcal{M}'$.
9. Montrer que $0 \hat{\in}' 0$. Montrer que la proposition $\exists x \ (x \in x)$ est valide dans $\mathcal{M}'$.

2.3.5 La conservativité

Définition 2.8 (Extension)

Soient $\mathcal{L}$ et $\mathcal{L}'$ deux langages tels que $\mathcal{L} \subseteq \mathcal{L}'$. Soit $\mathcal{T}$ une théorie exprimée dans $\mathcal{L}$ et $\mathcal{T}'$ une théorie exprimée dans $\mathcal{L}'$. La théorie $\mathcal{T}'$ est une *extension* de $\mathcal{T}$ si toutes les propositions démontrables dans $\mathcal{T}$ sont démontrables dans $\mathcal{T}'$.

Définition 2.9 (Extension conservatrice)

Soient $\mathcal{L}$ et $\mathcal{L}'$ deux langages tels que $\mathcal{L} \subseteq \mathcal{L}'$. Soit $\mathcal{T}$ une théorie exprimée dans $\mathcal{L}$ et $\mathcal{T}'$ une théorie exprimée dans $\mathcal{L}'$ qui est une extension de $\mathcal{T}$. La théorie $\mathcal{T}'$ est une extension *conservatrice* de $\mathcal{T}$ si toutes les propositions de $\mathcal{L}$ qui sont démontrables dans $\mathcal{T}'$ sont démontrables dans $\mathcal{T}$.

Par exemple, si le langage $\mathcal{L}$ contient une constante c et un symbole de prédicat P et que la théorie $\mathcal{T}$ est formée de l'axiome $P(c)$, alors, en ajoutant une constante d et l'axiome $P(d)$, on obtient une extension conservatrice : certes la proposition $P(d)$ est démontrable dans $\mathcal{T}'$ alors qu'elle ne l'était pas dans $\mathcal{T}$, mais, comme nous allons le voir, toutes les propositions du langage $\mathcal{L}$ — ce qui n'est pas le cas de $P(d)$ — qui sont démontrables dans $\mathcal{T}'$ le sont également dans $\mathcal{T}$.

En revanche, si le langage $\mathcal{L}$ contient une constante c et un symbole de prédicat P et que la théorie $\mathcal{T}$ est vide, alors, en ajoutant une constante d et l'axiome $P(d)$, on obtient une extension qui n'est pas conservatrice, car la proposition $\exists x\, P(x)$, qui est bien formée dans $\mathcal{T}$, est démontrable dans $\mathcal{T}'$ mais pas dans $\mathcal{T}$.

Si cela est possible dans les cas simples comme celui-ci, il est en général difficile de montrer qu'une extension est conservatrice en montrant directement qu'une démonstration dans $\mathcal{T}'$ se traduit en une démonstration dans $\mathcal{T}$. En revanche le théorème de complétude fournit un outil efficace pour montrer qu'une théorie est une extension conservatrice d'une autre.

Définition 2.10 (Extension d'un modèle)

Soit $\mathcal{L}$ et $\mathcal{L}'$ deux langages tels que $\mathcal{L} \subseteq \mathcal{L}'$. Soient $\mathcal{M}$ un modèle de $\mathcal{L}$ et $\mathcal{M}'$ un modèle de $\mathcal{L}'$. Le modèle $\mathcal{M}'$ est une *extension* de $\mathcal{M}$ si pour toute sorte s de $\mathcal{L}$ on a $\mathcal{M}_s = \mathcal{M}'_s$ et pour tout symbole de fonction ou de prédicat f de $\mathcal{L}$ on a $\hat{f}^{\mathcal{M}} = \hat{f}^{\mathcal{M}'}$.

Proposition 2.8

Soit $\mathcal{L}$ un langage et $\mathcal{T}$ une théorie exprimée dans ce langage. Soit $\mathcal{L}'$ un langage tel que $\mathcal{L} \subseteq \mathcal{L}'$ et $\mathcal{T}'$ une théorie dans $\mathcal{L}'$ telle que $\mathcal{T} \subseteq \mathcal{T}'$. Si pour tout modèle $\mathcal{M}$ de $\mathcal{T}$ il existe une extension $\mathcal{M}'$ de $\mathcal{M}$ qui est un modèle de $\mathcal{T}'$, alors $\mathcal{T}'$ est une extension conservatrice de $\mathcal{T}$.

Démonstration. Soit A une proposition du langage $\mathcal{L}$ qui est démontrable dans $\mathcal{T}'$. Soit $\mathcal{M}$ un modèle quelconque de $\mathcal{T}$, il existe un modèle $\mathcal{M}'$ de $\mathcal{T}'$ qui est une extension de $\mathcal{M}$. Le modèle $\mathcal{M}'$ étant un modèle de $\mathcal{T}'$, la proposition A est valide dans $\mathcal{M}'$, donc sa dénotation dans $\mathcal{M}'$ est 1. Comme $\mathcal{M}'$ est une extension de $\mathcal{M}$ la dénotation de A dans $\mathcal{M}$ est la même, c'est-à-dire 1 également. La proposition A est donc valide dans $\mathcal{M}$. La proposition A étant valide dans tous les modèles de $\mathcal{T}$, elle est démontrable dans $\mathcal{T}$.

Par exemple, si $\mathcal{L}$ contient c et P et que la théorie $\mathcal{T}$ est formé de l'axiome $P(c)$, en ajoutant une constante d et l'axiome $P(d)$ on obtient une extension conservatrice. En effet, un modèle $\mathcal{M}$ de $\mathcal{T}$ s'étend en un modèle de $\mathcal{T}'$ en posant $\hat{d} = \hat{c}$.

Exercice 2.4

À l'exercice 1.6, nous avons montré que les propositions A et les théories $\mathcal{T}$ exprimées dans un langage à plusieurs sortes de termes pouvaient se relativiser en des propositions $|A|$ et des théories $|\mathcal{T}|$ d'un langage à une seule sorte de termes, de manière à ce que, si la proposition close A est démontrable dans $\mathcal{T}$, alors la proposition $|A|$ est démontrable dans $|\mathcal{T}|$.

Montrer que, réciproquement, si $|A|$ est démontrable dans $|\mathcal{T}|$ alors A est démontrable dans $\mathcal{T}$.

Quand on formule l'arithmétique ou la théorie des ensembles on peut éviter d'utiliser la notion de classe, et donc les sortes κ et σ et les symboles ϵ et ϵ_2, en remplaçant chaque axiome qui utilise les classes par un schéma d'axiome, c'est-à-dire une infinité d'axiomes : par exemple, l'axiome de récurrence est remplacé par le schéma de récurrence qui est l'ensemble contenant, pour chaque proposition A, l'axiome

$$\forall x_1 \ldots \forall x_n \; ((0/y)A \Rightarrow \forall m \; ((m/y)A \Rightarrow (S(m)/y)A) \Rightarrow \forall n \; (n/y)A)$$

où x_1, …, x_n sont les variables libres de A distinctes de y. Par exemple, la proposition $y + 0 = y$ donne l'axiome

$$0 + 0 = 0 \Rightarrow \forall m \; (m + 0 = m \Rightarrow S(m) + 0 = S(m)) \Rightarrow \forall n \; (n + 0 = n)$$

On obtient alors la théorie suivante.

Définition 2.11

Le langage de l'arithmétique contient une constante 0, un symbole de fonction unaire S, deux symboles de fonction binaire $+$ et $\times$ et un symbole de prédicat binaire $=$. Aux axiomes de l'égalité, on ajoute les axiomes

$$\forall x \forall y \ (S(x) = S(y) \Rightarrow x = y)$$

$$\forall x \ \neg(0 = S(x))$$

$$\forall x_1 \ldots \forall x_n \ ((0/y)A \Rightarrow \forall m \ ((m/y)A \Rightarrow (S(m)/y)A) \Rightarrow \forall n \ (n/y)A)$$

$$\forall y \ (0 + y = y)$$

$$\forall x \forall y \ (S(x) + y = S(x + y))$$

$$\forall y \ (0 \times y = 0)$$

$$\forall x \forall y \ (S(x) \times y = (x \times y) + y)$$

On peut montrer que la théorie avec une sorte de termes pour les classes est une extension conservatrice de cette théorie et, plus généralement, qu'une théorie qui contient un schéma d'axiome a une formulation alternative dans la théorie des classes en remplaçant ce schéma par un axiome unique. La notion de schéma d'axiome étant un peu lourde à définir dans le cas général, on montre ce résultat uniquement pour le cas de l'arithmétique.

Proposition 2.9

La formulation de l'arithmétique de la définition 1.33 est une extension conservatrice de celle de la définition 2.11.

Démonstration. Chaque instance du schéma de récurrence peut se démontrer dans la théorie de la définition 1.33. Cette théorie est donc une extension de celle de la définition 2.11. Pour montrer que cette extension est conservatrice, on montre que tout modèle de la théorie de la définition 2.11 s'étend en un modèle de la théorie de la définition 1.33.

Soit $(\mathcal{M}, \hat{0}, \hat{S}, \hat{+}, \hat{\times}, \hat{=})$ un modèle de la théorie de la définition 2.11. Une partie E de $\mathcal{M}$ est dite *définissable dans l'arithmétique* s'il existe une proposition A dans le langage de la théorie de la définition 2.11, dont les variables libres sont parmi $x_1, \ldots, x_n, y$ et des éléments $a_1, \ldots, a_n$ de $\mathcal{M}$ tels que b appartienne à E si et seulement si

$$[\![A]\!]_{x_1=a_1,\ldots,x_n=a_n,y=b} = 1$$

Soit $\overline{\wp}(\mathcal{M})$ l'ensemble des parties définissables de $\mathcal{M}$. On étend le modèle $\mathcal{M}$ en posant $\mathcal{M}_\kappa = \overline{\wp}(\mathcal{M})$ et $\hat{\epsilon}(b, E) = 1$ si b est un élément de E et 0 sinon.

Le modèle ainsi construit est un modèle de schéma de compréhension. On montre que c'est un modèle de l'axiome de récurrence

$$\forall c\ (0\ \epsilon\ c \Rightarrow \forall m\ (m\ \epsilon\ c \Rightarrow S(m)\ \epsilon\ c) \Rightarrow \forall n\ n\ \epsilon\ c)$$

Pour cela, on considère un élément arbitraire E de $\mathcal{M}_\kappa$ et on montre que

$$[\![(0\ \epsilon\ c \Rightarrow \forall m\ (m\ \epsilon\ c \Rightarrow S(m)\ \epsilon\ c) \Rightarrow \forall n\ n\ \epsilon\ c)]\!]_{c=E} = 1$$

L'ensemble E est une partie définissable de $\mathcal{M}$. Soient A et $a_1, \ldots, a_n$ une proposition et des éléments de $\mathcal{M}$ définissant E. Le modèle $\mathcal{M}$ est un modèle de l'instance du schéma de récurrence correspondant à la proposition A et donc

$$[\![(0/y)A \Rightarrow \forall m\ ((m/y)A \Rightarrow (S(m)/y)A) \Rightarrow \forall n\ (n/y)A]\!]_{x_1=a_1,\ldots,x_n=a_n} = 1$$

Comme les dénotations des propositions $t\ \epsilon\ c$ et $(t/y)A$ sont identiques dans une valuation dans laquelle $\phi c = E$, $\phi x_1 = a_1\ \ldots$, $\phi x_n = a_n$, on en déduit que

$$[\![(0\ \epsilon\ c \Rightarrow \forall m\ (m\ \epsilon\ c \Rightarrow S(m)\ \epsilon\ c) \Rightarrow \forall n\ n\ \epsilon\ c)]\!]_{c=E} = 1$$

Exercice 2.5

Donner une formulation de la théorie ZF avec un schéma d'axiome. Montrer que la théorie formulée avec des classes binaires est une extension conservatrice de cette théorie.

Quand on formule l'axiome des parties comme à la définition 1.36

$$\forall x \exists z \forall w\ (w \in z \Leftrightarrow (\forall v\ (v \in w \Rightarrow v \in x)))$$

on peut montrer que si A est un ensemble, il existe un ensemble qui contient les parties de A, mais on ne dispose pas d'une notation, comme $\wp(A)$, pour désigner cet ensemble.

Une alternative est d'introduire un symbole de fonction $\wp$ et l'axiome

$$\forall x \forall w\ (w \in \wp(x) \Leftrightarrow (\forall v\ (v \in w \Rightarrow v \in x)))$$

On peut montrer que la théorie ainsi obtenue est une extension conservatrice de la théorie des ensembles.

Théorème 2.3 (Skolem)

Soit une théorie $\mathcal{T}$ et A une proposition de la forme $\forall x_1 \ldots \forall x_n \exists y\ B$, démontrable dans $\mathcal{T}$, alors la théorie obtenue en ajoutant un symbole de fonction f et l'axiome $\forall x_1 \ldots \forall x_n\ (f(x_1, \ldots, x_n)/y)B$ est une extension conservatrice de $\mathcal{T}$.

Démonstration. Soit $\mathcal{M}$ un modèle de $\mathcal{T}$. On montre que ce modèle s'étend en un modèle de cet axiome. Soit $a_1, \ldots, a_n$ des éléments quelconques de $\mathcal{M}$, il existe un élément b de $\mathcal{M}$ tel que

$$[\![B]\!]_{x_1=a_1,\ldots,x_n=a_n,y=b} = 1$$

pour chaque n-uplet $a_1, \ldots, a_n$, on choisit un tel élément b et on définit $\hat{f}$ comme la fonction qui, à chaque n-uplet $a_1, \ldots, a_n$, associe l'élément b correspondant.

2.4 D'autres usages de la notion de modèle

Dans ce chapitre, nous avons montré que la notion de modèle permettait de démontrer de nombreuses propriétés des démonstrations, par exemple des propriétés d'indépendance, de cohérence, de cohérence relative et de conservativité. Cependant, en mathématiques et en informatique, cette notion a bien d'autres usages que celui d'être un outil pour étudier les démonstrations. Nous donnons, pour finir, quelques exemples, dans lesquels les notions de modèle et de langage sont utilisées, sans que le but soit d'étudier les démonstrations.

2.4.1 Les structures algébriques

Faire des démonstrations dans la théorie formée des axiomes de l'égalité et des axiomes

$$\forall x \forall y \forall z\ ((x + y) + z = x + (y + z))$$

$$\forall x\ (x + 0 = x \wedge 0 + x = x)$$

$$\forall x \exists y\ (x + y = 0 \wedge y + x = 0)$$

n'a pas grand intérêt. En revanche, les modèles égalitaires de cette théorie sont intéressant en eux-mêmes : ce sont les groupes. Démontrer des propriétés des modèles de cette théorie donnera donc des résultats de théorie des groupes. Donnons un exemple.

Théorème 2.4 (Löwenheim-Skolem)

Soit $\mathcal{L} = (\mathcal{S}, \mathcal{F}, \mathcal{P})$ un langage fini ou dénombrable, $\mathcal{T}$ une théorie dans ce langage et κ un ensemble infini quelconque. Si la théorie $\mathcal{T}$ a un modèle infini, elle a un modèle de même cardinal que κ.

Démonstration. Ce théorème est une simple conséquence du théorème de complétude en cardinalité quelconque.

Soit le langage $(\mathcal{S}, \mathcal{F} \uplus \kappa, \mathcal{P} \uplus \{=\})$ obtenu en ajoutant au langage de départ un symbole $=$ et une constante pour chaque élément de κ et la théorie $\mathcal{T}'$ obtenue en ajoutant à $\mathcal{T}$ les axiomes de l'égalité et les axiomes $\neg a = b$ pour tout couple (a, b) d'éléments distincts de κ. Soit $\mathcal{M}$ un modèle infini de la théorie $\mathcal{T}$.

Il n'est pas difficile de montrer que tout sous-ensemble fini de $\mathcal{T}'$ est cohérent : un sous-ensemble fini de $\mathcal{T}'$ n'utilise qu'un nombre fini de constantes de κ, il suffit d'étendre le modèle $\mathcal{M}$ en associant à ces constantes des éléments distincts de $\mathcal{M}$, ce qui est possible car $\mathcal{M}$ est infini, et en associant à toutes les autres constantes de κ un élément quelconque.

On en déduit que la théorie $\mathcal{T}'$ elle-même est cohérente. En effet, si elle ne l'était pas, il existerait un sous-ensemble fini Γ de $\mathcal{T}'$, tel que le séquent $\Gamma \vdash \bot$ soit démontrable, ce qui est contradictoire puisque tous les sous-ensembles finis de $\mathcal{T}'$ sont cohérents.

Soit $\mathcal{M}'$ le modèle égalitaire de $\mathcal{T}'$ construit dans la démonstration du théorème de complétude. Ce modèle a au moins autant d'éléments que κ puisque les éléments de κ sont associés à des éléments différents. Il est facile de démontrer que, l'ensemble κ étant infini, il y a autant de termes clos du langage $(\mathcal{S}, \mathcal{F} \uplus \kappa, \mathcal{P} \uplus \{=\})$ que d'éléments dans κ et donc que le modèle construit a au plus autant d'éléments que κ. Il a donc exactement le même cardinal que κ.

On en déduit le corollaire suivant.

Proposition 2.10

Il existe des groupes de toute cardinalité infinie. Tout ensemble infini peut être muni d'une structure de groupe.

Ce théorème, qui ne mentionne pas la notion de modèle ou de théorie, est donc un résultat de théorie des groupes, obtenu comme corollaire d'un résultat de logique.

Le théorème de Löwenheim-Skolem a aussi comme conséquence l'existence de modèles égalitaires non dénombrables de l'arithmétique, et plus généralement de toute théorie qui a $\mathbb{N}$ comme modèle. Ce résultat peut surprendre, car

tous les modèles égalitaires de l'arithmétique semblent, à première vue, avoir le même cardinal que $\mathbb{N}$. En effet, considérons un modèle égalitaire $\mathcal{M}$ de l'arithmétique et l'application F de $\mathbb{N}$ dans $\mathcal{M}$ qui à n associe $\hat{S}^n(\hat{0})$. L'image I de cette fonction contient $\hat{0}$, elle est close par $\hat{S}$ et $\mathcal{M}$ est un modèle de l'axiome de récurrence. Il semble donc que tous les éléments de $\mathcal{M}$ appartiennent à I et que $\mathcal{M}$ est donc au plus dénombrable. Où l'erreur se trouve-t-elle ?

L'erreur vient de ce que nous avons abusivement considéré que dans un modèle $\mathcal{M}$ de l'axiome de récurrence, tout ensemble I qui contient $\hat{0}$ et qui est clos par $\hat{S}$ contient l'ensemble $\mathcal{M}$ en entier, alors que cela n'est vrai que quand I appartient à l'ensemble $\mathcal{M}_\kappa$. Or, le schéma d'axiome de compréhension nous impose que $\mathcal{M}_\kappa$ contienne les sous-ensembles de $\mathcal{M}$ qui sont définissables par une proposition A et cela n'est pas le cas de l'ensemble I. Autrement dit, parmi la multitude non dénombrable de parties de $\mathbb{N}$, le schéma de compréhension pose l'existence du petit nombre — dénombrable — d'ensembles définissables et l'axiome de récurrence affirme que si l'un de ces ensembles là contient 0 et est clos par successeur, alors il contient tous les entiers. Ce qui laisse un important degré de liberté.

Définition 2.12 (Modèle standard)

Un modèle *standard* de la théorie des classe est un modèle dans lequel $\mathcal{M}_\kappa = \wp(\mathcal{M}_\iota)$.

Exercice 2.6

Montrer que tous les modèles standards de l'arithmétique ont même cardinal que $\mathbb{N}$.

De même, alors que l'on sait que tous les corps totalement ordonnés, archimédiens et complets sont isomorphes à $\mathbb{R}$, c'est-à-dire que tous les modèles standards de la théorie des corps totalement ordonnés, archimédiens et complets sont isomorphes à $\mathbb{R}$, il existe des modèles non standards de cette théorie qui sont dénombrables.

Ce concept de modèle standard est essentiel pour les applications de la théorie des modèles en algèbre. En revanche, il a un intérêt limité quand on utilise la notion de modèle pour étudier les démonstrations car, d'après le théorème de Löwenheim-Skolem, il n'y a pas de théorie exprimée dans un langage fini ou dénombrable dont tous les modèles soient standards.

2.4.2 La définissabilité

Une deuxième application de la notion de modèle est la définition de la notion d'ensemble, et plus généralement de relation, *définissable*.

Définition 2.13

Soit $\mathcal{M}$ un ensemble et $R_1, \ldots, R_n$ des relations sur cet ensemble. On dit qu'une relation S sur $\mathcal{M}$ est *définissable* dans la structure $(\mathcal{M}, R_1, \ldots, R_n)$ s'il existe une proposition A, dans le langage formé des symboles $P_1, \ldots, P_n$ et contenant des variables libres $x_1, \ldots, x_p$, telle que les éléments $a_1, \ldots, a_p$ soient reliés par S si et seulement si

$$[\![A]\!]_{x_1=a_1,\ldots,x_n=a_n} = 1$$

dans le modèle $(\mathcal{M}, R_1, \ldots, R_n)$.

Si R_1 et R_2 sont deux relations binaires, l'intersection de ces deux relations est définissable à partir de R_1 et R_2, par la proposition $P_1(x,y) \wedge P_2(x,y)$. Plus généralement, si deux relations de même arité sont définissables, leur intersection l'est également. L'ensemble des relations définissables est donc clos par intersection. Cet ensemble est également clos par réunion et complémentation. Cependant l'ensemble des relations définissables contient bien plus d'éléments que l'ensemble inductivement défini comme le plus petit ensemble contenant $R_1, \ldots, R_n$ et clos par intersection, réunion et complémentation. En effet, la possibilité d'utiliser plusieurs fois la même variable et de permuter les variables permet, par exemple, de définir l'ensemble des objets en relation avec eux-mêmes, $P(x,x)$, ou la relation réciproque d'une relation, $P(y,x)$, mais c'est surtout l'utilisation des quantificateurs qui donne toute sa richesse à cet ensemble, puisqu'il devient possible de définir, par exemple, la composée de deux relations, $\exists z\ (P_1(x,z) \wedge P_2(z,y))$.

Cependant, toutes les relations ne sont pas définissables. On peut, par exemple, démontrer que la clôture réflexive-transitive d'une relation n'est pas définissable à partir de cette relation.

En théorie des bases de données, les relations définissables correspondent aux requêtes exprimables.

Deuxième partie

Les algorithmes

3
Les fonctions calculables

Nous nous intéressons dans ce chapitre à des algorithmes qui prennent en argument des entiers $p_1, \ldots, p_n$ et calculent un entier q. À chacun de ces algorithmes, on peut associer la fonction de $\mathbb{N}^n$ dans $\mathbb{N}$, qui, aux entiers $p_1, \ldots, p_n$, associe l'entier q. Une telle fonction, associée à un algorithme, est dite *calculable*. Cette notion de fonction calculable, qui abstrait et simplifie celle d'algorithme, se révèle être la notion importante quand on étudie les limites du calcul. Avant d'introduire, au chapitre 4, une vision plus opérationnelle du calcul, nous commençons, dans ce chapitre, par cette notion de fonction calculable.

3.1 Les fonctions calculables

Soit F_n l'ensemble des fonctions partielles de $\mathbb{N}^n$ dans $\mathbb{N}$ et F la réunion des F_n.

Définition 3.1 (Fonction calculable)

L'ensemble des *fonctions calculables* est le sous-ensemble de F inductivement défini comme le plus petit ensemble contenant

– les projections

$$x_1, \ldots, x_n \mapsto x_i$$

– les fonctions identiquement nulles

$$x_1, \ldots, x_n \mapsto 0$$

– la fonction successeur

$$x \mapsto x + 1$$

et clos par

– la composition, c'est-à-dire l'opération associant, à h de $\mathbb{N}^m$ dans $\mathbb{N}$ et $g_1, \ldots, g_m$ de $\mathbb{N}^n$ dans $\mathbb{N}$, la fonction de $\mathbb{N}^n$ dans $\mathbb{N}$

$$x_1, \ldots, x_n \mapsto h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

– la définition par récurrence, c'est-à-dire l'opération associant à g de $\mathbb{N}^{n-1}$ dans $\mathbb{N}$ et h de $\mathbb{N}^{n+1}$ dans $\mathbb{N}$, la fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ définie par

$$f(x_1, \ldots, x_{n-1}, 0) = g(x_1, \ldots, x_{n-1})$$

$$f(x_1, \ldots, x_{n-1}, y+1) = h(x_1, \ldots, x_{n-1}, y, f(x_1, \ldots, x_{n-1}, y))$$

– et la minimisation, c'est-à-dire l'opération associant à g de $\mathbb{N}^{n+1}$ dans $\mathbb{N}$ la fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ telle que $f(x_1, \ldots, x_n)$ soit le plus petit entier y tel que $g(x_1, \ldots, x_n, y) = 0$.

Précisons les ensembles de définition de ces fonctions. Les projections, les fonctions identiquement nulles et la fonction successeur sont totales. La fonction $x_1, \ldots, x_n \mapsto h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ est définie en $p_1, \ldots, p_n$ si les fonctions $g_1, \ldots, g_m$ sont toutes définies en $p_1, \ldots, p_n$ et si la fonction h est définie en $g_1(p_1, \ldots, p_n), \ldots, g_m(p_1, \ldots, p_n)$. Une fonction f définie par récurrence à partir des fonctions g et h est définie en $p_1, \ldots, p_{n-1}, 0$ si g est définie en $p_1, \ldots, p_{n-1}$ et elle est définie en $p_1, \ldots, p_{n-1}, q+1$ si elle est définie en $p_1, \ldots, p_{n-1}, q$ et si h est définie en $p_1, \ldots, p_{n-1}, q, f(p_1, \ldots, p_{n-1}, q)$. La fonction f définie par minimisation d'une fonction g est définie en $p_1, \ldots, p_n$ s'il existe un entier q tel que g soit définie et non nulle en $p_1, \ldots, p_n, r$ pour tout r strictement inférieur à q et si elle est définie et nulle en $p_1, \ldots, p_n, q$.

C'est cette dernière règle qui introduit la partialité. Les trois premières règles ne permettent de construire que des fonctions totales et les deux suivantes préservent cette totalité. En revanche, la minimisation transforme la fonction totale $x, y \mapsto x$ en une fonction partielle, qui prend la valeur 0 en 0, mais n'est définie nulle part ailleurs. En effet, si p est un entier non nul, il n'existe pas de q tel que $(x, y \mapsto x)(p, q) = 0$.

Comme nous le verrons à la section 3.4.2 et au chapitre 4, le fait qu'une fonction f ne soit pas définie en un entier p peut s'interpréter comme le fait que le programme qui calcule la fonction f en p ne termine pas. En effet, quand

une fonction f est définie par minimisation d'une fonction g, on calcule $f(p)$ en calculant successivement la valeur de g en $(p,0)$, en $(p,1)$, en $(p,2)$, ... jusqu'à trouver une valeur d'annulation, s'il en existe une. S'il n'en existe pas, la recherche d'une valeur d'annulation se poursuit indéfiniment.

Proposition 3.1 (La fonction prédécesseur)

La fonction *prédécesseur*, définie par $f(n+1) = n$ et $f(0) = 0$ est calculable.

Démonstration. Elle est définie par récurrence.

Proposition 3.2 (Les quatre opérations)

L'addition, la multiplication, la « soustraction » définie par $n \dot{-} p = n - p$ si $n \geq p$ et $n \dot{-} p = 0$ sinon, le quotient et le reste de la division euclidienne sont calculables.

Démonstration. Les fonctions $+$, $\dot{-}$ et $\times$ se définissent par récurrence. Le quotient se définit à partir de ces opérations avec la minimisation et le reste à partir du quotient, de la multiplication et de la soustraction.

Proposition 3.3 (La fonction $\chi_{\leq}$)

La fonction caractéristique de la relation d'ordre $\chi_{\leq}$, définie par $\chi_{\leq}(x,y) = 1$ si $x \leq y$ et $\chi_{\leq}(x,y) = 0$ sinon, est calculable.

Démonstration. $\chi_{\leq}(x,y) = 1 \dot{-} (x \dot{-} y)$.

Proposition 3.4 (Le test)

Soient f, g et h trois fonctions calculables. La fonction i définie sur l'intersection des trois domaines de définition de f, g et h par $i(x_1,\ldots,x_n) = g(x_1,\ldots,x_n)$ si $f(x_1,\ldots,x_n) = 0$ et $i(x_1,\ldots,x_n) = h(x_1,\ldots,x_n)$ sinon est calculable.

Démonstration. La fonction qui à trois entiers p,q,r associe q si $p = 0$ et r sinon peut être définie par récurrence

$$k(0,q,r) = q$$
$$k(p+1,q,r) = r$$

et la fonction i peut être définie par composition à partir de f, g, h et k.

Définition 3.2 (Ensemble décidable, semi-décidable)

Une partie A de $\mathbb{N}$ est dite *décidable* si sa fonction caractéristique est calculable, c'est-à-dire s'il existe une fonction calculable f telle que $f(x) = 1$ si $x \in A$ et $f(x) = 0$ sinon.

Elle est dite *semi-décidable* s'il existe une fonction calculable f telle que $f(x) = 1$ si $x \in A$ et f n'est pas définie en x sinon.

Exercice 3.1

Montrer que tout ensemble décidable est semi-décidable.

Définition 3.3 (Les fonctions récursives primitives)

L'ensemble des fonctions *récursives primitives* est inductivement défini comme le plus petit ensemble de fonctions contenant les projections, les fonctions identiquement nulles, la fonction successeur et clos par composition et par définition par récurrence.

Exercice 3.2

La *fonction d'Ackermann* est définie par $A_0(x) = 2^x$ et $A_{n+1}(x) = \underbrace{A_n \circ \ldots \circ A_n}_{x \text{ fois}}(1)$.

C'est-à-dire

$$A_0(x) = 2^x$$
$$A_{n+1}(0) = 1$$
$$A_{n+1}(x+1) = A_n(A_{n+1}(x))$$

1. Montrer que, pour tout i et pour tout x, $A_i(x) \geq x + 1$.
2. Montrer que, pour tout i, la fonction $x \mapsto A_i(x)$ est strictement croissante.
3. Montrer que, pour tout x, la fonction $i \mapsto A_i(x)$ est croissante.
4. Montrer que, pour tout x, $A_0(x) \geq 2x$ et, si $x \geq 2$, alors $A_0(x) \geq x + 2$. Montrer que, pour tout i et pour tout x, $A_i(x) \geq 2x$ et, si $x \geq 2$, alors $A_i(x) \geq x + 2$.
5. Montrer que, si $x \geq 2$, alors $A_{i+1}(x + 2) \geq A_i(A_i(x + 2))$. Montrer que, si $x \geq 4$, alors $A_{i+1}(x) \geq A_i(A_i(x))$.
6. On dit qu'une fonction f d'arité n est *dominée* par une fonction unaire g si pour tout $x_1, \ldots, x_n$, $f(x_1, \ldots, x_n) \leq g(max(x_1, \ldots, x_n, 4))$.
   Montrer que les projections, les fonctions identiquement nulles et la fonction successeur sont toutes dominées par la fonction A_0, c'est-à-dire par la fonction $x \mapsto 2^x$.

7. Soient $g_1, \ldots, g_m$ et h et des fonctions respectivement dominées par les fonctions $A_{i_1}, \ldots, A_{i_m}$ et A_j. Soit k le plus grand des éléments $i_1, \ldots, i_m$ et j. Montrer que la composée de h et $g_1, \ldots, g_m$ est dominée par A_{k+1}.
8. Soient g et h des fonctions dominées par A_i et A_j et soit k le plus grand élément de i et j. Soit f la fonction définie par récurrence à partir de g et h. Montrer que $f(x_1, \ldots x_{n-1}, y) \leq A_{k+1}(y + max(x_1, \ldots, x_{n-1}, 4))$. Montrer que $f(x_1, \ldots, x_{n-1}, y) \leq A_{k+1}(2\, max(x_1, \ldots, x_{n-1}, y, 4))$. Montrer que $f(x_1, \ldots, x_{n-1}, y) \leq A_{k+1}(A_{k+1}(max(x_1, \ldots, x_{n-1}, y, 4)))$.
   Montrer que f est dominée par A_{k+2}.
9. Montrer que, pour toute fonction récursive primitive f, il existe un entier i tel que f soit dominée par A_i.
10. Montrer que la fonction $i \mapsto A_i(i)$ n'est pas récursive primitive.
11. Montrer que la fonction d'Ackermann n'est pas récursive primitive.

3.2 La calculabilité sur les listes et les arbres

3.2.1 La calculabilité sur les listes

La notion de fonction calculable introduite à la définition 3.1 suppose que les données avec lesquelles on calcule sont des entiers.

On peut étendre cette notion aux listes d'entiers. Pour cela, on commence par associer un entier $\ulcorner l \urcorner$ à chaque liste l, son *numéro*, puis on dit qu'une fonction qui à une liste l associe une liste $F(l)$ est *calculable* si la fonction qui au numéro de l associe le numéro de $F(l)$, qui est une fonction des entiers dans les entiers, est calculable.

Pour associer ainsi un entier à une liste, on utilise la fonction suivante.

Définition 3.4

La fonction ; est définie par

$$p; q = (p + q)(p + q + 1)/2 + p + 1$$

Proposition 3.5

La fonction ; est une bijection de $\mathbb{N}^2$ dans $\mathbb{N}^*$.

Démonstration. Soit n un entier non nul. Soit k le plus grand entier tel que $k(k+1)/2 \leq n-1$ et $p = n - 1 - k(k+1)/2$. De la maximalité de k, on tire

$n-1 < (k+1)(k+2)/2$ et donc $p < (k+1)(k+2)/2 - k(k+1)/2 = k+1$, soit $p \leq k$. On pose $q = k-p$ et on a

$$n = k(k+1)/2 + p + 1 = (p+q)(p+q+1)/2 + p + 1 = p;q$$

La fonction ; est donc surjective.

Si maintenant $p;q = p';q'$, alors soit k le plus grand entier tel que $k(k+1)/2 \leq (p;q) - 1$. On a $k(k+1)/2 \leq (p;q) - 1 = (p+q)(p+q+1)/2 + p < (p+q)(p+q+1)/2+p+q+1 = (p+q+1)(p+q+2)/2$, et donc $k < p+q+1$, c'est-à-dire $k \leq p+q$. On a également $(p+q)(p+q+1)/2 \leq (p+q)(p+q+1)/2+p = (p;q) - 1 < (k+1)(k+2)/2$ et donc $p+q < k+1$, c'est-à-dire $p+q \leq k$. On en déduit $p+q = k$. De même, on a $p'+q' = k$ et donc $p'+q' = p+q$. Comme $(p+q)(p+q+1)/2 + p + 1 = (p'+q')(p'+q'+1)/2 + p' + 1$, on en déduit $p = p'$ et donc $q = q'$. La fonction ; est donc injective.

Définition 3.5

On appelle *hd*, *head*, la fonction qui à un entier non nul n associe l'unique entier p tel qu'il existe q tel que $n = p;q$ et qui à 0 associe 0 et *tl*, *tail*, la fonction qui à un entier non nul n associe l'unique entier q tel qu'il existe p tel que $n = p;q$ et qui à 0 associe 0.

Proposition 3.6

Les fonctions ;, *hd* et *tl* sont calculables.

Démonstration. La fonction ; est définie à partir des quatre opérations et les deux fonctions réciproques à partir des quatre opérations, de la fonction caractéristique de la relation d'ordre et de la minimisation. Elles sont donc calculables.

Proposition 3.7

Si $n \neq 0$, alors $hd(n) < n$ et $tl(n) < n$.

Démonstration. $p;q > p$ et $p;q > q$.

Définition 3.6 (La numérotation des listes d'entiers)

À chaque liste d'entiers l, on associe un *numéro* de la manière suivante

$$\ulcorner p_1, \ldots, p_n \urcorner = p_1; (p_2; (\ldots; (p_n; 0) \ldots))$$

Proposition 3.8

La fonction *sub*, qui, à deux entiers p et n, associe le numéro de l'unique liste de numéro p, privée de ses n premiers éléments, si cette liste a au moins n éléments, et le numéro de la liste vide, c'est-à-dire 0, sinon, est calculable.

Démonstration. Cette fonction est définie par

$$sub(p, 0) = p$$

$$sub(p, n + 1) = tl(sub(p, n))$$

Proposition 3.9

La fonction *length*, qui, à un entier p, associe la longueur de la liste de numéro p, est calculable.

Démonstration. La longueur d'une liste p est le plus petit entier n tel que $sub(p, n) = 0$.

Proposition 3.10

La fonction *nth*, qui, à deux entiers n et p, associe le n-ième élément de la liste de numéro p, si cet élément existe, et l'entier 0 sinon, est calculable.

Démonstration. Cette fonction est définie par $nth(p, n) = hd(sub(p, n))$.

Le schéma de définition par récurrence de la définition 3.1 peut sembler, à première vue, restrictif car, pour définir la valeur de h en $y + 1$, on peut utiliser uniquement la valeur de h en y et non toutes les valeurs de h en z, pour $z < y + 1$. Ainsi, il n'est pas immédiat que la fonction de Fibonacci définie par récurrence bien fondée par $f(0) = f(1) = 1$ et $f(n + 2) = f(n) + f(n + 1)$ soit calculable.

Cependant, il n'est pas difficile de montrer que, si une fonction f est ainsi définie par récurrence bien fondée, la fonction F qui à x associe $\ulcorner f(0), f(1), \ldots, f(x) \urcorner$ peut être définie par récurrence ordinaire et donc que F, et donc f, sont calculables.

3.2.2 La calculabilité sur les arbres

Il est également utile d'étendre la notion de calculabilité aux expressions d'un langage et plus généralement aux arbres, car cela nous permettra, entre autres exemples, de calculer avec des programmes, des propositions et des démonstrations.

Pour étendre ainsi la notion de fonction calculable, on commence par associer un entier à chaque arbre, son *numéro*, puis on dit qu'une fonction qui à un arbre t associe un arbre $F(t)$ est *calculable* si la fonction qui au numéro de t associe le numéro de $F(t)$, qui est une fonction des entiers dans les entiers, est calculable.

Définition 3.7 (La numérotation des arbres)

Soit E un ensemble muni d'une injection dans $\mathbb{N}$ qui, à chaque élément f de E, associe son *numéro* $\ulcorner f \urcorner$. On associe, à chaque arbre t, étiqueté par des éléments de E, un numéro $\ulcorner t \urcorner$, par récurrence structurelle, de la manière suivante

$$\ulcorner f(t_1, \ldots, t_n) \urcorner = \ulcorner f \urcorner ; (\ulcorner t_1 \urcorner ; (\ulcorner t_2 \urcorner ; \ldots (\ulcorner t_n \urcorner ; 0) \ldots))$$

On peut ainsi numéroter tous les arbres étiquetés par les éléments d'un ensemble fini E : il suffit pour cela d'associer un entier quelconque à chaque élément de E. Et on montre que l'ensemble des fonctions calculables est indépendant de la numérotation choisie des éléments de l'ensemble E.

Mais bien souvent l'ensemble E est infini et est lui-même un ensemble d'arbres. Ainsi, les démonstrations sont des arbres étiquetés par des séquents, qui sont eux-mêmes des arbres étiquetés par des propositions, qui sont elles-mêmes des arbres étiquetés par des variables, des symboles de fonction et des symboles de prédicat, qui sont eux-mêmes des arbres étiquetés par les éléments d'un ensemble fini. L'ensemble des démonstrations est donc un ensemble d'arbres articulé.

Il suffit alors d'associer un entier arbitraire à chaque élément de cet ensemble fini et la définition 3.7 permet de numéroter successivement les variables, les symboles de fonction et les symboles de prédicat, puis les propositions, puis les séquents et enfin les démonstrations.

On peut procéder de même pour tous les ensembles d'arbres articulés. Et on montre que l'ensemble des fonctions calculables est indépendant de la numérotation choisie des symboles de l'ensemble fini de départ.

Proposition 3.11

Les fonctions qui au numéro p d'un arbre a associent l'étiquette de sa racine, le nombre d'enfants de sa racine, et qui à p et i associe le numéro du i-ième sous-arbre immédiat de a sont calculables.

Démonstration. Si p est le numéro de l'arbre a, l'étiquette de la racine de a est $hd(p)$, le nombre d'enfants de la racine de a est $length(tl(p))$, le numéro du i-ième sous-arbre immédiat est $nth(tl(p), i)$.

Proposition 3.12

La fonction qui, à l'entier p, associe le numéro $\ulcorner S^p(0) \urcorner$ de l'arbre $S^p(0)$ est calculable. La fonction qui, au numéro $\ulcorner S^p(0) \urcorner$ de l'arbre $S^p(0)$, associe l'entier p et aux entiers, qui ne sont pas de la forme $\ulcorner S^p(0) \urcorner$, associe 0, est calculable.

Démonstration. La première fonction est définie par récurrence, la seconde par récurrence bien fondée.

3.2.3 Les dérivations

Définition 3.8 (Règles effectives)

Soit E un ensemble d'arbres articulé et $f_1, f_2, \ldots$ des règles sur l'ensemble E. L'ensemble de règles $f_1, f_2, \ldots$ est *effectif* si l'ensemble G des numéros des listes $b, a_1, \ldots, a_n$ tels qu'il existe une règle f_i telle que $b = f_i\ a_1\ \ldots\ a_n$ est décidable, c'est-à-dire si la réunion des graphes des fonctions $f_1, f_2, \ldots$ est un ensemble décidable.

Proposition 3.13

Soit E un ensemble d'arbres articulé et $f_1, f_2, \ldots$ un ensemble de règles effectif. Alors, l'ensemble des dérivations dans $f_1, f_2, \ldots$ est décidable.

Démonstration. On montre qu'il existe une fonction calculable g qui prend en argument une liste d'arbres et retourne 1 ou 0 selon que tous ces arbres sont des dérivations ou non. On commence par remarquer que la fonction qui à une liste d'arbres associe la liste des racines de ces arbres est calculable, car elle se définit par récurrence bien fondée à partir de fonctions calculables. Soit l une liste d'arbres. Si l est la liste vide, alors on pose $g(l) = 1$. Sinon, soit $a = hd(l)$ le premier arbre de cette liste et $l' = tl(l)$ la liste formée des autres arbres de

cette liste. Soit $r = hd(a)$ la racine de a et $l'' = tl(a)$ la liste des sous-arbres immédiats de a et s la liste des racines des arbres de l''. Alors si $r; s$ appartient à G, $g(l') = 1$ et $g(l'') = 1$ on pose $g(l) = 1$ sinon on pose $g(l) = 0$. La fonction g se définit donc par récurrence bien fondée à partir de fonctions calculables, car, d'après la proposition 3.7, les numéros des listes l' et l'' sont des entiers strictement inférieurs à celui de l. Elle est donc calculable.

L'ensemble A, inductivement défini par les règles $f_1, f_2, \ldots$, en revanche, n'est pas toujours décidable. Toutefois, il est semi-décidable. Soit x un élément de E, on peut énumérer tous les arbres l'un après l'autre, et tester si chacun est une dérivation dont la racine est x, ou non. Si x est un élément de A, alors une telle dérivation finira bien par se présenter. Sinon, le processus d'énumération se poursuivra à l'infini.

Proposition 3.14 (Les ensembles d'arbres inductivement définis)

Soit E un ensemble d'arbres articulé et $f_1, f_2, \ldots$ un ensemble de règles effectif sur l'ensemble E. Alors, le sous-ensemble A de E inductivement défini par ces règles est semi-décidable.

Démonstration. Soit $g(x, y)$ la fonction calculable telle que $g(x, y) = 1$ si x est le numéro d'une dérivation d'un élément de E et la racine de x est y et $g(x, y) = 0$ sinon. La fonction h définie par $h(y)$ est le plus petit entier x tel que $1 \dot{-} g(x, y) = 0$ composée avec la fonction constante égale à 1 est un algorithme de semi-décision pour A. Si y appartient à A, alors $h(y) = 1$, sinon h n'est pas définie en y.

3.3 L'élimination de la récurrence

L'ensemble des fonctions calculables a une définition alternative, qui, bien que moins naturelle que la définition 3.1, sera utile dans la suite de ce livre.

Définition 3.9

L'ensemble $\mathcal{C}$ est inductivement défini comme le plus petit ensemble contenant
- les projections,
- les fonctions identiquement nulles,
- la fonction successeur,
- l'addition,

- la multiplication et
- la fonction caractéristique de la relation d'ordre $\chi_{\leq}$, définie par $\chi_{\leq}(x, y) = 1$ si $x \leq y$ et $\chi_{\leq}(x, y) = 0$ sinon,

et clos par

- la composition et
- la minimisation.

Cette définition est similaire à la définition 3.1, mais la clôture par la récurrence y est remplacée par trois fonctions de base : l'addition, la multiplication et la fonction $\chi_{\leq}$. On veut montrer que ces deux définitions sont équivalentes, c'est-à-dire qu'une fonction appartient à l'ensemble $\mathcal{C}$ si et seulement si elle est calculable. Pour cela, on doit montrer que l'ensemble $\mathcal{C}$ est clos par définition par récurrence, c'est-à-dire que si g et h sont deux fonctions de $\mathcal{C}$ et si f est définie par récurrence à partir de g et h, alors f appartient à $\mathcal{C}$.

La fonction définie par récurrence à partir de g et h prend la valeur r en $x_1, \ldots, x_{n-1}, y$ s'il existe une suite finie $s_0, \ldots, s_y$ telle que $s_0 = g(x_1, \ldots, x_{n-1})$, pour tout i strictement inférieur à y, $s_{i+1} = h(x_1, \ldots, x_{n-1}, i, s_i)$ et $s_y = r$. On veut exprimer la suite finie $s_0, \ldots, s_y$ par un entier. La première idée est de l'exprimer par l'entier $s_0; (\ldots; (s_y; 0) \ldots)$ et il n'est pas difficile de montrer que les fonctions $;$, *hd* et *tl* appartiennent à l'ensemble $\mathcal{C}$. Toutefois, comme on ne dispose pas encore de la définition par récurrence, on ne peut pas encore montrer que la fonction *nth* appartient à l'ensemble $\mathcal{C}$. On utilise donc une autre expression des suites finies, qui repose sur une fonction appelée *la fonction β de Gödel*.

Définition 3.10

La fonction β de Gödel est définie par

$$\beta(k, l, i) = l \bmod (k(i + 1) + 1)$$

où $x \bmod y$ est le reste de la division euclidienne de x par y.

Proposition 3.15

Pour toute suite finie $s_0, \ldots, s_y$, il existe deux entiers k, l tels que pour tout i inférieur à y, $s_i = \beta(k, l, i)$.

Démonstration. Soit m un entier strictement supérieur à $y + 1, s_0, \ldots, s_y$. Il est facile de vérifier que les entiers $e_i = m!(i + 1) + 1$ pour i inférieur à y sont premiers entre eux deux à deux : si e_i et e_j avaient un diviseur commun premier c, alors c diviserait aussi leur différence $m!(i - j)$, il serait donc inférieur à m,

il diviserait $m!(i+1)$ et il ne pourrait donc pas diviser $e_i = m!(i+1)+1$. On utilise ensuite un résultat de théorie des nombres appelé *théorème des restes chinois* : si $e_0, \ldots, e_y$ est une suite d'entiers premiers entre eux deux à deux et $s_0, \ldots, s_y$ une suite d'entiers quelconques, il existe un entier z tel que pour tout i, $s_i \equiv z[\text{mod } e_i]$. D'après ce théorème, il existe un entier z tel que $s_i \equiv z[\text{mod } e_i]$. On pose $k = m!$ et $l = z$. On a $s_i \equiv l[\text{mod } e_i]$ et comme, par ailleurs s_i est inférieur à m qui est lui-même inférieur à k et donc à $k(i+1)+1$, s_i est le reste de la division euclidienne de l par $k(i+1)+1$, c'est-à-dire $s_i = \beta(k,l,i)$.

Le fait que la fonction définie par récurrence à partir de g et h prenne la valeur r en $x_1, \ldots, x_{n-1}, y$ s'exprime désormais par le fait qu'il existe deux nombres k et l tels que $\beta(k,l,0) = g(x_1, \ldots, x_{n-1})$, pour tout i strictement inférieur à y, $\beta(k,l,i+1) = h(x_1, \ldots, x_{n-1}, i, \beta(k,l,i))$ et $\beta(k,l,y) = r$.

Le fait que pour tout i strictement inférieur à y, on ait $\beta(k,l,i+1) = h(x_1, \ldots, x_{n-1}, i, \beta(k,l,i))$ peut s'exprimer comme le fait que la plus petite valeur d'annulation de la fonction f_1, qui à $x_1, \ldots, x_{n-1}, y, k, l, i$ associe 1 si $i < y$ et $\beta(k,l,i+1) = h(x_1, \ldots, x_{n-1}, i, \beta(k,l,i))$ et 0 sinon, est y. Toutefois, il n'est pas si facile de montrer l'appartenance à $\mathcal{C}$ de la fonction f_1, qui doit être définie et valoir 0 en y que h soit définie en $x_1, \ldots, x_{n-1}, y, \beta(k,l,y)$ ou non. Pour contourner cette difficulté, on montre un théorème un peu plus général : si la fonction f est calculable, alors la fonction f^* appartient à $\mathcal{C}$, où la fonction f^* est définie de la manière suivante.

Définition 3.11

Soit f une fonction de n arguments, on note f^* la fonction de $n+1$ arguments définie par $f^*(0, x_1, \ldots, x_n) = 0$ et $f^*(w, x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ si $w \neq 0$.

Proposition 3.16

L'ensemble $\mathcal{C}$ contient la soustraction $\dot{-}$, le quotient et le reste de la division euclidienne, les fonctions $\chi_{\mathbb{N}^*}$, $\chi_<$ et $\chi_=$ caractéristiques de l'ensemble $\mathbb{N}^*$ et des relations $<$ et $=$, la fonction β et les fonctions ;, *hd* et *tl*.

Démonstration. La soustraction, le quotient et le reste de la division euclidienne se définissent avec l'addition, la multiplication, la fonction caractéristique de la relation d'ordre et la minimisation. Les fonctions $\chi_{\mathbb{N}^*}$, $\chi_<$ et $\chi_=$ se définissent avec la soustraction. La fonction β avec l'addition la multiplication et le reste de la division euclidienne. La fonction ; avec l'addition, la multiplication et le quotient de la division euclidienne. Les fonctions *hd* et *tl* avec l'addition, la multiplication, le quotient de la division euclidienne et la minimisation.

Proposition 3.17

Si f est une fonction définie par récurrence à partir de deux fonctions g et h telles que g^* et h^* appartiennent à $\mathcal{C}$, alors f^* appartient à $\mathcal{C}$.

Démonstration. La fonction f^* vérifie les propriétés

$$f^*(w, x_1, \ldots, x_{n-1}, 0) = g^*(w, x_1, \ldots, x_{n-1})$$

$$f^*(w, x_1, \ldots, x_{n-1}, y+1) = h^*(w, x_1, \ldots, x_{n-1}, y, f^*(w, x_1, \ldots, x_{n-1}, y))$$

c'est donc la fonction définie par récurrence à partir de g^* et h^*.

Soit f_1 la fonction de $\mathcal{C}$ qui à $w, x_1, \ldots, x_{n-1}, y, k, l$ et i associe le nombre $\chi_<(i,y) \times \chi_=(h^*(w \times \chi_<(i,y), x_1, \ldots, x_{n-1}, i, \beta(k,l,i)), \beta(k,l,i+1))$. Si $i \geq y$ alors f^* est définie et nulle en $w, x_1, \ldots, x_{n-1}, y, k, l, i$ et si $i < y$, alors

$$f_1(w, x_1, \ldots, x_{n-1}, y, k, l) = \chi_=(h^*(w, x_1, \ldots, x_{n-1}, i, \beta(k,l,i)), \beta(k,l,i+1))$$

Soit f_2 la fonction de $\mathcal{C}$ qui à $w, x_1, \ldots, x_{n-1}, y$, k et l associe le plus petit entier i tel que $f_1(w, x_1, \ldots, x_{n-1}, y, k, l, i) = 0$. L'entier $f_2(w, x_1, \ldots, x_{n-1}, y, k, l)$ est égal à y si et seulement si pour tout i strictement inférieur à y, h^* est définie en $w, x_1, \ldots, x_{n-1}, i, \beta(k,l,i)$ et $h^*(w, x_1, \ldots, x_{n-1}, i, \beta(k,l,i)) = \beta(k,l,i+1)$.

On en déduit qu'il existe dans $\mathcal{C}$ une fonction f_3 qui à $w, x_1, \ldots, x_{n-1}, y$, k, l et r associe l'entier 0 si et seulement si la fonction g^* est définie en $w, x_1, \ldots, x_{n-1}$ et prend la valeur $\beta(k,l,0)$, pour tout i strictement inférieur à y, la fonction h^* est définie en $w, x_1, \ldots, x_{n-1}, i, \beta(k,l,i)$ et prend la valeur $\beta(k,l,i+1)$ et $\beta(k,l,y) = r$.

Soit f_4 la fonction de $\mathcal{C}$ qui à $w, x_1, \ldots, x_{n-1}, y$ associe le plus petit j tel que $f_3(w, x_1, \ldots, x_{n-1}, y, hd(hd(j)), tl(hd(j)), tl(j)) = 0$ et f_5 la fonction de $\mathcal{C}$ qui à $w, x_1, \ldots, x_{n-1}, y$ associe $tl(f_4(w, x_1, \ldots, x_{n-1}, y))$. D'après la proposition 3.15, la fonction f_5 prend la valeur r en $x_1, \ldots, x_{n-1}, y$ si et seulement s'il existe une suite s telle que $s_0 = g^*(w, x_1, \ldots, x_{n-1})$, pour tout i strictement inférieur à y, $s_{i+1} = h^*(w, x_1, \ldots, x_{n-1}, i, s_i)$ et $s_y = r$. La fonction f_5 est donc la fonction f^* qui, de ce fait, appartient à l'ensemble $\mathcal{C}$.

Proposition 3.18

Une fonction appartient à l'ensemble $\mathcal{C}$ si et seulement si elle est calculable.

Démonstration. D'après la définition 3.1 et les propositions 3.2 et 3.3, l'ensemble des fonctions calculables est clos par toutes les règles de la définition inductive de l'ensemble $\mathcal{C}$, il contient donc l'ensemble $\mathcal{C}$.

Réciproquement, on commence par montrer, par récurrence sur la construction de f que si f est une fonction calculable, alors f^* appartient à $\mathcal{C}$. Si f est

une projection, une fonction constante égale à 0 ou la fonction successeur, alors $f^*(w, x_1, \ldots, x_n) = \chi_{\mathbb{N}^*}(w) \times f(x_1, \ldots, x_n)$ et la fonction f^* appartient à l'ensemble $\mathcal{C}$. Si f est définie comme la composition de fonctions h et $g_1, \ldots, g_m$, alors, par hypothèse de récurrence, les fonctions h^*, $g_1^*, \ldots, g_m^*$ appartiennent à $\mathcal{C}$ et $f^*(w, x_1, \ldots, x_n) = h^*(w, g_1^*(w, x_1, \ldots, x_n), \ldots, g_m^*(w, x_1, \ldots, x_n))$. La fonction f^* est donc la composition d'une projection, de g_1^*, ..., g_m^* et de h^*. Elle appartient donc à l'ensemble $\mathcal{C}$. Si f est définie par récurrence à partir de deux fonctions g et h, alors, par hypothèse de récurrence, les fonctions g^* et h^* appartiennent à l'ensemble $\mathcal{C}$ et d'après la proposition 3.17, la fonction f^* également. Si f est définie par minimisation à partir d'une fonction g, alors, par hypothèse de récurrence, la fonction g^* appartient à $\mathcal{C}$ et la fonction f^* est la fonction qui à $w, x_1, \ldots, x_n$ associe le plus petit i tel que $g^*(w, x_1, \ldots, x_n, i) = 0$, c'est donc la fonction définie par minimisation à partir de g^* et elle appartient à $\mathcal{C}$.

Puisque la fonction f^* appartient à $\mathcal{C}$, c'est également le cas de la fonction qui à $x_1, \ldots, x_n$ associe l'entier $f^*(1, x_1, \ldots, x_n)$ qui n'est autre que la fonction f.

3.4 Les programmes

L'ensemble des fonctions calculables est défini de manière inductive. Pour chaque fonction calculable, il existe donc une ou plusieurs dérivations de cette fonction. Ces dérivations sont des arbres, que l'on appelle des *programmes*.

Les nœuds de ces arbres sont étiquetés par des symboles qui correspondent aux huit règles de la définition 3.9 : π_i^n, avec i et n entiers tels que i soit compris entre 1 et n, pour la règle des projections, Z^n pour la règle des fonctions identiquement nulle, *Succ* pour la règle du successeur, $+$ pour la règle de l'addition, $\times$ pour la règle de la multiplication, $\chi_{\leq}$ pour la règle de la fonction caractéristique de la relation d'ordre, $\circ_m^n$ pour la règle de composition et μ^n pour la règle de la minimisation. Les programmes sont donc des arbres étiquetés par les symboles π_i^n, Z^n, *Succ*, $+$, $\times$, $\chi_{\leq}$, $\circ_m^n$ et μ^n qui sont eux-mêmes des arbres étiquetés par les éléments d'un ensemble fini.

Naturellement, si, au lieu de s'appuyer sur la définition 3.9, on s'appuie sur la définition 3.1, les programmes sont des arbres étiquetés par les symboles π_i^n, Z^n, *Succ*, $\circ_m^n$, μ^n et Rec^n pour la règle de récurrence.

Définition 3.12 (Terminaison, valeur)

On dit qu'un programme *termine* en $p_1, \ldots, p_n$ si la fonction f calculée par ce

programme est définie en $p_1, \ldots, p_n$. On dit qu'il *prend la valeur* q en $p_1, \ldots, p_n$ si, en outre, $f(p_1, \ldots, p_n) = q$.

Exercice 3.3

Quelle est la fonction calculée par le programme $\circ_1^1(Succ, Succ)$?

Exercice 3.4

Quelle est la fonction calculée par le programme $\mu^1(\pi_1^2)$?

3.4.1 L'indécidabilité du problème de l'arrêt

Théorème 3.1 (L'indécidabilité du problème de l'arrêt)

L'ensemble des couples d'entiers (p, q) tels que p est le numéro d'un programme qui termine en q est indécidable.

Démonstration. Par l'absurde. Supposons l'existence d'un programme t tel que t appliqué à deux entiers p et q termine toujours, donne le résultat 1 si p est le numéro d'un programme qui termine quand on l'applique à l'entier q et le résultat 0 sinon. Rappelons que le programme $b = \mu^1(\pi_1^2)$ termine quand on l'applique à l'entier 0 et ne termine pas quand on l'applique à un entier non nul et considérons le programme $u = \circ_1^1(b, \circ_2^1(t, \pi_1^1, \pi_1^1))$. Quand on applique le programme u à un entier p, on applique d'abord le programme $\circ_2^1(t, \pi_1^1, \pi_1^1)$ à cet entier. On obtient le résultat 1 si p est le numéro d'un programme qui termine quand on l'applique à l'entier p et le résultat 0 sinon. Donc le programme u appliqué à l'entier p ne termine pas si p est le numéro d'un programme qui termine quand on l'applique à l'entier p et il termine sinon.

Soit m l'entier $\ulcorner u \urcorner$. Le programme u appliqué à l'entier m ne termine pas si m est le numéro d'un programme qui termine quand on l'applique à l'entier m et il termine sinon. Autrement dit, le programme u appliqué à m ne termine pas si le programme u termine quand on l'applique à m et il termine sinon. Le programme u appliqué à m ne termine pas si et seulement s'il termine, ce qui est contradictoire.

3.4.2 L'interpréteur

Il existe, en revanche, une fonction calculable G^n qui est un *interpréteur*, c'est-à-dire qui prend en argument le numéro d'un programme u d'arité n et des entiers $p_1, \ldots, p_n$ et retourne la valeur de la fonction calculable représentée par le programme u en $p_1, \ldots, p_n$, si cette valeur existe. Montrer l'existence de cet interpréteur nous donnera une généralisation du théorème d'indécidabilité du problème de l'arrêt. Cela nous mènera aussi à décrire un calcul comme une suite de petits pas, c'est-à-dire comme un processus se déroulant dans le temps, thème que nous reprendrons au chapitre 4.

Pour définir cette fonction G^n, nous avons besoin d'exprimer dans un même langage les programmes et les nombres entiers auxquels ces programmes s'appliquent. Nous étendons donc le langage des programmes avec des symboles 0 et S, distincts des symboles Z^n et *Succ* du langage des programmes, pour représenter ces entiers. Nous notons $\underline{p}$, le terme $S(S(\ldots(S(0))\ldots))$ avec p occurrences du symbole S, qui exprime l'entier p. Nous introduisons, ensuite, une famille de symboles App^n qui permettent de former des termes de la forme $App^n(u, \underline{p_1}, \ldots, \underline{p_n})$ qui expriment l'application du programme u aux entiers $p_1, \ldots, p_n$.

Le problème qui se pose est alors celui de définir une fonction calculable F qui au numéro d'un terme de la forme $App^n(u, \underline{p_1}, \ldots, \underline{p_n})$, associe le numéro d'un terme $\underline{q}$ exprimant l'entier q, qui est la valeur de la fonction calculable exprimée par le programme u en $p_1, \ldots, p_n$, si cette valeur existe. Le terme $\underline{q}$ est appelé la *valeur* du terme $App^n(u, \underline{p_1}, \ldots, \underline{p_n})$. Une fois cette fonction F définie, la fonction G^n se définira alors simplement, en utilisant la proposition 3.12, comme la fonction qui au numéro du terme u et aux entiers $p_1, \ldots, p_n$ associe l'entier q tel que $\ulcorner \underline{q} \urcorner = F(\ulcorner App^n(u, \underline{p_1}, \ldots, \underline{p_n}) \urcorner)$, si un tel entier existe.

Quand le programme u a la forme $\circ_m^n(w, v_1, \ldots, v_m)$, nous voulons, dans un premier temps calculer les valeurs $q_1, \ldots, q_m$ des programmes v_1, ..., v_m en $p_1, \ldots, p_n$, puis, dans un second temps, la valeur du programme w en $q_1, \ldots, q_m$. Pour cela, nous formons d'abord le terme $App^m(w, App^n(v_1, \underline{p_1}, \ldots, \underline{p_n}), \ldots, App^n(v_m, \underline{p_1}, \ldots, \underline{p_n}))$, puis nous calculons la valeur des termes $App^n(v_i, \underline{p_1}, \ldots, \underline{p_n})$ de manière à obtenir des entiers $\underline{q_1}, \ldots, \underline{q_m}$ et, enfin, nous calculons la valeur du terme $App^m(w, \underline{q_1}, \ldots, \underline{q_m})$. Procéder ainsi demande donc de pouvoir appliquer le symbole App^m, non seulement à des termes exprimant des entiers, mais aussi à d'autres termes formés avec le symbole App^n en attente d'être calculés. Et pour calculer la valeur d'un terme de cette forme, il faut commencer par calculer la valeur de ses arguments.

Pour le calcul des programmes formés par minimisation, nous aurons également besoin d'introduire une famille de symboles M^n et un symbole *Ifz*. La valeur du terme $M^{n+1}(u, \underline{p_1}, \ldots, \underline{p_n}, \underline{q})$ est le plus petit entier r supérieur ou

égal à q tel que $f(p_1, \ldots, p_n, r) = 0$ où f est la fonction calculée par le programme u. La valeur du terme $\mathit{Ifz}(\underline{p}, v, w)$ est la valeur du terme v si $p = 0$ et celle du terme w sinon.

Nous commençons par définir une fonction F_1 qui associe, à chaque numéro d'un terme de la forme $\mathit{App}^n(u, \underline{p_1}, \ldots, \underline{p_n})$, $M^{n+1}(u, v_1, \ldots, v_n, w)$ ou $\mathit{Ifz}(\underline{p}, v, w)$, le numéro d'un terme un peu plus calculé.

Définition 3.13 (Une étape de calcul à la racine)

La fonction calculable F_1 est définie par cas

- si t est le numéro d'un terme de la forme $\mathit{App}^n(u, \underline{p_1}, \ldots, \underline{p_n})$, alors
  - si u est de la forme π_i^n, on pose $F_1(t) = \ulcorner \underline{p_i} \urcorner$,
  - sinon, si u est de la forme Z^n, on pose $F_1(t) = \ulcorner \underline{0} \urcorner$,
  - sinon, si $u = \mathit{Succ}$, on pose $F_1(t) = \ulcorner S(\underline{p_1}) \urcorner$, c'est-à-dire $\ulcorner S \urcorner ; (\ulcorner \underline{p_1} \urcorner ; 0)$, ou encore $\ulcorner S \urcorner ; (\mathit{hd}(\mathit{tl}(\mathit{tl}(t))) ; 0)$,
  - sinon, si $u = +$, alors on pose $F_1(t) = \ulcorner \underline{p_1 + p_2} \urcorner$,
  - sinon, si $u = \times$, alors on pose $F_1(t) = \ulcorner \underline{p_1 \times p_2} \urcorner$,
  - sinon, si $u = \chi_{\leq}$, alors on pose $F_1(t) = \ulcorner \underline{\chi_{\leq}(p_1, p_2)} \urcorner$,
  - sinon, si u est de la forme $\circ_m^n(w, v_1, \ldots, v_m)$, on pose

$$F_1(t) = \ulcorner \mathit{App}^m(w, \mathit{App}^n(v_1, \underline{p_1}, \ldots, \underline{p_n}), \ldots, \mathit{App}^n(v_m, \underline{p_1}, \ldots, \underline{p_n})) \urcorner$$

  - sinon, si u est de la forme $\mu^n(v)$, on pose

$$F_1(t) = \ulcorner M^{n+1}(v, \underline{p_1}, \ldots, \underline{p_n}, \underline{0}) \urcorner$$

- sinon, si t est le numéro d'un terme de la forme $M^{n+1}(u, v_1, \ldots, v_n, w)$, on pose

$$F_1(t) = \ulcorner \mathit{Ifz}(\mathit{App}^{n+1}(u, v_1, \ldots, v_n, w), w, M^{n+1}(u, v_1, \ldots, v_n, S(w))) \urcorner$$

- sinon, si t est le numéro d'un terme de la forme $\mathit{Ifz}(\underline{p}, v, w)$, alors on pose $F_1(t) = \ulcorner v \urcorner$, si $p = 0$ et $F_1(t) = \ulcorner w \urcorner$ sinon,
- sinon, on pose $F_1(t) = 0$.

La fonction F_1 permet d'effectuer une étape de calcul dans un terme de la forme $\mathit{App}^n(u, v_1, \ldots, v_n)$ dans le cas où les termes $v_1, \ldots, v_n$ sont des entiers $\underline{p_1}, \ldots, \underline{p_n}$. Si, maintenant, ces termes demandent eux-mêmes à être calculés, la fonction F_2 effectue une étape de calcul dans l'un de ces termes.

Définition 3.14 (Une étape de calcul)

La fonction calculable F_2 est définie par récurrence bien fondée

- si t est le numéro d'un terme de la forme $App^n(u, v_1, \ldots, v_n)$, alors si les termes $v_1, \ldots, v_n$ sont de la forme $\underline{p_1}, \ldots, \underline{p_n}$, on pose $F_2(t) = F_1(t)$ sinon, soit v_i le premier terme qui n'est pas de la forme $\underline{p}$ et v' le terme de numéro $F_2(\ulcorner v_i \urcorner)$, on pose

$$F_2(t) = \ulcorner App^n(u, v_1, \ldots, v_{i-1}, v', v_{i+1}, \ldots, v_n) \urcorner$$

- sinon, si t est le numéro d'un terme de la forme $M^{n+1}(u, v_1, \ldots, v_n, w)$, on pose $F_2(t) = F_1(t)$,
- sinon, si t est le numéro d'un terme de la forme $Ifz(u, v, w)$, alors si le terme u est de la forme $\underline{p}$, on pose $F_2(t) = F_1(t)$, sinon soit u' le terme de numéro $F_2(\ulcorner u \urcorner)$, on pose $F_2(t) = \ulcorner Ifz(u', v, w)) \urcorner$,
- sinon, on pose $F_2(t) = 0$.

Pour définir l'interpréteur F, il suffit maintenant d'itérer la fonction F_2 sur un terme t jusqu'à obtenir un terme de la forme $\underline{q}$. Pour cela nous définissons successivement une fonction F_3 qui itère la fonction F_2 p fois, une fonction F_4 qui indique si $F_3(t, p)$ est de la forme $\underline{q}$ ou non, une fonction F_5 qui indique le nombre d'étapes nécessaires pour obtenir un tel terme et, enfin, la fonction F.

Définition 3.15 (L'interpréteur)

Soit F_3 la fonction calculable telle que $F_3(t, p) = F_2^p(t)$. Cette fonction est définie par récurrence par

$$F_3(t, 0) = t$$

$$F_3(t, p + 1) = F_2(F_3(t, p))$$

Soit F_4 la fonction calculable telle que $F_4(t, n) = 0$ si $F_3(t, n)$ est le numéro d'un terme de la forme $\underline{q}$ et $F_4(t, n) = 1$ sinon. Cette fonction est définie par composition entre F_3 et la fonction qui vaut 0 sur les numéros des termes de la forme $\underline{q}$ et 1 ailleurs. Soit F_5 la fonction calculable telle que $F_5(t)$ soit le nombre d'étapes nécessaires pour calculer t, $F_5(t)$ est défini par minimisation comme le plus petit entier p tel que $F_4(t, p) = 0$. Soit F la fonction calculable définie par $F(t) = F_3(t, F_5(t))$. Soit, enfin, G^n la fonction calculable qui à t et $p_1, \ldots, p_n$ associe l'entier q tel que $\ulcorner \underline{q} \urcorner = F(\ulcorner App^n(t, \underline{p_1}, \ldots, \underline{p_n}) \urcorner)$.

Si $F_3(t, n)$ n'est jamais le numéro d'un terme exprimant un entier, c'est-à-dire si les itérations de F_2 se poursuivent à l'infini, alors F_5 et F ne sont pas définies en t : l'interpréteur ne termine pas sur un programme qui ne termine pas.

Proposition 3.19

Soit f une fonction calculable, u le numéro d'un programme exprimant cette fonction et $p_1, \ldots, p_n$ des entiers. Alors, la fonction f est définie en $p_1, \ldots, p_n$ si et seulement si la fonction F est définie en $\ulcorner App^n(u, \underline{p_1}, \ldots, \underline{p_n})\urcorner$ et si ces deux fonctions sont définies, alors $F(\ulcorner App^n(u, \underline{p_1}, \ldots, \underline{p_n})\urcorner) = \ulcorner \underline{f(p_1, \ldots, p_n)}\urcorner$.

Démonstration. Par récurrence sur la construction de f.

Proposition 3.20

Soit f une fonction calculable, u le numéro d'un programme exprimant cette fonction et $p_1, \ldots, p_n$ des entiers. Alors, la fonction f est définie en $p_1, \ldots, p_n$ si et seulement si la fonction G^n est définie en $u, p_1, \ldots, p_n$ et si ces deux fonctions sont définies, alors $G^n(u, p_1, \ldots, p_n) = f(p_1, \ldots, p_n)$.

Démonstration. D'après la proposition 3.19.

Exercice 3.5

Montrer que la fonction F_3 est récursive primitive.

Exercice 3.6

Définir un interpréteur pour les programmes écrits dans le langage formé des symboles π_i^n, Z^n, $Succ$, $\circ_m^n$, μ^n et Rec^n, correspondant à la définition 3.1.

Un corollaire de l'existence de cet interpréteur est la généralisation suivante du théorème d'indécidabilité du problème de l'arrêt.

Proposition 3.21

Soit A un sous-ensemble décidable de l'ensemble des programmes qui terminent toujours. Alors, il existe une fonction calculable totale qui n'est représentée par aucun programme de A.

Démonstration. Soit H la fonction calculable suivante : si n est le numéro d'un programme unaire de A et p un entier, alors $H(n,p) = G^1(n,p)$, sinon $H(n,p) = 0$. La fonction H est calculable, c'est un interpréteur pour tous les programmes unaires de A et elle est totale. Soit la fonction H' telle que

$H'(p) = H(p,p)+1$. S'il y avait dans A un terme t représentant la fonction H', on aurait, pour tout p, $H(\ulcorner t \urcorner, p) = H'(p) = H(p,p)+1$. En particulier, pour $p = \ulcorner t \urcorner$, on aurait $H(\ulcorner t \urcorner, \ulcorner t \urcorner) = H(\ulcorner t \urcorner, \ulcorner t \urcorner)+1$ ce qui est contradictoire.

Un langage de programmation dont tous les programmes terminent est donc toujours incomplet, car il ne permet pas d'exprimer son propre interpréteur. C'est, par exemple, le cas du langage impératif formé de la déclaration de variable, de l'affectation, de la séquence, du test et de la boucle `for`.

On peut redémontrer ainsi l'indécidabilité du problème de l'arrêt, puisque si l'ensemble de tous les programmes qui terminent toujours était décidable, il existerait une fonction calculable totale qui ne serait représentée par aucun programme de cet ensemble, ce qui est contradictoire.

On peut aussi redémontrer l'existence d'une fonction calculable totale qui n'est pas récursive primitive, puisque l'ensemble des programmes exprimés dans le langage π_i^n, Z^n, $Succ$, $\circ_m^n$, Rec^n est décidable.

4
Le calcul comme une suite de petits pas

La définition de l'interpréteur, à la section 3.4.2, nous a donné une nouvelle manière d'aborder la notion de calcul, dans laquelle les programmes sont les expressions d'un langage $\mathcal{L}$. À partir d'un programme t et d'arguments p_1, ..., p_n, on construit un terme, qui est une expression d'un langage $\mathcal{L}'$, qui étend le langage $\mathcal{L}$. L'exécution des programmes est définie par une fonction calculable totale des termes de $\mathcal{L}'$ dans les termes de $\mathcal{L}'$. Cette fonction décrit un petit pas de calcul. On l'itère ensuite jusqu'à obtenir un entier, qui est le résultat du calcul, ou alors à l'infini, auquel cas le programme t ne termine pas sur les arguments $p_1, \ldots, p_n$. La notion de terminaison de la définition 3.12 rejoint ici l'acception courante : un programme qui ne termine pas est un programme qui calcule éternellement.

Les langages $\mathcal{L}$ et $\mathcal{L}'$ et la fonction qui décrit un petit pas de calcul définissent un langage de programmation et une fonction partielle f est dite *représentable* dans ce langage s'il existe un programme t tel que le terme formé du programme t et des entiers $p_1, \ldots, p_n$ se calcule en l'entier $f(p_1, \ldots, p_n)$ quand la fonction f est définie en p_1, ..., p_n et ne termine pas sinon. Il est facile de montrer que toutes les fonctions partielles représentables dans un tel langage sont calculables.

Dans certains langages de programmation, comme dans celui de la section 3.4.2, toutes les fonctions calculables sont représentables, on dit alors que ces langages sont *complets au sens de Turing*.

Entrent dans ce cadre les langages de programmation traditionnels comme Java, Caml, C, ... dont on peut décrire l'exécution des programmes comme une suite de petits pas.

Entrent également dans ce cadre un certain nombre de langages de programmation théoriques, plus dépouillés. Ces langages ont une utilité différente des langages de programmation traditionnels : les algorithmes sont bien entendu plus difficiles à exprimer dans ces langages organisés autour d'un nombre minimal de constructions, mais ils sont plus faciles à étudier : simplifier le langage dans lequel un algorithme est exprimé permet de simplifier les raisonnements qui permettent d'établir que cet algorithme termine, préserve un certain invariant ou a une certaine complexité.

Nous allons voir dans ce chapitre trois exemples de tels langages minimaux : la réécriture, le lambda-calcul et les machines de Turing.

4.1 La réécriture

La réécriture est le plus simple des langages dans lesquels le calcul est exprimé comme une suite de petits pas. Les termes avec lesquels on calcule sont simplement des termes d'un langage sans lieurs et le pas élémentaire de calcul est défini par un ensemble de règles, appelées *règles de réécriture*, qui spécifient comment transformer un terme en un autre. Par exemple, la règle

$$0 + y \longrightarrow y$$

spécifie que n'importe quel terme de la forme $0 + t$ peut se transformer en t.

Définition 4.1 (Règle de réécriture)

Soit $\mathcal{L}$ un langage sans lieurs. Une *règle de réécriture* du langage $\mathcal{L}$ est un couple formé de deux termes de $\mathcal{L}$, l et r, noté $l \longrightarrow r$.

Définition 4.2 (Une étape de réduction à la racine)

Soit $\mathcal{R}$ un ensemble de règles de réécriture, *une étape de $\mathcal{R}$-réduction à la racine* est la relation définie sur les termes du langage $\mathcal{L}$ par $t \longrightarrow u$ s'il existe une règle de réécriture $l \longrightarrow r$ dans $\mathcal{R}$ et une substitution σ telle que $\sigma l = t$ et $\sigma r = u$.

Définition 4.3 (Radical)

Soit $\mathcal{R}$ un ensemble de règles de réécriture, un terme t est un *radical* s'il est réductible par la relation $\longrightarrow$, c'est-à-dire s'il existe une règle de réécriture $l \longrightarrow r$ et une substitution σ telle que $\sigma l = t$.

La relation $\longrightarrow$ s'étend en une relation qui permet d'effectuer une réduction dans un sous-terme.

Définition 4.4 (Une étape de réduction)

Soit $\mathcal{R}$ un ensemble de règles de réécriture, *une étape de $\mathcal{R}$-réduction* est la relation inductivement définie par

- si $t \to u$ alors $t \rhd u$,
- si $t \rhd u$, alors $f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \rhd f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)$.

Définition 4.5 (La réduction)

La réduction $\rhd^*$ est la fermeture réflexive-transitive de la relation $\rhd$.

Exercice 4.1

Soit un ensemble formé des deux règles de réécriture

$$0 + y \longrightarrow y$$

$$S(x) + y \longrightarrow S(x + y)$$

Montrer que $S(S(0)) + S(S(0)) \rhd^* S(S(S(S(0))))$.

Définition 4.6 (Irréductibilité, terminaison)

Soit R une relation binaire. On dit qu'un élément t est *irréductible* pour la relation R s'il n'existe pas d'élément u tel que $t\ R\ u$.

On dit qu'un élément t *termine* s'il existe un élément t' irréductible tel que $t\ R^*\ t'$.

Définition 4.7 (Irréductibilité, terminaison d'un terme)

Soit $\mathcal{R}$ un ensemble de règles de réécriture. On dit qu'un terme t est *irréductible*, s'il est irréductible pour la relation $\rhd$, c'est-à-dire si aucun de ses sous-termes n'est un radical.

On dit qu'un terme t *termine* s'il termine pour la relation $\rhd$, c'est-à-dire s'il existe un terme irréductible t' tel que $t \rhd^* t'$.

Par exemple, si on a deux règles $f(x) \longrightarrow a$ et $\omega \longrightarrow \omega$, alors dans le terme ω ne termine pas car le seul terme en lequel il se réduise est ω lui-même. En revanche, le terme $f(\omega)$ termine. En effet, on peut réduire le sous-terme ω, ce

qui donne le terme $f(\omega)$ lui-même, mais aussi effectuer la réduction à la racine, ce qui donne le terme irréductible a.

Définition 4.8 (Confluence)

Une relation binaire R est dite *confluente* si à chaque fois que $t\ R^*\ u$ et $t\ R^*\ v$, il existe un w tel que $u\ R^*\ w$ et $v\ R^*\ w$.

Quand la relation $\triangleright$ est confluente, alors un terme u se réduit en au plus un terme irréductible : si $t \triangleright^* u$ et $t \triangleright^* v$ et u et v sont irréductibles, alors $u = v$. Plus généralement si t, u et v sont trois termes tels que $t \triangleright^* u$ et $t \triangleright^* v$ et v est irréductible, alors $u \triangleright^* v$. En revanche, comme nous l'avons vu avec le terme $f(\omega)$ ci-avant, il se peut qu'une manière de réduire le terme aboutisse à un terme irréductible et une autre non.

Définition 4.9 (Ensemble de règles orthogonal)

Un ensemble $\mathcal{R}$ de règles de réécriture est *orthogonal* si

- si $l \longrightarrow r$ est une règle de réécriture de $\mathcal{R}$, alors toutes les variables de r apparaissent dans l,
- si $l \longrightarrow r$ est une règle de réécriture de $\mathcal{R}$, alors chaque variable x a au plus une occurrence dans l,
- – si $l \longrightarrow r$ et $l' \longrightarrow r'$ sont deux règles de réécriture de $\mathcal{R}$ distinctes, et l'' est un sous-terme de l' distinct d'une variable, alors pour toute substitution σ et τ, $\sigma l \neq \tau l''$,
  - si $l \longrightarrow r$ est une règle de réécriture de $\mathcal{R}$, et l'' est un sous-terme de l distinct d'une variable et de l, alors pour toute substitution σ et τ, $\sigma l \neq \tau l''$.

Exercice 4.2

L'ensemble formé de la règle

$$c \longrightarrow x$$

est-t-il orthogonal ?

Montrer que le terme c se réduit en deux termes irréductibles distincts.

Exercice 4.3

L'ensemble formé des règles

$$g(h(x)) \longrightarrow a$$

$$f(g(x)) \longrightarrow b$$

est-t-il orthogonal ?

Montrer que le terme $f(g(h(c)))$ se réduit en deux termes irréductibles distincts.

Exercice 4.4

L'ensemble formé des règles

$$x - x \longrightarrow 0$$
$$S(x) - x \longrightarrow 1$$
$$\infty \longrightarrow S(\infty)$$

est-t-il orthogonal ?

Montrer que le terme $\infty - \infty$ se réduit en deux termes irréductibles distincts.

Cette notion d'orthogonalité est motivée par le résultat suivant que nous ne démontrons pas ici.

Proposition 4.1 (La confluence des ensembles de règles orthogonaux)

Si $\mathcal{R}$ est un ensemble orthogonal de règles de réécriture, alors la relation $\triangleright$ est confluente.

Définition 4.10 (La représentation des entiers)

Si le langage $\mathcal{L}$ contient une constante 0 et un symbole unaire S et si p est un entier, on note $\underline{p}$ le terme $S(S(\ldots(S(0))\ldots))$ avec p occurrences du symbole S.

Définition 4.11 (La représentation des fonctions)

Soit $\mathcal{L}$ un langage qui contient les symboles 0 et S et $\mathcal{R}$ un ensemble confluent de règles de réécriture tel que les termes de la forme $\underline{p}$ soient irréductibles, et F un symbole de $\mathcal{L}$. Soit f une fonction partielle. Le couple $\mathcal{R}, F$ *représente la fonction* f si pour tout $p_1, \ldots, p_n$

- si $f(p_1, \ldots, p_n) = q$ alors $F(\underline{p_1}, \ldots, \underline{p_n}) \triangleright^* \underline{q}$,
- si f n'est pas définie en $p_1, \ldots, p_n$, alors $F(\underline{p_1}, \ldots, \underline{p_n})$ ne termine pas.

Cette définition n'entre pas complètement dans le cadre que nous avons défini dans l'introduction de ce chapitre puisqu'un terme dont plusieurs sous-termes sont des radicaux peut se réduire en plusieurs autres termes.

On peut, toutefois, définir une stratégie particulière : la réduction *en appel par nom* qui supprime ce non-déterminisme quand l'ensemble de règles est orthogonal.

Définition 4.12 (Une étape de réduction en appel par nom)

Une étape de $\mathcal{R}$-réduction en appel par nom est la relation inductivement définie par

- si $t \longrightarrow t'$, alors $t \succ t'$,
- si $f(t_1, \ldots, t_n)$ n'est pas un radical, si $t_1, \ldots, t_{i-1}$ sont irréductibles, et si $t_i \succ t'_i$ alors $f(t_1, \ldots, t_{i-1}, t_i, t_{i+1}, \ldots, t_n) \succ f(t_1, \ldots, t_{i-1}, t'_i, t_{i+1}, \ldots, t_n)$.

Autrement dit, face à un terme qui contient plusieurs radicaux, on choisit un radical prioritaire : celui qui est le plus à gauche dans le terme.

On peut remarquer si un terme est irréductible pour la relation $\triangleright$, il ne contient pas de radicaux et il est donc également irréductible pour la relation $\succ$. Si, en revanche, un terme peut être réduit par la relation $\triangleright$, alors il contient un ou plusieurs radicaux et il peut être réduit par la relation $\succ$. Dans ce cas, cependant, si l'ensemble de règles est orthogonal, il existe un terme unique en lequel il se réduise en appel par nom.

Définition 4.13 (La réduction en appel par nom)

La réduction $\succ^*$ est la fermeture réflexive-transitive de la relation $\succ$, inductivement définie par

- $t \succ^* t$,
- si $t \succ t'$ et $t' \succ^* t''$, alors $t \succ^* t''$.

Cette stratégie nous donne une autre notion de représentation des fonctions.

Définition 4.14 (La représentation des fonctions en appel par nom)

Soit $\mathcal{L}$ un langage qui contient les symboles 0 et S et $\mathcal{R}$ un ensemble de règles de réécriture tel que les termes de la forme $\underline{p}$ soient irréductibles, et F un symbole de $\mathcal{L}$. Soit f une fonction partielle. Le couple $\mathcal{R}, F$ *représente la fonction f en appel par nom* si pour tout $p_1, \ldots, p_n$,

- si $f(p_1, \ldots, p_n) = q$, alors $F(\underline{p_1}, \ldots, \underline{p_n}) \succ^* \underline{q}$,
- si f n'est pas définie en $p_1, \ldots, p_n$, alors $F(\underline{p_1}, \ldots, \underline{p_n})$ ne termine pas en appel par nom.

La réduction en appel par nom nous permet de revenir dans le cadre que nous avons défini dans l'introduction de ce chapitre. Un programme est simplement un couple formé d'un ensemble orthogonal de règles de réécriture et d'un symbole de fonction, le terme formé du programme $(\mathcal{R}, F)$ et des entiers p_1, $\ldots$, p_n est le couple formé de l'ensemble de règles $\mathcal{R}$ et du terme $F(\underline{p_1}, \ldots, \underline{p_n})$ et le pas élémentaire de calcul est la réduction en appel par nom par les règles de $\mathcal{R}$.

Nous voulons maintenant associer, à chaque fonction calculable f, un ensemble de règles de réécriture qui représente cette fonction, à la fois en général et en appel par nom.

La principale difficulté est illustrée par cet exemple. Si g est une fonction qui n'est pas définie en 4 et h est la fonction identiquement nulle, la fonction $f = h \circ g$ n'est pas définie en 4. Pourtant, si on pose naïvement les règles $H(x) \longrightarrow 0$ et $F(x) \longrightarrow H(G(x))$, alors le terme $F(\underline{4})$ se réduit en $H(G(\underline{4}))$ puis en 0, alors qu'il ne devrait pas terminer. On modifie donc les règles $H(x) \longrightarrow 0$ et $F(x) \longrightarrow H(G(x))$ en $H(x) \longrightarrow 0\&x$ et $F(x) \longrightarrow H(G(x))\&x$ en introduisant un symbole binaire & tel que $t\&u$ se réduise en t si u se réduit en un entier, mais $t\&u$ ne termine pas si u ne termine pas. Cette propriété s'obtient en posant les règles $x\&0 \longrightarrow x$ et $x\&S(y) \longrightarrow x\&y$, qui effacent le terme u progressivement, à condition que ce soit la représentation d'un entier.

Définition 4.15 (La représentation des fonctions calculables)

Soit f une fonction calculable à n arguments, on associe à f un ensemble de règles de réécriture et un symbole F. Tous les ensembles contiennent les règles suivantes

$$x\&0 \longrightarrow x$$

$$x\&S(y) \longrightarrow x\&y$$

$$\mathit{Ifz}(0, y, z) \longrightarrow y$$

$$\mathit{Ifz}(S(x), y, z) \longrightarrow z\&x$$

Puis, on ajoute des règles spécifiques à la fonction f, par récurrence sur sa construction.

– Si la fonction f est la i-ième projection, on ajoute la règle

$$F(x_1, \ldots, x_n) \longrightarrow ((((x_i\&x_1)\& \ldots \&x_{i-1})\&x_{i+1})\& \ldots \&x_n)$$

– Si la fonction f est identiquement nulle, on ajoute la règle

$$F(x_1, \ldots, x_n) \longrightarrow ((0\&x_1)\& \ldots \&x_n)$$

– Si la fonction f est la fonction successeur, on ajoute la règle

$$F(x) \longrightarrow S(x)$$

– Si la fonction f est l'addition, on ajoute les règles

$$F(0, y) \longrightarrow y$$

$$F(S(x), y) \longrightarrow S(F(x, y))$$

– Si la fonction f est la multiplication, on ajoute les règles

$$F(0, y) \longrightarrow 0\&y$$

$$F(S(x), y) \longrightarrow F'(F(x, y), y)$$

$$F'(0, y) \longrightarrow y$$

$$F'(S(x), y) \longrightarrow S(F'(x, y))$$

– Si la fonction f est la fonction caractéristique de la relation d'ordre, on ajoute les règles

$$F(0, y) \longrightarrow S(0)\&y$$

$$F(S(x), 0) \longrightarrow 0\&x$$

$$F(S(x), S(y)) \longrightarrow F(x, y)$$

– Si la fonction f est la composée de h et $g_1, \ldots, g_m$, alors on considère les ensembles de règles de réécriture associés à ces fonctions en renommant les symboles afin que ces ensembles de règles ne partagent pas d'autres symboles que 0, S, & et *Ifz*, on prend l'union de ces ensembles de règles de réécriture et on ajoute la règle

$$F(x_1, \ldots, x_n) \longrightarrow (H(G_1(x_1, \ldots, x_n), \ldots, G_m(x_1, \ldots, x_n)))\&x_1\& \ldots \&x_n$$

– Si la fonction f est définie par minimisation à partir de la fonction g, alors on considère l'ensemble de règles de réécriture associé à cette fonction et on ajoute les règles

$$F(x_1, \ldots, x_n) \longrightarrow F'(x_1, \ldots, x_n, 0)$$

$$F'(x_1, \ldots, x_n, y) \longrightarrow \mathit{Ifz}(G(x_1, \ldots, x_n, y), y, F'(x_1, \ldots, x_n, S(y)))$$

Proposition 4.2

L'ensemble de règles construit à la définition 4.15 est confluent.

Démonstration. Cet ensemble est orthogonal. D'après la proposition 4.1, il est donc confluent.

Proposition 4.3

Si $f(p_1, \ldots, p_n) = q$ et les termes u_1, ..., u_n se réduisent en $\underline{p_1}$, ..., $\underline{p_n}$ en appel par nom, alors le terme $F(u_1, \ldots, u_n)$ se réduit en $\underline{q}$ en appel par nom.

Démonstration. Par récurrence sur la construction de f. Si f est une projection $F(u_1, \ldots, u_n)$ se réduit en $((((u_i \& u_1) \& \ldots \& u_{i-1}) \& u_{i+1}) \& \ldots \& u_n)$ qui se réduit, en appel par nom, en $\underline{p_i}$. Le cas où f est une fonction identiquement nulle, la fonction successeur, l'addition, la multiplication et la fonction caractéristique de la relation d'ordre sont similaires.

Si la fonction f est la composée de h et g_1, ..., g_m, alors $F(u_1, \ldots, u_n)$ se réduit, en appel par nom, en $(H(G_1(u_1, \ldots, u_n), \ldots, G_m(u_1, \ldots, u_n))) \& u_1 \& \ldots \& u_n$. Par hypothèse de récurrence, ce terme se réduit, en appel par nom, en $\underline{q} \& u_1 \& \ldots \& u_n$, puis en $\underline{q}$.

Si la fonction f est définie par minimisation à partir de la fonction g, alors $g(p_1, \ldots, p_n, r)$ est défini et prend une valeur non nulle pour tous les entiers r strictement inférieurs à q, et $g(p_1, \ldots, p_n, q) = 0$. Le terme $F(u_1, \ldots, u_n)$ se réduit, en appel par nom, en $F'(u_1, \ldots, u_n, 0)$, puis en $F'(u_1, \ldots, u_n, \underline{1}) \& v_0$, ..., $F'(u_1, \ldots, u_n, \underline{q}) \& v_{q-1} \& \ldots \& v_0$, où v_0 se réduit en $\underline{g(p_1, \ldots, p_n, 0)}$, ..., v_{q-1} en $\underline{g(p_1, \ldots, p_n, q-1)}$, puis en $\mathit{Ifz}(G(u_1, \ldots, u_n, \underline{q}), \underline{q}, F'(u_1, \ldots, u_n, \underline{q+1})) \& v_{q-1} \& \ldots \& v_0$, en $\mathit{Ifz}(0, \underline{q}, F'(u_1, \ldots, u_n, \underline{q+1})) \& v_{q-1} \& \ldots \& v_0$, en $\underline{q} \& v_{q-1} \& \ldots \& v_0$ et enfin en $\underline{q}$.

On veut montrer maintenant que si la fonction f n'est pas définie en $p_1, \ldots, p_n$, alors le terme $F(\underline{p_1}, \ldots, \underline{p_n})$ ne termine pas. On commence par la proposition suivante.

Proposition 4.4

Si l'un des termes $u_1, \ldots, u_n$ ne termine pas, alors $F(u_1, \ldots, u_n)$ ne termine pas, c'est-à-dire que si $F(u_1, \ldots, u_n) \rhd^* t'$, alors t' n'est pas irréductible.

Démonstration. On remarque tout d'abord que si un terme de la forme $S(u)$ ne termine pas, alors u non plus. Ensuite, si t est un terme, on définit l'ensemble des *sous-termes stricts* de t par récurrence sur la structure de t

- si $t = x$, alors $STS(t) = \{t\}$,
- si f est un symbole de fonction distinct de *Ifz* (c'est-à-dire l'un des symboles 0, S, & ou un symbole F associé à une fonction calculable) et $t = f(u_1, \ldots, u_n)$, alors $STS(t) = \{t\} \cup \bigcup_i STS(u_i)$,
- si $t = \mathit{Ifz}(u_1, u_2, u_3)$, alors $STS(t) = \{t\} \cup STS(u_1)$.

On montre que pour un ensemble de règles de réécriture construit à la définition 4.15, si $t \longrightarrow t'$ et $STS(t)$ contient un terme qui ne termine pas,

alors $STS(t')$ également. Soit u un élément de $STS(t)$ qui ne termine pas. Si le terme u est t lui-même, alors t' ne termine pas et $STS(t')$ contient donc un élément qui ne termine pas. Si cet élément est distinct de t, alors on vérifie, règle par règle, que ou bien cet élément appartient à $STS(t')$ qui contient donc un élément qui ne termine pas ou bien cet élément est de la forme $S(u')$ et $STS(t')$ contient u', qui, d'après la remarque ci-avant, ne termine pas.

On montre ensuite, par récurrence sur la structure de t, que si $t \rhd t'$ et $STS(t)$ contient un terme qui ne termine pas, alors $STS(t')$ également, puis que si $t \rhd^* t'$ et $STS(t)$ contient un terme qui ne termine pas, alors $STS(t')$ également.

On en déduit que si l'un des u_i ne termine pas et $F(u_1, \ldots, u_n) \rhd^* t'$, alors $STS(t')$ contient un terme qui ne termine pas. L'un des sous-termes de t' est donc un radical et t' n'est pas irréductible.

On peut alors montrer que si la fonction f n'est pas définie en $p_1, \ldots, p_n$, alors le terme $F(\underline{p_1}, \ldots, \underline{p_n})$ ne termine pas.

Proposition 4.5

Si les termes $u_1, \ldots, u_n$ se réduisent en les termes $\underline{p_1}$, ..., $\underline{p_n}$ et f n'est pas définie en $p_1, \ldots, p_n$, alors $F(u_1, \ldots, u_n)$ ne termine pas, c'est-à-dire que si $F(u_1, \ldots, u_n) \rhd^* t'$, alors t' n'est pas irréductible.

Démonstration. Soit t' un terme tel que $F(u_1, \ldots, u_n) \rhd^* t'$, on montre par récurrence sur la construction de f que t' n'est pas irréductible.

Les projections, les fonctions identiquement nulles, la fonction successeur, l'addition, la multiplication et la fonction caractéristique de la relation d'ordre sont totales.

Si la fonction f est définie comme la composée de h et $g_1, \ldots, g_m$, si, dans la suite de réduction de $F(u_1, \ldots, u_n)$ à t' on ne réduit jamais un terme à la racine, t' est lui-même un radical. Si on réduit un radical à la racine après un certain nombre d'étapes, on obtient le terme $H(G_1(u'_1, \ldots, u'_n), \ldots, G_m(u'_1, \ldots, u'_n))$ $\& u'_1 \& \ldots \& u'_n$ où les u'_i sont des réduits de u_i et t' est un réduit de ce terme. Par confluence, u'_i se réduit en $\underline{p_i}$. Si l'une des fonctions g_i n'est pas définie en $p_1, \ldots, p_n$, alors, par hypothèse de récurrence, l'un des termes $G_i(u'_1, \ldots, u'_n)$ ne termine pas et donc, d'après la proposition 4.4, le terme $H(G_1(u'_1, \ldots, u'_n), \ldots, G_m(u'_1, \ldots, u'_n))$ non plus et donc le terme $H(G_1(u'_1, \ldots, u'_n), \ldots, G_m(u'_1, \ldots, u'_n)) \& u'_1 \ldots \& u'_n$ non plus. Sinon, $g_i(p_1, \ldots, p_n) = q_i$ et la fonction h n'est pas définie en $q_1, \ldots, q_m$. Dans ce cas, $G_i(u'_1, \ldots, u'_n)$ se réduit en $\underline{q_i}$ et donc par hypothèse de récurrence $H(G_1(u'_1, \ldots, u'_n), \ldots, G_m(u'_1, \ldots, u'_n))$ ne termine pas. Le terme

$H(G_1(u'_1,\ldots,u'_n),\ldots,G_m(u'_1,\ldots,u'_n))\&u'_1\&\ldots\&u'_n$ ne termine donc pas non plus. Comme le terme $H(G_1(u'_1,\ldots,u'_n),\ldots,G_m(u'_1,\ldots,u'_n))\&u'_1\ldots\&u'_n$ ne termine pas, t' n'est pas un terme irréductible.

Si f est définie par minimisation d'une fonction g. Alors, ou bien la fonction g est partout définie et elle prend partout une valeur non nulle, ou bien elle prend une valeur non nulle jusqu'à un certaine valeur $q-1$, puis n'est pas définie en q.

Dans le premier cas, par hypothèse de récurrence si les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1}$, …, $\underline{p_n}$ et u' se réduit en un entier quelconque, le terme $G(u'_1,\ldots,u'_n,u')$ se réduit en un entier non nul. On construit inductivement un ensemble de termes qui contient

- les termes de la forme $F(u'_1,\ldots,u'_n)$ où les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1}$, …, $\underline{p_n}$,
- les termes de la forme $F'(u'_1,\ldots,u'_n,u')\&w_1\&\ldots\&w_s$ où les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1}$, …, $\underline{p_n}$ et $u',w_1,\ldots,w_s$ en des entiers quelconques,
- les termes de la forme $\mathit{Ifz}(t,u,v)\&w_1\&\ldots\&w_s$, où le terme t se réduit en un entier non nul, u est quelconque, v appartient à l'ensemble et $w_1,\ldots,w_s$ se réduisent en des entiers quelconques.

On montre que la réduction ne sort pas de cet ensemble et donc que le terme t' appartient à cet ensemble et qu'il n'est, de ce fait, pas irréductible.

Dans le second cas, par hypothèse de récurrence si les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1},\ldots,\underline{p_n}$ et u' se réduit en $\underline{r}$ pour $r<q$, le terme $G(u'_1,\ldots,u'_n,u')$ se réduit en un entier non nul. On construit inductivement un ensemble de termes qui contient

- les termes de la forme $F(u'_1,\ldots,u'_n)$ où les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1}$, …, $\underline{p_n}$,
- les termes de la forme $F'(u'_1,\ldots,u'_n,u')\&w_1\&\ldots\&w_s$ où les termes $u'_1,\ldots,u'_n$ se réduisent en $\underline{p_1}$, …, $\underline{p_n}$, u' en $\underline{r}$ pour $r<q$ et $w_1,\ldots,w_s$ en des entiers quelconques,
- les termes de la forme $\mathit{Ifz}(t,u,v)\&w_1\&\ldots\&w_s$ où le terme t se réduit en un entier non nul, u est quelconque, v appartient à l'ensemble et $w_1,\ldots,w_s$ se réduisent en des entiers quelconques,
- les termes de la forme $\mathit{Ifz}(t,u,v)\&w_1\&\ldots\&w_s$ où le terme t ne termine pas u et v sont quelconques et $w_1,\ldots,w_s$ se réduisent en des entiers quelconques,
- les termes de la forme $v\&t\&w_1\&\ldots\&w_s$ où le terme t ne termine pas v est quelconque et $w_1,\ldots,w_s$ se réduisent en des entiers quelconques.

On montre que la réduction ne sort pas de cet ensemble et donc que le terme t' appartient à cet ensemble et qu'il n'est, de ce fait, pas irréductible.

On peut enfin conclure.

Théorème 4.1

Toute fonction calculable est représentable par un ensemble de règles de réécriture en général et en appel par nom.

Ce théorème a une réciproque : toutes les fonctions représentables en appel par nom par un ensemble de règles de réécriture sont calculables. En effet, les termes du langage $\mathcal{L}$ sont des arbres, ils peuvent donc naturellement être numérotés. Il suffit ensuite de montrer que la fonction qui décrit un pas élémentaire de calcul, c'est-à-dire la fonction qui à t associe le terme u tel que $t \succ u$ est calculable.

Exercice 4.5

Donner une démonstration directe du fait que l'ensemble des fonctions exprimables par un ensemble de règles de réécriture est clos par définition par récurrence.

Exercice 4.6

Une relation R définie sur un ensemble E est dite *fortement confluente* si à chaque fois que $t\ R\ u$ et $t\ R\ v$, il existe un élément w tel que ($u\ R\ w$ ou $u = w$) et ($v\ R\ w$ ou $v = w$).

Montrer qu'une relation fortement confluente est confluente.

Exercice 4.7

Cet exercice demande d'avoir fait l'exercice 4.6.

Le but de cet exercice est de montrer un cas particulier du théorème selon lequel un ensemble de règles orthogonal définit une relation $\rhd$ confluente. Soit le langage formé des constantes a et b, du symbole de fonction unaire f et du symbole de fonction binaire g. Soit l'ensemble de règles

$$a \longrightarrow b$$

$$f(x) \longrightarrow g(x, x)$$

1. La relation $\rhd$ pour cet ensemble de règles est inductivement définie par les règles

$$\frac{}{a \rhd b}$$

$$\frac{}{f(t) \rhd g(t, t)}$$

$$\frac{t \rhd t'}{f(t) \rhd f(t')}$$

$$\frac{t_1 \rhd t'_1}{g(t_1, t_2) \rhd g(t'_1, t_2)}$$

$$\frac{t_2 \rhd t'_2}{g(t_1, t_2) \rhd g(t_1, t'_2)}$$

A-t-on $g(a, a) \rhd g(b, b)$? La relation $\rhd$ définie par cet ensemble de règles est-elle fortement confluente ?

2. Soit la variante de cette relation, *la réduction parallèle*, inductivement définie par les règles

$$\frac{}{t \rhd^{\|} t}$$

$$\frac{}{a \rhd^{\|} b}$$

$$\frac{}{f(t) \rhd^{\|} g(t, t)}$$

$$\frac{t \rhd^{\|} t'}{f(t) \rhd^{\|} f(t')}$$

$$\frac{t_1 \rhd^{\|} t'_1 \quad t_2 \rhd^{\|} t'_2}{g(t_1, t_2) \rhd^{\|} g(t'_1, t'_2)}$$

A-t-on $g(a, a) \rhd^{\|} g(b, b)$? Montrer que la relation $\rhd^{\|}$ est fortement confluente. Montrer que la relation $\rhd^{\|}$ est confluente.

3. Montrer que si $t \rhd u$ alors $t \rhd^{\|} u$. Montrer que si $t \rhd^{*} u$ alors $t \rhd^{\|*} u$. Montrer que si $t \rhd^{\|} u$ alors $t \rhd^{*} u$. Montrer que si $t \rhd^{\|*} u$ alors $t \rhd^{*} u$. Montrer que la relation $\rhd$ est confluente.

Exercice 4.8 (Le principe de récurrence nœthérienne)

Soit R une relation définie sur un ensemble E. Une *suite de réductions* pour cette relation est une suite finie ou infinie $x_0, x_1, x_2, \ldots$ telle que pour tout i, $x_i \ R \ x_{i+1}$. On dit qu'un élément x de E *termine fortement* si toute suite de réductions issue de x est finie.

On dit que la relation R *termine fortement* ou encore qu'elle est *bien fondée* ou encore qu'elle est *nœthérienne* si tout élément termine fortement.

1. Montrer qu'un élément qui termine fortement termine.
2. Donner une relation pour laquelle tout élément termine, mais il existe des éléments qui ne terminent pas fortement.

3. Soit R une relation sur un ensemble E. On dit qu'un élément u est un réduit de t, $t\ R^+\ u$, s'il existe une suite de réductions finie et non réduite à un élément qui va de t à u. Soit A un sous-ensemble de E tel que

$$\textit{pour tout élément } x \textit{ de } E$$
$$\textit{si tous les réduits de } x \textit{ sont dans } A\textit{, alors } x \textit{ est dans } A$$

Montrer que si x n'est pas dans A, il ne termine pas fortement. Montrer que si x termine fortement, il appartient à A. Montrer que si R est bien fondée alors tous les éléments de E appartiennent à A.

Exercice 4.9 (Le théorème de Newman)

Cet exercice demande d'avoir fait l'exercice 4.8.

Une relation R définie sur un ensemble E est dite *localement confluente* si à chaque fois que $t\ R\ u$ et $t\ R\ v$, il existe un élément w tel que $u\ R^*\ w$ et $v\ R^*\ w$.

1. On considère un ensemble formé de quatre éléments a, b, c et d et la relation définie par $a\ R\ b$, $b\ R\ a$, $a\ R\ c$ et $b\ R\ d$. Cette relation est-elle localement confluente ? Est-elle confluente ? Une relation localement confluente est-elle confluente ?
2. Montrer qu'une relation bien fondée et localement confluente est confluente.

4.2 Le lambda-calcul

L'idée du lambda-calcul est de rapprocher le langage des programmes de celui utilisé habituellement, en mathématiques, pour exprimer les fonctions.

Si e est une fonction, qui, à l'entier p, associe l'entier 2^p, la fonction qui à l'entier p associe l'entier 2^{2^p} s'exprime dans le langage de la section 3.4.2 par le terme $\circ_1^1(e, e)$ ou encore par le terme $\circ_1^1(e, \circ_1^1(e, \pi_1^1))$. On peut, cependant, l'écrire plus simplement $x \mapsto \mathit{App}(e, \mathit{App}(e, x))$, ou $\lambda x\ \mathit{App}(e, \mathit{App}(e, x))$, ou encore $\mathit{fun}\ x \to \mathit{App}(e, \mathit{App}(e, x))$, en utilisant un symbole binaire App, qui ne lie pas de variables dans ses arguments et un symbole unaire $\mapsto$, aussi noté λ ou fun, qui lie une variable dans son argument. En notant $(t\ u)$ le terme $\mathit{App}(t, u)$ l'expression ci-avant s'écrit plus simplement $\mathit{fun}\ x \to (e\ (e\ x))$.

Il n'est pas nécessaire d'étendre la notation fun pour construire des fonctions de plusieurs arguments, car on peut toujours construire de telles fonctions comme des fonctions d'un argument unique en utilisant l'isomorphisme $(A \times B) \to C = A \to (B \to C)$. Par exemple, la fonction qui à x et y associe le nombre $x \times x + y \times y$ est définie comme la fonction qui à x associe la fonction qui à y associe le nombre $x \times x + y \times y$: $\mathit{fun}\ x \to \mathit{fun}\ y \to (x \times x + y \times y)$.

Appliquer cette fonction f aux nombres 3 et 4 demande de l'appliquer d'abord à 3, ce qui donne le terme $(f\ 3)$, qui est la fonction qui à y associe $3\times 3+y\times y$, puis à 4, ce qui donne le terme $((f\ 3)\ 4)$.

Définition 4.16 (Le langage du lambda-calcul)

Le langage du lambda-calcul est formé d'un symbole binaire *App* qui ne lie pas de variables et d'un symbole unaire *fun* qui lie une variable dans son argument.

Quand on applique une fonction $\mathit{fun}\ x \to t$ à un terme u, on veut pouvoir transformer cette expression en l'expression $(u/x)t$ dans laquelle l'argument formel x a été substitué par l'argument réel u. Cette transformation est le pas élémentaire de calcul du lambda-calcul que l'on itère.

Définition 4.17 (Une étape de bêta-réduction à la racine)

Une étape de bêta-réduction à la racine est la relation sur les termes du lambda-calcul $\longrightarrow$ définie par

$$((\mathit{fun}\ x \to t)\ u) \longrightarrow (u/x)t$$

Définition 4.18 (Radical)

Un *radical* est un terme réductible par la relation $\longrightarrow$, c'est-à-dire un terme de la forme $((\mathit{fun}\ x \to t)\ u)$.

La relation $\longrightarrow$ s'étend en une relation qui permet de réduire un terme de la forme $((\mathit{fun}\ x \to t)\ u)$ dans un sous-terme.

Définition 4.19 (Une étape de bêta-réduction)

Une étape de bêta-réduction est la relation sur les termes du lambda-calcul $\rhd$ inductivement définie par

- si $t \longrightarrow t'$, alors $t \rhd t'$,
- si $t \rhd t'$, alors $(t\ u) \rhd (t'\ u)$,
- si $u \rhd u'$, alors $(t\ u) \rhd (t\ u')$,
- si $t \rhd t'$, alors $(\mathit{fun}\ x \to t) \rhd (\mathit{fun}\ x \to t')$.

Définition 4.20 (La bêta-réduction)

La bêta-réduction $\rhd^*$ est la fermeture réflexive-transitive de la relation $\rhd$.

Définition 4.21 (Irréductibilité, terminaison)

On dit qu'un terme t est *irréductible*, s'il est irréductible pour la relation $\rhd$, c'est-à-dire si aucun de ses sous-termes n'est un radical.

On dit qu'un terme t *termine* s'il termine pour la relation $\rhd$, c'est-à-dire s'il existe un terme irréductible t' tel que $t \rhd^* t'$.

Par exemple, le terme $((\mathit{fun}\, x \to (x\ x))\ y)$ termine car il se réduit en le terme irréductible $(y\ y)$. En revanche, le terme $\omega = ((\mathit{fun}\, x \to (x\ x))\ (\mathit{fun}\, x \to (x\ x)))$ ne termine pas, car le seul terme en lequel il se réduit est ω lui-même. Le terme $((\mathit{fun}\, x \to y)\ \omega)$, quant à lui, termine car il se réduit en le terme irréductible y.

Puisque, quand un terme contient plusieurs radicaux, on peut réduire n'importe lequel de ces radicaux, rien n'empêche, *a priori*, un terme de se réduire en plusieurs termes irréductibles distincts. On peut cependant montrer que ce n'est pas le cas, en utilisant un théorème de confluence de la relation $\rhd$, qui se démontre en montrant la confluence forte d'une relation de bêta-réduction parallèle, comme à l'exercice 4.7, mais que nous ne démontrons pas ici.

Proposition 4.6 (La confluence de la bêta-réduction)

La relation $\rhd$ est confluente.

La relation $\rhd$ étant confluente, un terme u se réduit en au plus un terme irréductible : si $t \rhd^* u$ et $t \rhd^* v$ et u et v sont irréductibles, alors $u = v$. Plus généralement, si t, u et v sont trois termes tels que $t \rhd^* u$ et $t \rhd^* v$ et v est irréductible, alors $u \rhd^* v$. En revanche, certains termes, comme le terme $((\mathit{fun}\, x \to y)\ \omega)$ ci-avant, se réduisent en un terme irréductible, si on choisit de réduire un radical, et se réduisent à l'infini, si on choisit d'en réduire un autre.

On peut remarquer que le lambda-calcul n'est pas tout à fait un ensemble de règles de réécriture au sens de la section précédente, car le symbole *fun* lie une variable, alors que les langages considérés à la section précédente était sans lieurs. De plus, le membre droit de la règle de bêta-réduction utilise une opération annexe : la substitution. Enfin, dans le membre gauche de la règle de bêta-réduction, on ne peut pas considérer t et u comme des variables que l'on instancierait avec une substitution σ, car la substitution évitant les captures, cela interdirait à x d'apparaître dans le terme t. Il serait donc nécessaire d'in-

troduire une distinction entre les variables comme x et les variables comme t afin que la substitution de la variable t par un terme permette de capturer x — le même genre de distinction est nécessaire si on veut étendre la logique des prédicats avec des lieurs dans les termes. Il existe des extensions de la notion de réécriture aux langages avec des symboles lieurs, qui vont dans ce sens, mais nous ne les aborderons pas dans ce livre.

Supposons que nous associons à chaque entier p un terme irréductible du lambda-calcul $\underline{p}$. Nous pouvons alors définir une notion de représentation des fonctions dans le lambda-calcul.

Définition 4.22 (La représentation des fonctions dans le lambda-calcul)

On dit qu'un terme F du lambda-calcul *représente* une fonction f des entiers dans les entiers si pour tout n-uplet d'entiers $p_1, \ldots, p_n$

- si $f(p_1, \ldots, p_n) = q$, alors $(F\ \underline{p_1}\ \ldots\ \underline{p_n}) \rhd^* \underline{q}$,
- si f n'est pas définie en $p_1, \ldots, p_n$, alors le terme $(F\ \underline{p_1}\ \ldots\ \underline{p_n})$ ne termine pas.

Comme dans le cas de la réécriture, cette définition n'entre pas complètement dans le cadre que nous avons défini dans l'introduction de ce chapitre puisqu'un terme dont plusieurs sous-termes sont des radicaux peut se réduire en plusieurs autres termes.

On peut, toutefois, définir une stratégie particulière : la réduction *en appel par nom* qui supprime ce non-déterminisme. De plus, un théorème, le théorème de standardisation, montre que l'on ne perd rien en se limitant à la réduction en appel par nom.

Définition 4.23 (Une étape de bêta-réduction en appel par nom)

Une étape de bêta-réduction en appel par nom est la relation sur les termes du lambda-calcul $\succ$ inductivement définie par

- si $t \longrightarrow t'$, alors $t \succ t'$,
- si $(t\ u)$ n'est pas un radical (c'est-à-dire si t n'est pas de la forme *fun*) et si $t \succ t'$, alors $(t\ u) \succ (t'\ u)$,
- si $(t\ u)$ n'est pas un radical et aucun sous-terme de t n'est un radical et $u \succ u'$, alors $(t\ u) \succ (t\ u')$,
- si $t \succ t'$, alors $(\mathit{fun}\ x \to t) \succ (\mathit{fun}\ x \to t')$.

Autrement dit, face à un terme qui contient plusieurs radicaux, on choisit un radical prioritaire. Si le terme a la forme $\mathit{fun}\ x \to t$, alors le radical prioritaire

est le radical prioritaire de t. Si le terme a la forme $(t\ u)$, alors on donne priorité au radical à la racine, s'il en existe un, sinon on donne priorité au radical prioritaire de t, s'il en existe un, et sinon au radical prioritaire de u. Le radical prioritaire est donc celui qui est le plus à gauche dans le terme.

On peut remarquer si un terme est irréductible pour la relation $\triangleright$, il ne contient pas de radicaux et il est donc également irréductible pour la relation $\succ$. Si, en revanche, un terme peut être réduit par la relation $\triangleright$, alors il contient un ou plusieurs radicaux et il peut être réduit par la relation $\succ$. Dans ce cas, cependant, il existe un terme unique en lequel il se réduit en appel par nom.

Définition 4.24 (La bêta-réduction en appel par nom)

La bêta-réduction $\succ^*$ est la fermeture réflexive-transitive de la relation $\succ$, inductivement définie par

- $t \succ^* t$,
- si $t \succ t'$ et $t' \succ^* t''$, alors $t \succ^* t''$.

Nous avons vu que certains termes, comme $((\mathit{fun}\ x \to y)\ \omega)$, se réduisent en un terme irréductible, si on choisit de réduire un radical, et se réduisent à l'infini, si on choisit d'en réduire un autre. Le théorème de standardisation, que nous ne démontrons pas ici, montre que pour un tel terme, la réduction en appel par nom termine toujours.

Proposition 4.7 (Le théorème de standardisation)

Si $t \triangleright^* t'$ et t' est irréductible, alors $t \succ^* t'$.

Une conséquence du théorème de standardisation est que si un terme ne termine pas pour la bêta-réduction en appel par nom, alors il ne termine pas pour la bêta-réduction en général.

Nous pouvons nous donc nous limiter à utiliser la réduction en appel par nom et définir de manière alternative les notions d'irréductibilité, de terminaison et de représentation des fonctions.

Proposition 4.8

- Un terme est irréductible si et seulement s'il ne peut pas être réduit par la relation $\succ$.
- Un terme t termine si et seulement s'il existe un terme irréductible t' tel que $t \succ^* t'$.

- Un terme F du lambda-calcul représente une fonction f des entiers dans les entiers si et seulement si pour tout n-uplet d'entiers $p_1, \ldots, p_n$
  - si $f(p_1, \ldots, p_n) = q$, alors $(F\ \underline{p_1}\ \ldots\ \underline{p_n}) \succ^* \underline{q}$,
  - si f n'est pas définie en $p_1, \ldots, p_n$, alors le terme $(F\ \underline{p_1}\ \ldots\ \underline{p_n})$ ne termine pas en appel par nom.

La réduction en appel par nom nous permet de revenir dans le cadre que nous avons défini dans l'introduction de ce chapitre. Un programme est un terme du lambda-calcul, le terme formé du programme F et des entiers $p_1, \ldots, p_n$ est simplement le terme $(F\ \underline{p_1}\ \ldots\ \underline{p_n})$ et le pas élémentaire de calcul est la réduction en appel par nom.

Nous voulons maintenant montrer que toutes les fonctions calculables sont représentables dans le lambda-calcul. Le choix du terme $\underline{p}$ représentant l'entier p a été originellement guidé par la volonté de représenter les fonctions définies par récurrence, qui a mené à donner une réponse originale à la question : qu'est-ce qu'un entier ? Au lieu de répondre que l'entier 3 est ce qu'il y a de commun à tous les ensembles de trois éléments, ce qui mène à la définition des entiers comme des cardinaux, on répond que l'entier 3 est un algorithme qui itère trois fois une fonction. Cela mène à la définition

$$\underline{3} = \mathit{fun}\ x \to \mathit{fun}\ f \to (f\ (f\ (f\ x)))$$

et plus généralement à la définition suivante.

Définition 4.25 (Les entiers de Church)

Le terme $\underline{p}$ est le terme

$$\underline{p} = \mathit{fun}\ x \to \mathit{fun}\ f \to (\underbrace{f\ (f\ \ldots\ (f}_{p \text{ fois}}\ x) \ldots))$$

Si le terme t est l'entier de Church $\underline{p}$ et que u et v sont des termes quelconques, alors le terme $(t\ u\ v)$ se réduit en appel par nom en deux étapes en le terme $w = (v\ (v\ \ldots\ (v\ u)\ldots))$ avec p occurrences du terme v.

Cependant, si le terme t se réduit, en appel par nom, en $\underline{p}$, sans être nécessairement égal à $\underline{p}$, on ne peut pas montrer que le terme $(t\ u\ v)$ se réduit en le terme $(v\ (v\ \ldots\ (v\ u)\ldots))$ car, quand on réduit $(t\ u\ v)$ en appel par nom, dès que le terme t a été réduit en un terme de la forme *fun*, le radical prioritaire n'est plus dans le réduit de t, mais à la racine. Toutefois, on peut montrer que si le terme t se réduit en $\underline{p}$ alors le terme $(t\ u\ v)$ appartient à l'ensemble $\mathcal{I}_p^{u,v}$ où la famille d'ensembles $(\mathcal{I}_p^{u,v})_p$ est définie par récurrence sur p de la manière suivante.

Définition 4.26

L'ensemble $\mathcal{I}_0^{u,v}$ est l'ensemble des termes qui se réduisent, en appel par nom, en u et l'ensemble $\mathcal{I}_{p+1}^{u,v}$ est l'ensemble des termes qui se réduisent en appel par nom, en un terme de la forme $(v\ w)$ où $w \in \mathcal{I}_p^{u,v}$.

Proposition 4.9

Si le terme t se réduit en $\underline{p}$, en appel par nom, alors le terme $(t\ u\ v)$ appartient à $\mathcal{I}_p^{u,v}$.

Démonstration. On montre, plus généralement, qu'il existe un terme w de $\mathcal{I}_p^{u,v}$, tel que le terme $(t\ u\ v)$ se réduise en w, en appel par nom, en deux étapes. Par récurrence double sur p et sur la longueur de la réduction de t à $\underline{p}$.

Si $t = \underline{p}$, alors le terme $(t\ u\ v)$ se réduit, en appel par nom, en deux étapes, en le terme $w = (v\ (v\ \ldots\ (v\ u)\ldots))$, avec p occurrences du terme v, qui appartient à $\mathcal{I}_p^{u,v}$.

Sinon, il existe un terme t' tel que $t \succ t'$ et t' se réduise en $\underline{p}$ en appel par nom en une étape de moins. Le cas où le terme t n'est pas de la forme *fun* est facile, car dans ce cas, le terme $(t\ u\ v)$ se réduit en $(t'\ u\ v)$ en appel par nom et il suffit d'appliquer l'hypothèse de récurrence.

Si, en revanche, t est de la forme *fun*, il s'écrit $\mathit{fun}\ y_1 \to \ldots\ \mathit{fun}\ y_n \to t_1$ où t_1 n'est pas de la forme *fun* et $n \neq 0$. Le terme t se réduisant en un entier de Church, on a $n = 1$ ou $n = 2$. Écrivons $t_1 = (r\ s_1\ \ldots\ s_m)$ où r n'est pas une application. Le terme r est donc une variable ou un terme de la forme *fun*.

Si le terme r est une variable, alors le terme t se réduisant en un entier de Church, mais n'étant pas irréductible, $n = 2$, $r = y_2$, $m = 1$. Le terme $(t\ u\ v)$ est donc égal à $((\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to (y_2\ s_1))\ u\ v)$ et il se réduit, en appel par nom, en deux étapes, en le terme $w = (v\ (u/y_1, v/y_2)s_1)$. On pose $w' = (u/y_1, v/y_2)s_1$. Le terme $\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to s_1$ se réduit en $\underline{p-1}$ en appel par nom et le terme $((\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to s_1)\ u\ v)$ se réduit en w', en appel par nom, en deux étapes. Par hypothèse de récurrence, le terme w' appartient à $\mathcal{I}_{p-1}^{u,v}$ et donc w appartient à $\mathcal{I}_p^{u,v}$.

Si le terme r est de la forme $\mathit{fun}\ z \to r'$, alors $t_1 = ((\mathit{fun}\ z \to r')\ s_1\ \ldots\ s_m)$ et ce terme n'étant pas de la forme *fun*, $m \neq 0$. Le terme t est donc de la forme $\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to \ldots\ \mathit{fun}\ y_n \to ((\mathit{fun}\ z \to r')\ s_1\ s_2\ \ldots\ s_m)$ et le terme t' en lequel il se réduit en appel par nom en une étape est $\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to \ldots\ \mathit{fun}\ y_n \to ((s_1/z)r'\ s_2\ \ldots\ s_m)$. Si $n = 1$, le terme $(t'\ u\ v)$ est égal à $((\mathit{fun}\ y_1 \to ((s_1/z)r'\ s_2\ \ldots\ s_m))\ u\ v)$ et il se réduit, en appel par nom, en une étape, en $w = (((u/y_1, (u/y_1)s_1/z)r'\ (u/y_1)s_2\ \ldots\ (u/y_1)s_m)\ v)$. Par hypothèse de récurrence, ce terme appartient à $\mathcal{I}_p^{u,v}$. Le terme $(t\ u\ v)$

est égal à $((\mathit{fun}\ y_1 \to ((\mathit{fun}\ z \to r')\ s_1\ \ldots\ s_m))\ u\ v)$, il se réduit, en appel par nom, en deux étapes, en w, dont on a montré qu'il appartenait à $\mathcal{I}_p^{u,v}$. Si $n = 2$, le terme $(t'\ u\ v)$ est égal à $((\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to ((s_1/z)r'\ s_2\ \ldots\ s_m))\ u\ v)$, il se réduit, en appel par nom, en deux étapes, en $b = ((u/y_1, v/y_2, (u/y_1, v/y_2)s_1/z)r'\ (u/y_1, v/y_2)s_2 \ldots (u/y_1, v/y_2)s_m)$. Par hypothèse de récurrence, ce terme appartient à $\mathcal{I}_p^{u,v}$. Le terme $(t\ u\ v)$ est égal à $((\mathit{fun}\ y_1 \to \mathit{fun}\ y_2 \to ((\mathit{fun}\ z \to r')\ s_1\ \ldots\ s_m))\ u\ v)$, il se réduit, en appel par nom, en trois étapes, en b. Le terme $(t\ u\ v)$ se réduit donc, en deux étapes, en un terme w qui se réduit en b. On a montré que le terme b appartenait à $\mathcal{I}_p^{u,v}$, c'est donc également le cas du terme w.

Proposition 4.10

Si t et u sont des termes qui se réduisent, en appel par nom, en des entiers de Church $\underline{n}$ et $\underline{p}$, et x, y et f sont des variables qui n'apparaissent pas dans t et u, alors

- le terme $\mathit{fun}\ x \to \mathit{fun}\ f \to (f\ (t\ x\ f))$ se réduit, en appel par nom, en le terme $\underline{n+1}$,
- le terme $\mathit{fun}\ x \to \mathit{fun}\ f \to (t\ (u\ x\ f)\ f)$ se réduit, en appel par nom, en le terme $\underline{n+p}$,
- le terme $\mathit{fun}\ x \to \mathit{fun}\ f \to (t\ x\ (\mathit{fun}\ y \to (u\ y\ f)))$ se réduit, en appel par nom, en le terme $\underline{n \times p}$,
- le terme $(t\ (K\ \underline{1})\ T\ (u\ (K\ \underline{0})\ T))$, où $K = \mathit{fun}\ x \to \mathit{fun}\ y \to x$ et $T = \mathit{fun}\ g \to \mathit{fun}\ h \to (h\ g)$, se réduit, en appel par nom, en le terme $\underline{\chi_{\leq}(n,p)}$.

Démonstration. On commence par montrer le lemme suivant par récurrence sur n : si un terme appartient à $\mathcal{I}_n^{v,f}$ où f est une variable et v un terme quelconque, alors ce terme se réduit, en appel par nom, en $(f\ (f\ \ldots\ (f\ v)\ldots))$ avec n occurrences du symbole f. On démontre ensuite les quatre propositions.

- Le terme $(t\ x\ f)$ se réduit, en appel par nom, en $(f\ (f\ \ldots\ (f\ x)\ldots))$ avec n occurrences du symbole f, le terme $(f\ (t\ x\ f))$ en $(f\ (f\ \ldots\ (f\ x)\ldots))$ avec $n+1$ occurrences du symbole f et le terme $\mathit{fun}\ x \to \mathit{fun}\ f \to (f\ (t\ x\ f))$ en $\underline{n+1}$.
- En utilisant la proposition 4.9, le terme $v = (u\ x\ f)$ appartient à $\mathcal{I}_p^{x,f}$ et le terme $(t\ (u\ x\ f)\ f)$ appartient à $\mathcal{I}_p^{v,f}$. En utilisant le lemme ci-avant, le terme $(t\ (u\ x\ f)\ f)$ se réduit, en appel par nom, en $(f\ (f\ \ldots\ (f\ v)\ldots))$ avec n occurrences du symbole f, puis en $(f\ (f\ \ldots\ (f\ x)\ldots))$ avec $n+p$ occurrences du symbole f. Le terme $\mathit{fun}\ x \to \mathit{fun}\ f \to (t\ (t\ x\ f)\ f)$ se réduit donc, en appel par nom, en $\underline{n+p}$.
- On montre, par récurrence sur n, que si un terme v appartient à

$\mathcal{I}_n^{x,\mathit{fun}\ y\to(u\ y\ f)}$, alors il se réduit, en appel par nom, en $(f\ (f\ \ldots\ (f\ x)\ldots))$, avec $n\times p$ occurrences du symbole f. Dans le cas $n=0$, le terme v se réduit, en appel par nom, en x. Sinon, il se réduit, en appel par nom, en $((\mathit{fun}\ y\to(u\ y\ f))\ v')$, puis en $(u\ v'\ f)$ avec v' dans $\mathcal{I}_{n-1}^{x,\mathit{fun}\ y\to(u\ y\ f)}$. D'après la proposition 4.9 ce terme appartient à $\mathcal{I}_p^{v',f}$ et, d'après le lemme ci-avant, il se réduit donc, en appel par nom, en $(f\ (f\ \ldots\ (f\ v')\ldots))$ avec p occurrences du symbole f, puis, par hypothèse de récurrence, en $(f\ (f\ \ldots\ (f\ x)\ldots))$ avec $p+(n-1)\times p=n\times p$ occurrences du symbole f. D'après la proposition 4.9, le terme $(t\ x\ (\mathit{fun}\ y\to(u\ y\ f)))$ appartient à $\mathcal{I}_n^{x,\mathit{fun}\ y\to(u\ y\ f)}$, il se réduit donc, en appel par nom, en $(f\ (f\ \ldots\ (f\ x)\ldots))$ avec $n\times p$ occurrences du symbole f. Le terme $\mathit{fun}\ x\to\mathit{fun}\ f\to(t\ x\ (\mathit{fun}\ y\to(u\ y\ f)))$ se réduit donc, en appel par nom, en $\underline{n\times p}$.

– On montre, par récurrence sur $n+p$, que si a est un terme de $\mathcal{I}_n^{(K\ \underline{\alpha}),T}$ et b est un terme de $\mathcal{I}_p^{(K\ \underline{\beta}),T}$, alors $(a\ b)$ se réduit, en appel par nom, en $\underline{\alpha}$, si $n\leq p$, et en $\underline{\beta}$, si $p+1\leq n$. Si $n=0$, alors le terme a se réduit, en appel par nom, en $(K\ \underline{\alpha})$ et comme ce terme n'est pas de la forme *fun*, le terme $(a\ b)$ se réduit, en appel par nom, en $(K\ \underline{\alpha}\ b)$, qui se réduit à son tour, en appel par nom, en $\underline{\alpha}$. Sinon, le terme a se réduit, en appel par nom, en $(T\ a')$, où a' est un élément de $\mathcal{I}_{n-1}^{(K\ \underline{\alpha}),T}$, et comme ce terme n'est pas de la forme *fun*, le terme $(a\ b)$ se réduit, en appel par nom, en $(T\ a'\ b)$, qui se réduit à son tour, en appel par nom, en $(b\ a')$ qui, par hypothèse de récurrence, se réduit, en appel par nom, en $\underline{\beta}$, si $p\leq n-1$, c'est-à-dire si $p+1\leq n$, et en $\underline{\alpha}$, si $n\leq p$. D'après la proposition 4.9, le terme $(t\ (K\ \underline{1})\ T)$ appartient à $\mathcal{I}_n^{(K\ \underline{1}),T}$ et le terme $(u\ (K\ \underline{0})\ T)$ à $\mathcal{I}_p^{(K\ \underline{0}),T}$. Le terme $(t\ (K\ \underline{1})\ T\ (u\ (K\ \underline{0})\ T))$ se réduit donc en $\underline{1}$, si $n\leq p$ et en $\underline{0}$ sinon, c'est-à-dire en $\underline{\chi_{\leq}(n,p)}$.

Définition 4.27 (Le test)

On pose

$$\mathit{Ifz}(t,u,v)=(t\ u\ \mathit{fun}\ x\to v)$$

où x est une variable qui n'apparaît pas dans v.

Proposition 4.11

Soient t, u et v trois termes du lambda-calcul tels que t se réduise, en appel par nom, en un entier de Church $\underline{p}$. Si $p=0$, alors $\mathit{Ifz}(t,u,v)\succ^* u$, et si $p\neq 0$, $\mathit{Ifz}(t,u,v)\succ^* v$.

Démonstration. D'après la proposition 4.9.

Comme dans le cas de la réécriture, si G est un terme qui représente une fonction g qui n'est pas définie en 4 et H un terme qui représente la fonction h identiquement nulle, alors il faut s'assurer que le terme qui représente la fonction $h \circ g$ ne termine pas en 4. Pour cela, comme dans le cas de la réécriture, la fonction h ne sera pas représentée par le terme $\mathit{fun}\ x \to \underline{0}$ mais par un terme un peu plus compliqué $\mathit{fun}\ x \to \underline{0}\&x$ qui s'assure que son argument termine sur un entier de Church et, de même, la fonction f sera représentée par le terme $\mathit{fun}\ x \to (H\ (G\ x))\&x$.

Définition 4.28

Pour tout terme t et pour tout terme u, on pose

$$t\&u = \mathit{Ifz}(u, t, t) = (u\ t\ (\mathit{fun}\ x \to t))$$

où x est une variable qui n'apparaît pas dans t.

Proposition 4.12

Soient t et u deux termes du lambda-calcul tels que u se réduise, en appel par nom, en un entier de Church. Alors $t\&u \succ^* t$.

Démonstration. D'après la proposition 4.11.

Enfin, pour représenter les fonctions définies par minimisation, il faut formuler, dans le lambda-calcul, un mécanisme qui permet d'itérer perpétuellement le test de la valeur de $g(0)$, $g(1)$, $g(2)$, ... jusqu'à obtenir la valeur 0. Pour cela on utilise le fait qu'il est possible dans le lambda-calcul d'appliquer une fonction à elle-même.

Définition 4.29 (Le point fixe)

Pour tout terme t, on pose

$$Y_t = ((\mathit{fun}\ x \to (t\ (x\ x)))\ (\mathit{fun}\ x \to (t\ (x\ x))))$$

Proposition 4.13

$Y_t \succ (t\ Y_t)$.

Nous pouvons à présent associer à chaque fonction calculable un terme du lambda-calcul.

Définition 4.30 (La représentation des fonctions calculables)

Soit f une fonction calculable à n arguments, on associe à f un terme du lambda-calcul, défini par récurrence sur la construction de f.

Si f est la i-ième projection, on lui associe le terme

$$\mathit{fun}\ x_1 \to\ \ldots\ \mathit{fun}\ x_n \to ((((x_i \& x_1) \& \ldots \& x_{i-1}) \& x_{i+1}) \& \ldots \& x_n)$$

Si f est la fonction identiquement nulle, on lui associe le terme

$$\mathit{fun}\ x_1 \to\ \ldots\ \mathit{fun}\ x_n \to ((\underline{0} \& x_1) \& \ldots \& x_n)$$

Si f est la fonction successeur, on lui associe le terme

$$S = \mathit{fun}\ n \to ((\mathit{fun}\ x \to \mathit{fun}\ f \to (f\ (n\ x\ f))) \& n)$$

Si f est l'addition, on lui associe le terme

$$\mathit{fun}\ p \to \mathit{fun}\ q \to ((\mathit{fun}\ x \to \mathit{fun}\ f \to (p\ (q\ x\ f)\ f)) \& p \& q)$$

Si f est la multiplication, on lui associe le terme

$$\mathit{fun}\ p \to \mathit{fun}\ q \to ((\mathit{fun}\ x \to \mathit{fun}\ f \to (p\ x\ (\mathit{fun}\ y \to (q\ y\ f)))) \& p \& q)$$

Si f est la fonction caractéristique de la relation d'ordre, on lui associe le terme

$$\mathit{fun}\ p \to \mathit{fun}\ q \to ((p\ (K\ \underline{1})\ T\ (q\ (K\ \underline{0})\ T)) \& p \& q)$$

où $K = \mathit{fun}\ x \to \mathit{fun}\ y \to x$ et $T = \mathit{fun}\ g \to \mathit{fun}\ h \to (h\ g)$. Si f est définie par composition des fonctions h et $g_1, \ldots, g_m$, alors soient $G_1, \ldots, G_m$ et H les termes associés à ces fonctions, on associe à f le terme

$$\mathit{fun}\, x_1 \to\ \ldots\ \mathit{fun}\, x_n \to ((H\ (G_1\ x_1\ \ldots\ x_n)\ \ldots\ (G_m\ x_1\ \ldots\ x_n)) \& x_1 \& \ldots \& x_n)$$

Si f est définie par minimisation d'une fonction g, alors soit G le terme associé à cette fonction, soit G' le terme $\mathit{fun}\ f \to \mathit{fun}\ x_1 \to\ \ldots\ \mathit{fun}\ x_n \to \mathit{fun}\ x_{n+1} \to (\mathit{Ifz}((G\ x_1\ \ldots\ x_n\ x_{n+1}), x_{n+1}, (f\ x_1\ \ldots\ x_n\ (S\ x_{n+1}))))$ on associe à f le terme

$$\mathit{fun}\ x_1 \to\ \ldots\ \mathit{fun}\ x_n \to ((Y_{G'}\ x_1\ \ldots\ x_n\ \underline{0}) \& x_1 \& \ldots \& x_n)$$

Nous voulons maintenant montrer que ces termes représentent les fonctions calculables auxquelles ils sont associés.

Proposition 4.14

Soit F un terme du lambda-calcul associé à une fonction calculable f et soient $p_1, \ldots, p_n$ des entiers tels que $f(p_1, \ldots, p_n) = q$, alors

$$(F\ \underline{p_1}\ \ldots\ \underline{p_n}) \succ^* \underline{q}$$

Démonstration. Par récurrence sur la construction de la fonction f, on montre plus généralement que si $u_1, \ldots, u_n$ sont des termes qui se réduisent, en appel par nom, en $\underline{p_1}, \ldots, \underline{p_n}$, alors $(F\ u_1\ \ldots\ u_n) \succ^* \underline{q}$.

Si f est une projection, une fonction identiquement nulle, la fonction successeur, l'addition, la multiplication ou la fonction caractéristique de la relation d'ordre, la propriété est une conséquence des propositions 4.10 et 4.12.

Si f est une fonction définie par composition des fonctions h et $g_1, \ldots, g_m$, alors il existe des entiers $r_1, \ldots, r_m$ tels que $g_1(p_1, \ldots, p_n) = r_1$, $\ldots$, $g_m(p_1, \ldots, p_n) = r_m$ et $h(r_1, \ldots, r_m) = q$. En utilisant la proposition 4.12, le terme $(F\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $(H\ (G_1\ u_1\ \ldots\ u_n)\ \ldots\ (G_m\ u_1\ \ldots\ u_n))$. Par hypothèse de récurrence, le terme $(G_1\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $\underline{r_1}$, $\ldots$, $(G_m\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $\underline{r_m}$ et $(H\ (G_1\ u_1\ \ldots\ u_n)\ \ldots\ (G_m\ u_1\ \ldots\ u_n))$ se réduit, en appel par nom, en $\underline{q}$.

Si f est une fonction définie par minimisation d'une fonction g, alors pour tout r strictement inférieur à q, $g(p_1, \ldots, p_n, r)$ est un entier non nul et $g(p_1, \ldots, p_n, q) = 0$. Si u est un terme qui se réduit, en appel par nom, vers un entier de Church $\underline{r}$ pour r strictement inférieur à q, alors le terme $(Y_{G'}\ u_1\ \ldots\ u_n\ u)$ se réduit, en appel par nom, en

$$\mathit{Ifz}((G\ u_1\ \ldots\ u_n\ u), u, (Y_{G'}\ u_1\ \ldots\ u_n\ (S\ u)))$$

Par hypothèse de récurrence, le terme $(G\ u_1\ \ldots\ u_n\ u)$ se réduit, en appel par nom, en un entier de Church non nul et donc, d'après la proposition 4.11, le terme $(Y_{G'}\ u_1\ \ldots\ u_n\ u)$ se réduit, en appel par nom, en $(Y_{G'}\ u_1\ \ldots\ u_n\ (S\ u))$. Ainsi, en utilisant la proposition 4.12, le terme $(F\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $(Y_{G'}\ u_1\ \ldots\ u_n\ \underline{0})$, puis en $(Y_{G'}\ u_1\ \ldots\ u_n\ (S\ \underline{0}))$, $(Y_{G'}\ u_1\ \ldots\ u_n\ (S\ (S\ \underline{0})))$, $\ldots (Y_{G'}\ u_1\ \ldots\ u_n\ (S^q\ \underline{0}))$. Enfin, ce terme se réduit, en appel par nom, en

$$\mathit{Ifz}((G\ u_1\ \ldots\ u_n\ (S^q\ \underline{0})), (S^q\ \underline{0}), (Y_{G'}\ u_1\ \ldots\ u_n\ (S\ (S^q\ \underline{0}))))$$

Par hypothèse de récurrence, le terme $(G\ u_1\ \ldots\ u_n\ (S^q\ \underline{0}))$ se réduit, en appel par nom, en $\underline{0}$ et donc, d'après la proposition 4.11, ce terme se réduit, en appel par nom, en $(S^q\ \underline{0})$ et finalement en $\underline{q}$.

Nous voulons maintenant montrer que si F est un terme du lambda-calcul qui est associé à une fonction calculable f et $u_1, \ldots, u_n$ sont des termes qui

se réduisent, en appel par nom, en des entiers de Church $\underline{p_1}, \ldots, \underline{p_n}$ tels que f ne soit pas définie en $p_1, \ldots, p_n$, alors le terme $(F\ u_1\ \ldots\ u_n)$ ne termine pas. Malheureusement, cette propriété de non-terminaison se compose mal. Par exemple, le terme $\mathit{fun}\ f \to (f\ \omega)$ ne termine pas, mais en appliquant ce terme à $\mathit{fun}\ x \to y$ on obtient un terme qui termine. Nous allons donc montrer une propriété un peu plus forte : que le terme $(F\ u_1\ \ldots\ u_n)$ est *isolé*.

Définition 4.31 (Terme isolé)

Un terme t est *isolé*, si pour tout terme t' tel que $t \succ^* t'$, le terme t' n'est ni irréductible ni de la forme *fun*.

Proposition 4.15

Si $t \succ^* u$ et u est isolé, alors t est isolé.

Démonstration. Soit t' un terme tel que $t \succ^* t'$. Comme t se réduit en appel par nom à la fois en u et en t', ou bien $u \succ^* t'$ ou bien $t' \succ^* u$. Dans le premier cas, t' n'est ni irréductible ni de la forme *fun*. Dans le second, t' n'est pas irréductible et s'il était de la forme *fun*, le terme u également serait de la forme *fun*, ce qui n'est pas le cas puisqu'il est isolé.

Proposition 4.16

Si t est isolé, alors les termes $(t\ u)$, $\mathit{Ifz}(t, u, v)$ et $u\&t$ sont isolés.

Démonstration. Si t est isolé, alors la suite de réductions $t = t_0, t_1, \ldots$ en appel par nom issue de t ne contient que des termes qui contiennent un radical et qui ne sont pas de la forme *fun*. La suite de réductions issue de $(t\ u)$ est donc $(t_0\ u), (t_1\ u), \ldots$ En effet, pour tout i, t_i contient un radical et n'est pas de la forme *fun*, le radical prioritaire de $(t_i\ u)$ est donc le radical prioritaire de t_i. On en déduit que $(t\ u)$ est isolé.

Les termes $\mathit{Ifz}(t, u, v)$ et $u\&t$ sont donc isolés.

Proposition 4.17

Soit F un terme du lambda-calcul associé à une fonction calculable f. Soient $u_1, \ldots, u_n$ des termes tels que chacun des u_i se réduise en un entier de Church ou soit isolé. Si au moins l'un des u_i est isolé, alors $(F\ u_1\ \ldots\ u_n)$ est isolé.

Démonstration. Par cas sur la construction de la fonction f, en utilisant les propositions 4.15 et 4.16 dans chacun des cas.

Proposition 4.18

Soit F un terme du lambda-calcul associé à une fonction calculable f et soient $p_1, \ldots, p_n$ des entiers tels que f ne soit pas définie en $p_1, \ldots, p_n$, alors le terme $(F\ \underline{p_1}\ \ldots\ \underline{p_n})$ ne termine pas.

Démonstration. Par récurrence sur la construction de la fonction f, on montre plus généralement que si $u_1, \ldots, u_n$ sont des termes qui se réduisent, en appel par nom, en des entiers de Church $\underline{p_1}, \ldots, \underline{p_n}$, alors le terme $(F\ u_1\ \ldots\ u_n)$ est isolé.

Les projections, les fonctions identiquement nulles, la fonction successeur, l'addition, la multiplication et la fonction caractéristique de la relation d'ordre sont totales.

Si la fonction f est définie par composition à partir des fonctions h et $g_1, \ldots, g_m$, alors, d'après la proposition 4.12, le terme $(F\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en

$$(H\ (G_1\ u_1\ \ldots\ u_n)\ \ldots\ (G_m\ u_1\ \ldots\ u_n))$$

Si l'une des fonctions g_i n'est pas définie en $p_1, \ldots, p_n$, alors, par hypothèse de récurrence, l'un des termes $(G_i\ u_1\ \ldots\ u_n)$ est isolé. D'après la proposition 4.17, le terme $(H\ (G_1\ u_1\ \ldots\ u_n)\ \ldots\ (G_m\ u_1\ \ldots\ u_n))$ est isolé. Et d'après la proposition 4.15, le terme $(F\ u_1\ \ldots\ u_n)$ également. Si, en revanche, il existe des entiers $r_1, \ldots, r_m$ tels que $r_1 = g_1(p_1, \ldots, p_n), \ldots, r_m = g_m(p_1, \ldots, p_n)$, alors h n'est pas définie en $r_1, \ldots, r_m$. Les termes $(G_i\ u_1\ \ldots\ u_n)$ se réduisent en $\underline{r_i}$ et, par hypothèse de récurrence, le terme $(H\ (G_1\ u_1\ \ldots\ u_n)\ \ldots\ (G_m\ u_1\ \ldots\ u_n))$ est isolé. D'après la proposition 4.15, le terme $(F\ u_1\ \ldots\ u_n)$ également.

Si la fonction f est définie par minimisation à partir d'une fonction g alors si g prend des valeurs non nulles en $(p_1, \ldots, p_n, 0)$, $(p_1, \ldots, p_n, 1)$, $\ldots$, alors, le terme $(F\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $(Y_{G'}\ u_1\ \ldots\ u_n\ \underline{0})$, $(Y_{G'}\ u_1\ \ldots\ u_n\ (S\ \underline{0}))$, $\ldots$ et il est donc isolé. Si, en revanche, la fonction g prend des valeurs non nulles en $(p_1, \ldots, p_n, 0)$, $(p_1, \ldots, p_n, 1)$, $\ldots$, $(p_1, \ldots, p_n, q-1)$ et n'est pas définie en $(p_1, \ldots, p_n, q)$, alors le terme $(F\ u_1\ \ldots\ u_n)$ se réduit, en appel par nom, en $\mathit{Ifz}((G\ u_1\ \ldots\ u_n\ (S^q\ \underline{0})), (S^q\ \underline{0}), (Y_{G'}\ u_1\ \ldots\ u_n\ (S^q\ \underline{0})))$ où le terme $(G\ u_1\ \ldots\ u_n\ (S^q\ \underline{0}))$ est isolé. D'après la proposition 4.16, le terme $\mathit{Ifz}((G\ u_1\ \ldots\ u_n\ (S^q\ \underline{0})), (S^q\ \underline{0}), (Y_{G'}\ u_1\ \ldots\ u_n\ (S^q\ \underline{0})))$ est isolé. Et donc, d'après la proposition 4.15, le terme $(F\ u_1\ \ldots\ u_n)$ également.

On peut enfin conclure.

Théorème 4.2

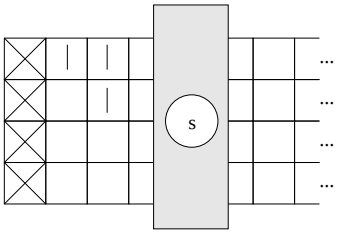
Toutes les fonctions calculables sont représentables dans le lambda-calcul.

Ce théorème a une réciproque : toutes les fonctions représentables dans le lambda-calcul sont calculables. En effet, les termes du lambda-calcul sont des arbres, ils peuvent donc naturellement être numérotés. Il suffit ensuite de montrer que la fonction qui décrit un pas élémentaire de calcul, c'est-à-dire la fonction qui à t associe le terme u tel que $t \succ u$ est calculable.

4.3 Les machines de Turing

Si le lambda-calcul tentait de rapprocher la notation des programmes de la notation habituelle des fonctions, les machines de Turing tentent, en revanche, de prendre en compte le fait qu'un calcul se déroule non seulement dans le temps, mais aussi dans l'espace.

Une *machine de Turing* est constituée d'un certain nombre k de bandes. Chaque bande contient une infinité de cases : une première, puis une deuxième située à droite de la première, une troisième située à droite de la deuxième, ...



Chaque case contient un symbole qui appartient à un ensemble fini Σ. Cet ensemble contient, parmi d'autres symboles, un symbole blanc b et une croix $\times$. Au départ, seul un nombre fini de cases contient un symbole différent de b et cette propriété est un invariant de l'évolution de la machine. Les croix servent à repérer la première case de chaque bande.

Un autre ingrédient qui entre dans la constitution d'une machine de Turing est une *tête de lecture et d'écriture* qui à chaque instant se trouve à une certaine position sur les bandes. Cette tête est dans un état s, qui varie au cours du temps dans un ensemble fini.

Un dernier ingrédient est une *table de transition* qui décrit l'évolution dans le temps de la machine. À chaque petit pas de calcul, la tête lit le contenu des k bandes à sa position courante. En fonction de ce k-uplet de symboles et de son

état, la table prescrit : un k-uplet de symboles à écrire sur les bandes, un nouvel état et un déplacement -1 : à gauche, 0 : sur place ou $+1$: à droite. La tête écrit les k symboles, change d'état et de position, puis passe au petit pas suivant. La table de transition est donc une fonction de $\Sigma^k \times S$ dans $\Sigma^k \times S \times \{-1, 0, +1\}$. On se limite au cas où la table de transition est telle que la machine n'écrit jamais ni n'efface de croix et ne peut pas se déplacer vers la gauche quand elle lit une suite de k croix.

Quand la machine démarre, la tête est toujours à gauche, c'est-à-dire sur la première case de chaque bande, et dans un état particulier, appelé *état initial*. Quand elle atteint un certain état, appelé *état final*, si jamais elle l'atteint, elle a terminé son calcul.

Une machine se définit donc par un ensemble fini Σ de symboles, un entier : son nombre de bandes, un ensemble fini d'états, qui contient deux états particuliers : un état initial et un état final, et une table de transition.

Comment calculer avec une telle machine ? On considère des machines dont l'ensemble de symboles Σ contient, outre le symbole blanc et la croix, un troisième symbole : le bâton |. Une fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$ peut se calculer avec une machine qui contient au moins $n+1$ bandes. Pour calculer la valeur de cette fonction en un n-uplet $p_1, \ldots, p_n$, on considère la configuration initiale dans laquelle la première bande contient une croix suivie de p_1 bâtons, la deuxième bande contient une croix, suivie de p_2 bâtons, ..., la n-ième bande contient une croix, suivie de p_n bâtons. Les autres bandes contiennent simplement une croix dans la première case. On fait démarrer la machine dans cette configuration, avec la tête à gauche et dans son état initial. La machine évolue alors petit pas après petit pas. Quand elle s'arrête, la $(n+1)$-ième bande doit contenir une croix suivie de q bâtons et les autres bandes doivent être identiques à ce qu'elles étaient dans la configuration initiale. L'entier q est $f(p_1, \ldots, p_n)$, c'est le résultat du calcul.

Le langage des machines de Turing entre dans le cadre que nous avons défini dans l'introduction de ce chapitre. Une fonction s'exprime par un entier k — le nombre de bandes —, un ensemble d'états et une table de transition. Quand on a un tel triplet (k, S, M) et des entiers $p_1, \ldots, p_n$, on peut agréger ces information en un terme qui est la machine à k bandes dont les n premières contiennent les entiers $p_1, \ldots, p_n$ et les autres simplement une croix et dont l'ensemble d'états est S et la table de transition M. Le pas élémentaire de calcul est une transition formée d'une opération de lecture, d'une opération d'écriture, d'un changement d'état et d'un mouvement de la tête.

On dit donc qu'une fonction est *représentable* par une machine de Turing s'il existe une machine qui la calcule.

Définissons, par exemple, une machine qui calcule le successeur d'un entier. Cette machine comporte deux bandes et trois états s_0 — l'état initial —, s_1 et

s_2 — l'état final. Elle commence par déplacer sa tête vers la droite en écrivant des bâtons sur la seconde bande tant qu'elle en trouve sur la première. Quand elle n'en trouve plus, elle écrit un bâton de plus, change d'état, ramène la tête à gauche et passe dans son état final. Sa table de transition est donc la suivante.

$M((\times,\times),s_0) = ((\times,\times),s_0,1)$
$M((|,b),s_0) = ((|,|),s_0,1)$
$M((b,b),s_0) = ((b,|),s_1,-1)$
$M((|,|),s_1) = ((|,|),s_1,-1)$
$M((\times,\times),s_1) = ((\times,\times),s_2,0)$

Naturellement, pour définir complètement la machine, il est nécessaire de compléter la table en indiquant quoi faire dans toutes les autres configurations. Celles-ci n'étant pas atteignables, peu importe la manière dont on complète cette table.

Pour montrer que toutes les fonctions calculables peuvent être calculées par une machine de Turing, on doit montrer que l'ensemble des fonctions calculables par une machine de Turing contient les projections, les fonctions nulles, la fonction successeur, l'addition, la multiplication, la fonction caractéristique de la relation d'ordre et qu'il est clos par composition et minimisation. Commençons par montrer la clôture par la composition qui demande de définir une manière de combiner des machines de Turing.

On remarque tout d'abord, que si une machine calcule une fonction f de $\mathbb{N}^n$ dans $\mathbb{N}$, il n'est pas difficile de la transformer de manière à ajouter des bandes neutres donc le contenu ne modifie pas l'évolution de la machine et sur laquelle la tête n'écrit jamais. Il n'est pas non plus difficile de transformer une telle machine de manière à ce qu'elle lise ses arguments sur des bandes $b_1,\ldots,b_n$, qui ne sont pas nécessairement les n premières bandes, et écrive son résultat sur une bande b_{n+1}, qui n'est pas nécessairement la $(n+1)$-ième bande.

Ces deux remarques faites, on peut alors montrer que l'ensemble des fonctions calculables par une machine de Turing est clos par composition. Soient h, g_1, …, g_m des fonctions calculées par des machines N et $M_1,\ldots,M_m$. Ces machines utilisent, outre les bandes permettant de lire les arguments et écrire le résultat, un certain nombre de bandes auxiliaires pour effectuer les calculs. On commence par les modifier de manière à ce qu'elles aient toutes $n+m+1+r$ bandes où r est le plus grand nombre de bandes auxiliaires utilisées par l'une des machines $M_1,\ldots,M_n,N$, que M_i lise ses arguments sur les bandes $1,\ldots,n$ et écrive son résultat sur la bande $n+1+i$, que N lise ses arguments sur les bandes $n+2,\ldots,n+m+1$ et écrive son résultat sur la bande $n+1$ et que les bandes auxiliaires utilisées par toutes ces machines soient au-delà des $n+m+1$ premières bandes.

On construit une machine en prenant comme ensemble d'états l'union disjointe des ensembles d'états de ces $m+1$ machines et d'un ensemble d'états

propres à cette machine, qui contient un état initial et un état final, et comme table la réunion des tables de toutes ces machines, qui est une fonction puisque ces tables sont des fonctions de domaines disjoints. On ajoute des transitions de manière à ce que, quand la machine est dans son état initial, elle effectue une transition vers l'état initial de M_1, quand elle est dans l'état final de M_1, elle effectue une transition vers l'état initial de M_2, ..., quand elle est dans l'état final de M_m, elle effectue une transition vers l'état initial de N et quand elle est dans l'état final de N, elle effectue une transition vers son propre état final. On obtient alors une machine qui, quand elle démarre avec les entiers $p_1, \ldots, p_n$ écrits sur ses n premières bandes, calcule $q_1 = g_1(p_1, \ldots, p_n)$, $q_2 = g_2(p_1, \ldots, p_n)$, ..., $q_m = g_m(p_1, \ldots, p_n)$ en écrivant les résultats sur les bandes $n+2, \ldots, m+n+1$ puis calcule $h(q_1, \ldots, q_m)$ en écrivant le résultat sur la bande $n+1$. Il suffit alors de la modifier de manière à ce qu'elle efface les entiers écrits sur les bandes $n+2, \ldots, n+m+1$ et ramène la tête à gauche, pour obtenir une machine qui calcule la composée de h et g_1, ..., g_m.

La machine calculant une fonction définie par minimisation se construit de manière similaire : elle commence par calculer $g(p_1, \ldots, p_n, 0)$ en imitant la machine qui calcule la fonction g, en lisant ses arguments sur les bandes $1, \ldots, n, n+1$ et en écrivant le résultat sur la bande $n+2$, si le deuxième élément de la bande $n+2$ est un blanc, alors la machine ramène la tête à gauche et passe dans son état final, sinon elle écrit un bâton sur la bande $n+1$ et recommence à calculer g en lisant ses arguments sur les bandes $1, \ldots, n, n+1$, ...

Nous avons déjà construit une machine qui calcule la fonction successeur. Une machine qui calcule une projection π_i^n se construit de manière similaire. Cette machine comporte $n+1$ bandes. Elle commence par déplacer sa tête vers la droite en écrivant des bâtons sur la bande $n+1$ tant qu'elle en trouve sur la bande i. Quand elle n'en trouve plus, elle ramène la tête à gauche et passe dans son état final.

Une machine qui calcule une fonction nulle est encore plus simple puisqu'il lui suffit de passer directement de son état initial à son état final.

Une machine qui calcule l'addition se construit ainsi. Cette machine commence par recopier le contenu de la première bande sur la quatrième et le contenu de la deuxième bande sur la troisième. Puis elle efface un a un les bâtons de la quatrième bande, de la droite vers la gauche, en ajoutant un bâton sur la troisième bande à chaque fois.

Une machine qui calcule la multiplication se construit de manière similaire. Elle recopie la première bande sur la quatrième. Puis elle efface un a un les bâtons de la quatrième bande en ajoutant à la troisième bande un nombre de bâtons égal au nombre de bâtons de la deuxième bande. Pour cela elle recopie le contenu de la deuxième bande sur la cinquième et elle efface un a un les

bâtons de la cinquième bande en écrivant un bâton sur la troisième à chaque fois.

Une machine qui calcule la fonction caractéristique de la relation d'ordre se construit ainsi. Tant que les deux premières lignes contiennent deux bâtons, elle avance sa tête vers la droite, si elle rencontre la configuration $(b, |)$ ou (b, b) elle passe dans un état s, si elle rencontre la configuration $(|, b)$ elle passe dans un état s'. Dans les deux cas, elle ramène la tête à gauche de bande, si elle est dans l'état s' elle passe alors dans son état final, si elle est dans l'état s, elle déplace sa tête à droite, écrit un bâton sur la troisième bande, déplace sa tête à gauche et passe dans son état final.

On peut donc conclure.

Théorème 4.3

Toutes les fonctions calculables sont représentables par une machine de Turing.

Cette proposition a une réciproque : toutes les fonctions représentables par une machine de Turing sont calculables. Il y a, en effet, de nombreuses manières de décrire l'état d'une machine de Turing, c'est-à-dire l'état des bandes, la position de la tête et son état, comme un arbre. On peut, par exemple, introduire une constante pour chaque état et une constante pour chaque élément de Σ. La partie des bandes située à gauche de la tête peut être représentée comme une liste de k-uplet — le premier élément de la liste étant le k-uplet des symboles situés juste à gauche de la tête — et, de même, la partie des bandes situées à droite de la tête comme une autre liste de k-uplets de symboles — le premier élément de la liste étant le k-uplet des symboles situés juste à droite de la tête. Un terme est donc un triplet formé d'un état, d'un k-uplet représentant les cases des bandes situées à la position de la tête et de deux listes de k-uplets représentant les parties gauches et droites des bandes. Ce états peuvent dont être numérotés. Il suffit ensuite de montrer que la fonction qui décrit un pas élémentaire de calcul est calculable.

Exercice 4.10

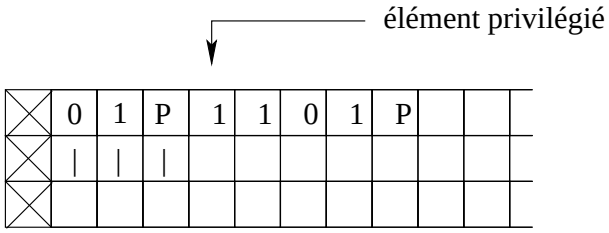
Donner une démonstration directe du fait que l'ensemble des fonctions calculables par une machine de Turing est clos par définition par récurrence.

Exercice 4.11

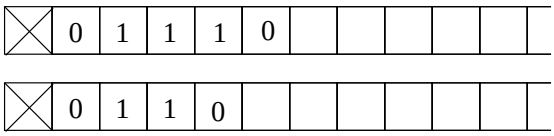
Numéroter les arbres est une bonne idée quand on s'intéresse à l'existence des algorithmes, mais non quand on s'intéresse à leur complexité, car ces opérations

de codage perturbent cette complexité. On considère donc, dans cet exercice, des machines de Turing qui calculent directement avec des arbres. Pour cela, on considère un ensemble de symboles Σ, qui contient, outre le symbole blanc et la croix, un nombre arbitraire d'autres symboles. On peut alors représenter un arbre, appartenant à un ensemble articulé, en notation préfixe ou postfixe sur une bande d'une machine de Turing.

Si la première bande d'une machine contient une suite de symboles $u_0, u_1, \ldots$ et la deuxième bande contient un entier k représenté par k bâtons, l'ensemble des deux bandes décrit une *suite pointée de symboles*, c'est-à-dire la suite $u_0, u_1, \ldots$ dans laquelle l'élément u_k est privilégié.



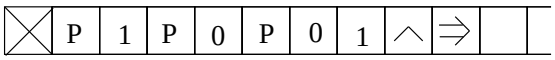
1. Montrer qu'il existe une machine de Turing qui lit un entier n, écrit avec n bâtons, sur la première bande et écrit $2n$ bâtons sur la deuxième bande.
2. Montrer qu'il existe une machine de Turing qui lit un entier n écrit en binaire, le chiffre de poids faible en tête, sur la première bande, à partir de la position k indiquée par le nombre de bâtons de la deuxième bande, qui écrit n bâtons sur la troisième bande et qui ajoute des bâtons sur la deuxième bande, jusqu'au premier élément de la première qui n'est pas un chiffre binaire.
3. Montrer qu'il existe une machine de Turing qui lit une suite de 0 et de 1 sur la première bande, supprime les deux derniers symboles et écrit à la fin de la suite obtenue un 1 si ces deux derniers symboles étaient des 1 et 0 sinon.



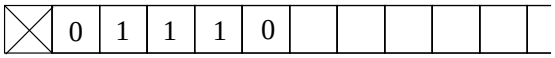
La *logique des propositions* est le fragment de la logique des prédicats dont les propositions sont construites avec des symboles de prédicat sans arguments, appelés *symboles de proposition*, et les symboles $\top$, $\bot$, $\neg$, $\wedge$, $\vee$ et $\Rightarrow$. Un exemple est la proposition

$$P_1 \Rightarrow (P_0 \wedge P_2)$$

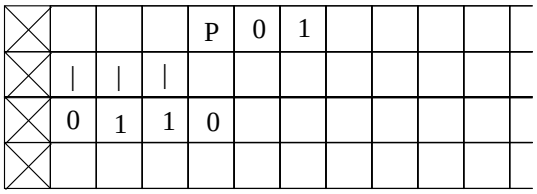
On appelle les symboles de proposition P_0, P_1, ..., P_n où les indices des symboles apparaissant dans une proposition forment un segment initial de l'ensemble des entiers. On écrit une telle proposition sur une bande sous forme postfixe en écrivant les indices en binaire, le chiffre de poids faible en tête. Ainsi, la proposition ci-avant s'écrit

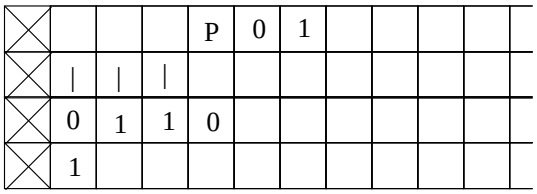


Un modèle d'une telle proposition est simplement une fonction qui associe une valeur de vérité 0 ou 1 à chaque symbole de proposition. On écrit un tel modèle sur une bande d'une machine de Turing en écrivant successivement la valeur de vérité associée à P_0, P_1, P_2, ... par exemple

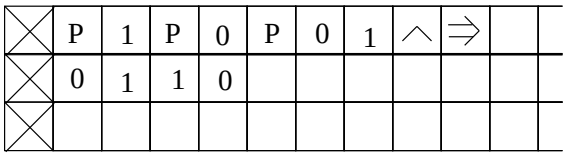


4. Montrer qu'il existe une machine de Turing qui lit un symbole de proposition P_n écrit sur la première bande à partir de la position k indiquée par le nombre de bâtons de la deuxième bande et un modèle sur la troisième bande et écrit la valeur de vérité de ce symbole de proposition sur la quatrième.





5. Montrer qu'il existe une machine de Turing qui lit une proposition sur la première bande et un modèle sur la deuxième et écrit la valeur de vérité de cette proposition sur la troisième.



	P	1	P	0	P	0	1	$\wedge$	$\Rightarrow$		
	0	1	1	0							
	0										

6. Soit n le nombre de symboles de la proposition, montrer que tous les indices des symboles de proposition sont majorés par n. Montrer que la longueur de la représentation de la proposition sur une bande d'une machine de Turing est comprise entre n et $n(2 + \log_2(n))$.

   Soit E un ensemble d'arbres étiquetés par les éléments d'un ensemble fini. On dit que l'ensemble E est *dans la classe* P s'il existe une machine de Turing qui termine toujours telle que
   - pour tout arbre a, a appartient à E si et seulement si l'exécution de la machine sur l'arbre a donne le résultat 1 et
   - il existe un polynôme f tel que le nombre d'étapes de l'exécution de cette machine sur un arbre de taille p soit majoré par $f(p)$.

7. Montrer que l'ensemble des couples formés d'une proposition A et d'un modèle $\mathcal{M}$ tels que A soit valide dans $\mathcal{M}$ est dans la classe P.

Exercice 4.12

Cet exercice demande d'avoir fait l'exercice 4.11.

On étend la notion de machine de Turing en une notion de *machine de Turing non déterministe*. Pour chaque couple formé d'une suite de symboles et d'un état, la table d'une machine de Turing ordinaire prescrit une transition unique consistant à écrire une suite de symboles, évoluer vers un nouvel état et effectuer un mouvement de la tête. En revanche, pour chaque couple, la table d'une machine de Turing non déterministe spécifie un ensemble fini de transitions possibles.

La table d'une telle machine n'est donc pas une fonction qui à tout élément de $\Sigma^k \times S$ associe un élément de $\Sigma^k \times S \times \{-1, 0, +1\}$, mais une fonction qui, à tout élément de $\Sigma^k \times S$, associe une partie finie et non vide de $\Sigma^k \times S \times \{-1, 0, +1\}$.

Une configuration initiale d'une machine ordinaire détermine une unique suite de transitions, une configuration initiale d'une machine non déterministe détermine, en revanche, un ensemble de suites de transitions où, à chaque étape, la machine effectue l'une des transitions spécifiées par la table. Ces différentes suites de transitions mènent à différents résultats. Une machine de Turing non déterministe définit donc une fonction qui à des arbres $p_1, \ldots, p_n$ associe, non un arbre, mais un ensemble d'arbres.

Soit E un ensemble d'arbres étiquetés par les éléments d'un ensemble fini.

On dit que l'ensemble E est *dans la classe* NP s'il existe une machine de Turing non déterministe qui termine toujours telle que

- pour tout arbre a, a appartient à E si et seulement si *l'une des* suites de transitions de la machine sur l'arbre a donne le résultat 1 et
- il existe un polynôme f tel que la longueur de toutes les suites de transitions de cette machine sur un arbre de taille p soit majorée par $f(p)$.

Montrer que l'ensemble SAT des propositions de la logique des propositions définie à l'exercice 4.11 qui sont cohérentes — on dit aussi *satisfiables* —, c'est-à-dire qui ont un modèle, est dans la classe NP.

La notion de calculabilité présente donc une certaine robustesse, puisque les fonctions définissables dans des langages aussi divers que celui des machines de Turing, le lambda-calcul ou la réécriture sont les mêmes : ce sont les fonctions calculables.

Toutefois, cette diversité apparente des langages masque en réalité une profonde unité : dans tous ces langages, l'exécution d'un calcul est définie comme une suite de petits pas.

Troisième partie

Les démonstrations et les algorithmes

5
Le théorème de Church

Une partie importante de l'activité mathématique consiste à concevoir des algorithmes qui permettent de résoudre des problèmes, par exemple calculer le plus grand diviseur commun de deux entiers, la solution d'un système linéaire ou la primitive d'une fonction polynomiale, sans construire de démonstration. Jusqu'à quel point est-il possible de remplacer la recherche d'une démonstration par l'exécution d'un algorithme ? Un ensemble de résultats, tant négatifs que positifs, précise ce qu'il est possible de faire.

Ce chapitre est consacré à deux résultats, l'un négatif et l'autre positif, qui montrent que l'ensemble des propositions démontrables dans la logique des prédicats n'est pas décidable, mais qu'il est semi-décidable.

Comme tous les arbres, les propositions peuvent se numéroter. Ces résultats montrent donc, plus précisément, que l'ensemble des numéros des propositions démontrables dans la logique des prédicats n'est pas décidable, mais qu'il est semi-décidable.

5.1 La notion de réduction

Commençons par montrer que l'ensemble des propositions démontrables dans la logique des prédicats n'est pas décidable. L'idée de la démonstration peut se formuler en une phrase : le fait qu'un programme f termine peut s'exprimer par la proposition « Le programme f termine » et, puisqu'il est impossible de décider la terminaison des programmes, il est impossible de décider

la démontrabilité des propositions de ce type et donc de la démontrabilité des propositions en général.

Qu'est-ce que la proposition : « Le programme f termine » ? Répondre à cette question consiste à associer à chaque programme f, une proposition qui est démontrable si et seulement si le programme f termine. Comme nous allons le voir, la fonction T qui, au programme f, associe la proposition « Le programme f termine » est calculable. De ce fait, s'il existait une fonction calculable F pour décider si une proposition est démontrable ou non, la fonction $F \circ T$ serait calculable en contradiction avec le théorème d'indécidabilité du problème de l'arrêt.

On voit apparaître ici une méthode assez générale pour montrer qu'un problème est indécidable : construire un algorithme qui permet de réduire, au problème en question, un problème déjà démontré indécidable, dans cet exemple, le problème de l'arrêt. Cela peut se formuler de manière abstraite comme le fait que, l'ensemble des fonctions calculables étant clos par composition, si T est calculable et $F \circ T$ ne l'est pas, alors F ne l'est pas non plus.

5.2 La représentation des programmes

On commence par associer à chaque programme f d'arité n une proposition A de l'arithmétique dont les variables libres sont parmi $x_1, \ldots, x_n, y$ qui *représente* le programme f. Cela signifie que la proposition $(\underline{p_1}/x_1, \ldots, \underline{p_n}/x_n, \underline{q}/y)A$, où $\underline{p}$ est le terme $S^p(0)$, est démontrable si et seulement si f prend la valeur q en $p_1, \ldots, p_n$. Pour simplifier les notations, on écrit $A[t_1, \ldots, t_n, u]$ la proposition $(t_1/x_1, \ldots, t_n/x_n, u/y)A$.

Si $f = \pi_i^n$, on peut poser $A = (y = x_i)$. Si $f = Z^n$, $A = (y = 0)$. Si $f = Succ$, $A = (y = S(x_1))$. Si $f = +$, $A = (y = x_1 + x_2)$. Si $f = \times$, $A = (y = x_1 \times x_2)$. Si $f = \chi_{\leq}$, $A = (x_1 \leq x_2 \wedge y = 1) \vee (x_2 < x_1 \wedge y = 0)$ où la proposition $x \leq y$ est une abréviation pour $\exists z\ (z + x = y)$ et la proposition $x < y$ pour $S(x) \leq y$.

Si $f = \circ_m^n(h, g_1, \ldots, g_m)$, on commence par construire des propositions B_1, $\ldots$, B_m et C représentant les programmes g_1, $\ldots$, g_m et h et on pose

$$A = \exists w_1 \ldots \exists w_m\ (B_1[x_1, \ldots, x_n, w_1] \wedge \ldots \wedge B_m[x_1, \ldots, x_n, w_m] \wedge C[w_1, \ldots, w_m, y])$$

Enfin, si $f = \mu^n(g)$, on commence par construire une proposition B représentant le programme g et on pose

$$A = (\forall z\ (z < y \Rightarrow \exists w\ (B[x_1, \ldots, x_n, z, S(w)]))) \wedge B[x_1, \ldots, x_n, y, 0]$$

Nous pourrions alors démontrer que f prend la valeur q en $p_1, \ldots, p_n$ si et seulement si la proposition $A[\underline{p_1}, \ldots, \underline{p_n}, \underline{q}]$ est démontrable dans l'arithmétique, et conclure que la démontrabilité dans l'arithmétique est indécidable.

Cependant, avant d'entreprendre cette démonstration, nous allons étendre quelque peu la définition ci-avant. Celle-ci suppose, en effet, que le langage contient des symboles 0, S, $+$, $\times$ et $=$ et également que l'univers du discours est limité aux entiers. Ces deux hypothèses sont vérifiées dans le cas de l'arithmétique, mais elles seront des obstacles, quand nous voudrons généraliser ce résultat à d'autres théories. Par exemple, nous avons vu que dans le langage de la théorie des ensembles, il n'y a pas de symbole S pour le successeur, mais il y a une proposition, contentant deux variables libres x et y, qui exprime le fait que y est le successeur de x

$$\forall z\ (z \in y \Leftrightarrow (z \in x \vee z = x))$$

Nous considérons donc un langage quelconque, dans lequel il est possible de construire des propositions N, *Null*, *Succ*, *Plus*, *Mult* et *Eq*. Nous écrivons $N[t]$ la proposition $(t/x)N$, $Succ[t,u]$ la proposition $(t/x,u/y)Succ$, ... Par exemple, dans l'arithmétique, N est la proposition $\top$, *Null* la proposition $x = 0$, *Succ* la proposition $y = S(x)$, *Plus* la proposition $z = x + y$, *Mult* la proposition $z = x \times y$ et *Eq* la proposition $x = y$. En théorie des ensembles, la proposition N est celle construite dans l'exercice 1.17, la proposition *Succ* est la proposition $\forall z\ (z \in y \Leftrightarrow (z \in x \vee z = x))$, ...

La proposition *Inf*, *relation d'ordre*, se définit comme $\exists z\ (N[z] \wedge Plus[z,x,y])$ et *InfS*, *relation d'ordre strict*, comme $\exists x'\ (N[x'] \wedge Succ[x,x'] \wedge Inf[x',y])$.

Dans ce langage, nous associons une proposition à chaque programme.

Définition 5.1 (Proposition représentant un programme)

Soit f un programme d'arité n, la proposition A *représentant* f est définie par récurrence sur la construction de f.

- Si $f = \pi_i^n$, on pose $A = Eq[x_i, y]$.
- Si $f = Z^n$, on pose $A = Null[y]$.
- Si $f = Succ$, on pose $A = Succ[x_1, y]$.
- Si $f = +$, on pose $A = Plus[x_1, x_2, y]$.
- Si $f = \times$, on pose $A = Mult[x_1, x_2, y]$.
- Si $f = \chi_{\leq}$, on pose $A = (Inf[x_1, x_2] \wedge \exists z\ (Null[z] \wedge Succ[z, y])) \vee (InfS[x_2, x_1] \wedge Null[y])$.
- Si $f = \circ_m^n(h, g_1, \ldots, g_m)$, alors soient B_1, ..., B_m et C les propositions représentant les programmes g_1, ..., g_m et h, on pose

  $$A = \exists w_1 \ldots \exists w_m\ (N[w_1] \wedge \ldots \wedge N[w_m] \wedge B_1[x_1, \ldots, x_n, w_1] \wedge \ldots \wedge B_m[x_1, \ldots, x_n, w_m] \wedge C[w_1, \ldots, w_m, y]).$$
- Si $f = \mu^n(g)$, alors soit B la proposition représentant le programme g, on pose

$$A = (\forall z\ (N[z] \wedge \mathit{InfS}[z,y] \Rightarrow \exists w \exists w'\ (N[w'] \wedge \mathit{Succ}[w',w] \wedge B[x_1,\ldots,x_n,z,w]))) \wedge (\forall w\ (\mathit{Null}[w] \Rightarrow B[x_1,\ldots,x_n,y,w])).$$

Nous voulons alors démontrer que f prend la valeur q en $p_1,\ldots,p_n$ si et seulement si la proposition A relie les nombres $p_1,\ldots,p_n,q$. Cependant, nous ne pouvons plus exprimer cela simplement en substituant aux variables des termes de la forme $S^p(0)$ dans la proposition A, car nous ne disposons plus nécessairement des symboles 0 et S. Nous devons alors introduire, pour chaque entier, une proposition N_n qui caractérise l'entier n, en posant $N_0 = \mathit{Null}[x]$ et $N_{n+1} = \exists y\ (N_n[y] \wedge \mathit{Succ}[y,x])$. Si une proposition A contient potentiellement une variable libre x, nous pouvons exprimer le fait que la propriété exprimée par A s'applique à l'entier n par la proposition $\forall x\ (N_n[x] \Rightarrow A)$.

Nous pouvons alors démontrer que f prend la valeur q en $p_1,\ldots,p_n$ si et seulement si la proposition

$$\forall x_1 \ldots \forall x_n \forall y\ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1,\ldots,x_n,y])$$

est démontrable. Naturellement, pour démontrer cette proposition, nous avons besoin de quelques propriétés des propositions N, *Null*, *Succ*, *Plus*, *Mult* et *Eq*. De façon surprenante, peu de propriétés suffisent. Ces propriétés sont rassemblées dans la théorie $\mathcal{T}_0$ suivante.

Définition 5.2 (La théorie $\mathcal{T}_0$)

La théorie $\mathcal{T}_0$ est formée des axiomes suivants.
Prédicat N :

$$\forall x\ (\mathit{Null}[x] \Rightarrow N[x])$$

$$\forall x \forall y\ ((N[x] \wedge \mathit{Succ}[x,y]) \Rightarrow N[y])$$

Existence des entiers :

$$\exists x\ \mathit{Null}[x]$$

$$\forall x\ (N[x] \Rightarrow \exists y\ \mathit{Succ}[x,y])$$

Égalité :

$$\forall x\ \mathit{Eq}[x,x]$$

$$\forall x \forall y\ (\mathit{Null}[x] \Rightarrow (\mathit{Null}[y] \Leftrightarrow \mathit{Eq}[x,y]))$$

$$\forall x \forall y \forall x' \forall y'\ ((N[x] \wedge \mathit{Succ}[x,x'] \wedge \mathit{Eq}[x,y]) \Rightarrow (\mathit{Succ}[y,y'] \Leftrightarrow \mathit{Eq}[x',y']))$$

Injectivité du successeur :

$$\forall x \forall y \forall x' \forall y'\ (\mathit{Succ}[x,x'] \wedge \mathit{Succ}[y,y'] \wedge \mathit{Eq}[x',y']) \Rightarrow \mathit{Eq}[x,y])$$

Un successeur est non nul :

$$\forall x \forall x' \ (Succ[x, x'] \Rightarrow \neg Null[x'])$$

Tout entier est nul ou un successeur :

$$\forall x \ (N[x] \Rightarrow (Null[x] \vee \exists y \ (N[y] \wedge Succ[y, x])))$$

Addition :

$$\forall x \forall y \forall z \ ((Null[x] \wedge N[y]) \Rightarrow (Eq[y, z] \Leftrightarrow Plus[x, y, z]))$$

$$\forall x \forall y \forall z \forall x' \forall z' \ ((N[x] \wedge N[y] \wedge Plus[x, y, z] \wedge Succ[x, x']) \Rightarrow (Succ[z, z'] \Leftrightarrow Plus[x', y, z']))$$

$$\forall x \forall y \forall x' \forall y' \ ((N[x] \wedge N[y] \wedge Succ[x, x'] \wedge Succ[z, z'] \wedge Plus[x', y, z']) \Rightarrow Plus[x, y, z])$$

$$\forall x \forall y \forall y' \forall z' \ ((N[x] \wedge N[y] \wedge N[y'] \wedge Succ[y, y'] \wedge Plus[x, y', z']) \Rightarrow \exists z \ (Plus[x, y, z] \wedge Succ[z, z']))$$

Multiplication :

$$\forall x \forall y \forall z \ ((Null[x] \wedge N[y]) \Rightarrow (Null[z] \Leftrightarrow Mult[x, y, z]))$$

$$\forall x \forall y \forall z \forall x' \forall z' \ ((N[x] \wedge N[y] \wedge Mult[x, y, z] \wedge Succ[x, x']) \Rightarrow (Plus[y, z, z'] \Leftrightarrow Mult[x', y, z'])$$

Proposition 5.1

Les propositions suivantes sont démontrables dans la théorie $\mathcal{T}_0$.

1. $\forall x \ (N_p[x] \Rightarrow N[x])$
2. $\exists x \ N_p[x]$
3. $\forall x \forall y \ (N_p[x] \Rightarrow (N_p[y] \Leftrightarrow Eq[x, y]))$
4. $\forall x \forall y \forall z \ ((N_p[x] \wedge N_q[y]) \Rightarrow (N_{p+q}[z] \Leftrightarrow Plus[x, y, z]))$
5. $\forall x \forall y \forall z \ ((N_p[x] \wedge N_q[y]) \Rightarrow (N_{p \times q}[z] \Leftrightarrow Mult[x, y, z]))$
6. $\forall x_1 \forall x_2 \ ((N_{p_1}[x_1] \wedge N_{p_2}[x_2]) \Rightarrow Inf[x_1, x_2])$, où $p_1 \leq p_2$
7. $\forall x_1 \forall x_2 \ ((N_{p_1}[x_1] \wedge N_{p_2}[x_2]) \Rightarrow InfS[x_1, x_2])$, où $p_1 < p_2$
8. $\forall x \forall y \forall y' \ ((N[x] \wedge Inf[x, y'] \wedge Succ[y, y']) \Rightarrow (Eq[x, y'] \vee Inf[x, y]))$
9. $\forall x \forall y \forall y' \ ((N[x] \wedge InfS[x, y'] \wedge Succ[y, y']) \Rightarrow (Eq[x, y] \vee InfS[x, y]))$
10. $\forall x \forall y \ ((N[x] \wedge InfS[x, y]) \Rightarrow \neg Null[y])$

Démonstration.

1. Par récurrence sur p, en utilisant les axiomes du prédicat N.
2. Par récurrence sur p, en utilisant les axiomes d'existence des entiers et (1.).
3. Par récurrence sur p, en utilisant les axiomes de l'égalité et (1.).

4. Par récurrence sur p, en utilisant les deux premiers axiomes de l'addition, (1.), (2.) et (3.).
5. Par récurrence sur p, en utilisant les axiomes de la multiplication, (1.), (2.) et (4.).
6. Comme $p_1 \leq p_2$, il existe un entier q tel que $q + p_1 = p_2$. Des hypothèses $N_q[z]$, $N_{p_1}[x_1]$ et $N_{p_2}[x_2]$, on déduit, en utilisant (1.) et (4.), les propositions $N[z]$ et $Plus[z, x_1, x_2]$ et donc $Inf[x_1, x_2]$. On élimine ensuite l'hypothèse $N_q[z]$ avec (2.).
7. On a $p_1 < p_2$, et donc $p_1 + 1 \leq p_2$. Des hypothèses $N_{p_1}[x_1]$, $Succ[x_1, w]$ et $N_{p_2}[x_2]$, on déduit $N_{p_1+1}[w]$ puis, en utilisant (1.) et (6.), les propositions $N[w]$ et $Inf[w, x_2]$ et donc la proposition $InfS[x_1, x_2]$. On élimine ensuite l'hypothèse $Succ[x_1, w]$ en utilisant $\exists w\ Succ[x_1, w]$, qui se montre avec le second axiome d'existence des entiers.
8. La proposition $Inf[x, y']$ est $\exists z'\ (N[z'] \wedge Plus[z', x, y'])$. On utilise l'axiome *Tout entier est nul ou un successeur* pour distinguer le cas où z' est nul et celui où c'est le successeur d'un entier z. Dans le premier cas, le premier axiome de l'addition donne $Eq[x, y']$. Dans le second, le troisième axiome de l'addition donne $Plus[z, x, y]$ et donc $Inf[x, y]$.
9. Conséquence de (8.) et de l'axiome *Injectivité du successeur*.
10. Conséquence du quatrième axiome de l'addition et de l'axiome *Un successeur est non nul*.

Proposition 5.2

Soit A une proposition. On écrit $A[t]$ la proposition $(t/x)A$. Si les propositions $\forall x\ (N_0[x] \Rightarrow A[x])$, $\forall x\ (N_1[x] \Rightarrow A[x])$, $\ldots$, $\forall x\ (N_{p-1}[x] \Rightarrow A[x])$ sont démontrables dans la théorie $\mathcal{T}_0$, alors c'est aussi le cas de la proposition $\forall x \forall y\ ((N[x] \wedge N_p[y] \wedge InfS[x, y]) \Rightarrow A[x])$.

Démonstration. Par récurrence sur p en utilisant la proposition 5.1 (9.).

Définition 5.3 ($\mathbb{N}$-modèle)

Soit $\mathcal{L}$ un langage et N, *Null*, *Succ*, *Plus*, *Mult* des propositions de ce langage. Un modèle $\mathcal{M}$ de ce langage est un $\mathbb{N}$-*modèle* si

$$\{a \in \mathcal{M} \mid [\![N]\!]_{x=a} = 1\} = \mathbb{N}$$

$$\{a \in \mathbb{N} \mid [\![Null]\!]_{x=a} = 1\} = \{0\}$$

$$\{(a, b) \in \mathbb{N}^2 \mid [\![Succ]\!]_{x=a, y=b} = 1\} = \{(a, b) \in \mathbb{N}^2 \mid b = a + 1\}$$

$$\{(a,b,c) \in \mathbb{N}^3 \mid [\![Plus]\!]_{x=a,y=b,z=c} = 1\} = \{(a,b,c) \in \mathbb{N}^3 \mid c = a+b\}$$
$$\{(a,b,c) \in \mathbb{N}^3 \mid [\![Mult]\!]_{x=a,y=b,z=c} = 1\} = \{(a,b,c) \in \mathbb{N}^3 \mid c = a \times b\}$$
$$\{(a,b) \in \mathbb{N}^2 \mid [\![Eq]\!]_{x=a,y=b} = 1\} = \{(a,b) \in \mathbb{N}^2 \mid a = b\}$$

Si $\mathcal{T}$ est une théorie dans le langage $\mathcal{L}$, un $\mathbb{N}$-modèle de $\mathcal{T}$ est un $\mathbb{N}$-modèle de $\mathcal{L}$ qui est, par ailleurs, un modèle de $\mathcal{T}$.

Les axiomes de la théorie $\mathcal{T}_0$ sont valides dans tous les $\mathbb{N}$-modèles.

On peut alors démontrer la proposition suivante.

Proposition 5.3

Soit $\mathcal{L}$ un langage, N, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dans ce langage qui démontre au moins les axiomes de la théorie $\mathcal{T}_0$ et qui a un $\mathbb{N}$-modèle $\mathcal{M}$. Alors, si f est un programme et A la proposition représentant f, les trois propositions suivantes sont équivalentes

- f prend la valeur q en $p_1, \ldots, p_n$,
- la proposition

$$\forall x_1 \ldots \forall x_n \forall y \; ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

  est démontrable dans la théorie $\mathcal{T}$,
- cette proposition est valide dans le modèle $\mathcal{M}$.

Démonstration. On suppose que f prend la valeur q en $p_1, \ldots, p_n$ et on montre, par récurrence sur la structure de f, que la proposition

$$\forall x_1 \ldots \forall x_n \forall y \; ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

est démontrable dans $\mathcal{T}$.

- Si $f = \pi_i^n$ alors $A = Eq[y, x_i]$ et la proposition

$$\forall x_1 \ldots \forall x_n \forall y \; ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_{p_i}[y]) \Rightarrow Eq[x_i, y])$$

  est une conséquence de la proposition 5.1 (3.). On procède de même pour le programme zéro, le successeur, l'addition et la multiplication en utilisant la proposition 5.1 (4.) et (5.).
- Si $f = \chi_{\leq}$, et $p_1 \leq p_2$, alors, d'après la proposition 5.1 (6.), la proposition $Inf[x_1, x_2]$ est démontrable sous les hypothèses $N_{p_1}[x_1]$ et $N_{p_2}[x_2]$. La proposition $\exists z' \; (Null[z'] \wedge Succ[z', y])$, quant à elle, est démontrable sous l'hypothèse $N_1[y]$ et donc la proposition A est démontrable sous les hypothèses $N_{p_1}[x_1]$, $N_{p_2}[x_2]$ et $N_1[y]$. On procède de même en utilisant la proposition 5.1 (7.) dans le cas où $p_2 < p_1$.

- Si $f = \circ_m^n(h, g_1, \ldots, g_m)$, alors comme f termine en $(p_1, \ldots, p_n)$, g_1, ..., g_m terminent en $(p_1, \ldots, p_n)$ et si on appelle r_i le nombre $g_i(p_1, \ldots, p_n)$, h termine en $r_1, \ldots, r_m$ et $q = h(r_1, \ldots, r_m)$. Soient $B_1, \ldots, B_m$ et C les propositions représentant les programmes $g_1, \ldots, g_m$ et h. Par hypothèse de récurrence, sous les hypothèses $N_{p_1}[x_1]$, ..., $N_{p_n}[x_n]$, $N_{r_1}[w_1]$, ..., $N_{r_m}[w_m]$ et $N_q[y]$, les propositions $B_1[x_1, \ldots, x_n, w_1]$, ..., $B_m[x_1, \ldots, x_n, w_m]$ et $C[w_1, \ldots, w_m, y]$ sont démontrables et d'après la proposition 5.1 (1.), sous ces mêmes hypothèses, les propositions $N[w_1]$, ..., $N[w_n]$ sont démontrables. On en déduit que la proposition A est démontrable. On élimine les hypothèses $N_{r_1}[w_1]$, ..., $N_{r_m}[w_m]$ en utilisant la proposition 5.1 (2.).
- Si $f = \mu^n(g)$ alors, comme f termine en $(p_1, \ldots, p_n)$ et vaut q, g termine en $(p_1, \ldots, p_n, 0)$, ..., $(p_1, \ldots, p_n, q-1)$ et prend une valeur non nulle et g est définie en $(p_1, \ldots, p_n, q)$ et prend la valeur 0. Il existe donc des entiers $r_0, \ldots, r_{q-1}$ tels que $g(p_1, \ldots, p_n, 0) = r_0 + 1$, ..., $g(p_1, \ldots, p_n, q-1) = r_{q-1} + 1$.
  Soit B la proposition représentant le programme g. Par hypothèse de récurrence, sous les hypothèses $N_{p_1}[x_1]$, ..., $N_{p_n}[x_n]$ et $N_q[y]$, pour tout i compris entre 0 et $q-1$, les propositions

$$\forall v \forall w \ ((N_i[v] \wedge N_{r_i+1}[w]) \Rightarrow B[x_1, \ldots, x_n, v, w])$$

  sont démontrables. En utilisant la proposition 5.1 (1.) et (2.), on en déduit que la proposition

$$\forall v \ (N_i[v] \Rightarrow \exists w \exists w' \ (N[w'] \wedge \mathit{Succ}[w', w] \wedge B[x_1, \ldots, x_n, v, w]))$$

  est démontrable. Avec la proposition 5.2, on en déduit que la proposition

$$\forall v \ (\mathit{InfS}[v, y] \Rightarrow \exists w \exists w' \ (N[w'] \wedge \mathit{Succ}[w', w] \wedge B[x_1, \ldots, x_n, v, w]))$$

  est démontrable. De même la proposition

$$\forall w \ (\mathit{Null}[w] \Rightarrow B[x_1, \ldots, x_n, y, w])$$

  est démontrable. On en conclut que la proposition A est démontrable.

Le modèle $\mathcal{M}$ étant un modèle de la théorie $\mathcal{T}$, si la proposition

$$\forall x_1 \ldots \forall x_n \forall y \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

est démontrable dans $\mathcal{T}$ elle est valide dans le modèle $\mathcal{M}$.

Enfin, si la proposition

$$\forall x_1 \ldots \forall x_n \forall y \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

est valide dans $\mathcal{M}$, il existe des entiers $p_1, \ldots, p_n, q$ tels que

$$[\![A[x_1, \ldots, x_n, y]]\!]_{x_1=p_1,\ldots,x_n=p_n,y=q} = 1$$

et, en utilisant le fait que $\mathcal{M}$ est un $\mathbb{N}$-modèle, on montre, par récurrence sur la structure de f, que f prend la valeur q en $p_1, \ldots, p_n$.

5.3 Le théorème de Church

Définition 5.4

Soit f un programme et A la proposition représentant ce programme. La proposition « Le programme f termine en $p_1, \ldots, p_n$ » est la proposition close

$$\forall x_1 \ldots \forall x_n \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n]) \Rightarrow \exists y \ (N[y] \wedge A[x_1, \ldots, x_n, y]))$$

Proposition 5.4

Soit $\mathcal{L}$ un langage, N, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dans ce langage qui démontre au moins les axiomes de la théorie $\mathcal{T}_0$ et qui a un $\mathbb{N}$-modèle $\mathcal{M}$. Alors, si f est un programme, les trois propositions suivantes sont équivalentes

- le programme f termine en $p_1, \ldots, p_n$,
- la proposition « Le programme f termine en $p_1, \ldots, p_n$ » est démontrable dans $\mathcal{T}$,
- cette proposition est valide dans $\mathcal{M}$.

Démonstration. Si le programme f termine en $p_1, \ldots, p_n$ alors il existe un entier q tel que f prenne la valeur q en $p_1, \ldots, p_n$. D'après la proposition 5.3, la proposition

$$\forall x_1 \ldots \forall x_n \forall y \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

est démontrable dans $\mathcal{T}$. On en déduit, en utilisant la proposition 5.1 (1.) et (2.), que la proposition $\forall x_1 \ldots \forall x_n \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n]) \Rightarrow \exists y \ (N[y] \wedge A[x_1, \ldots, x_n, y]))$ est démontrable dans $\mathcal{T}$.

Le modèle $\mathcal{M}$ étant un modèle de la théorie $\mathcal{T}$, si cette proposition est démontrable dans $\mathcal{T}$ elle est valide dans le modèle $\mathcal{M}$.

Enfin si cette proposition est valide dans $\mathcal{M}$, il existe un entier q tel que

$$[\![A[x_1, \ldots, x_n, y]]\!]_{x_1=p_1,\ldots,x_n=p_n,y=q} = 1$$

La proposition

$$\forall x_1 \ldots \forall x_n \forall y \ ((N_{p_1}[x_1] \wedge \ldots \wedge N_{p_n}[x_n] \wedge N_q[y]) \Rightarrow A[x_1, \ldots, x_n, y])$$

est donc valide dans $\mathcal{M}$. D'après la proposition 5.3, f prend la valeur q en $p_1, \ldots, p_n$ et termine donc en $p_1, \ldots, p_n$.

Proposition 5.5

La fonction T associant au numéro d'un programme f et à $p_1, \ldots, p_n$, le numéro de la proposition « Le programme f termine en $p_1, \ldots, p_n$ » et prenant la valeur 0 sur les entiers qui ne sont pas le numéro d'un programme est calculable.

Démonstration. Cette fonction est définie par récurrence bien fondée.

Nous obtenons ainsi un premier résultat d'indécidabilité de la démontrabilité.

Proposition 5.6

Soit $\mathcal{L}$ un langage, *N*, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dans ce langage qui démontre au moins les axiomes de la théorie $\mathcal{T}_0$ et qui a un $\mathbb{N}$-modèle. Alors l'ensemble des propositions closes de $\mathcal{L}$ démontrables dans $\mathcal{T}$ est indécidable.

Démonstration. S'il existait une fonction calculable F associant 1 ou 0 au numéro d'une proposition selon que cette proposition est démontrable dans $\mathcal{T}$ ou non, la fonction $F \circ T$ serait calculable, en contradiction avec le théorème d'indécidabilité du problème de l'arrêt.

Nous montrons ensuite que l'hypothèse, selon laquelle la théorie $\mathcal{T}$ doit démontrer les axiomes de la théorie $\mathcal{T}_0$, est superflue.

Proposition 5.7

Soit $\mathcal{L}$ un langage, *N*, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dans ce langage qui a un $\mathbb{N}$-modèle. Alors, l'ensemble des propositions closes de $\mathcal{L}$ démontrables dans la théorie $\mathcal{T}$ est indécidable.

Démonstration. La théorie $\mathcal{T} \cup \mathcal{T}_0$ démontre les axiomes de $\mathcal{T}_0$ et elle a un $\mathbb{N}$-modèle. D'après la proposition 5.6, la démontrabilité dans cette théorie est donc indécidable.

On utilise ensuite, encore une fois, une réduction. Soit H la conjonction des axiomes de la théorie $\mathcal{T}_0$. La proposition $H \Rightarrow A$ est démontrable dans la théorie $\mathcal{T}$ si et seulement si la proposition A est démontrable dans la théorie $\mathcal{T} \cup \mathcal{T}_0$. Soit T la fonction qui associe le numéro de la proposition $H \Rightarrow A$ au numéro de la proposition A. La fonction T est calculable et la proposition de numéro $T(\ulcorner A \urcorner)$ est démontrable dans la théorie $\mathcal{T}$ si et seulement si la proposition A est démontrable dans $\mathcal{T} \cup \mathcal{T}_0$. S'il existait un algorithme de décision F pour la théorie $\mathcal{T}$, $F \circ T$ serait un algorithme de décision pour $\mathcal{T} \cup \mathcal{T}_0$ en contradiction avec le fait que $\mathcal{T} \cup \mathcal{T}_0$ est indécidable.

Proposition 5.8

Soit $\mathcal{L}$ un langage, *N*, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage. Si $\mathcal{L}$ a un $\mathbb{N}$-modèle, alors l'ensemble des propositions closes de $\mathcal{L}$ démontrables dans la théorie vide est indécidable.

Démonstration. D'après 5.7, en prenant $\mathcal{T} = \varnothing$.

Théorème 5.1 (L'indécidabilité de l'arithmétique)

L'ensemble des propositions closes démontrables dans l'arithmétique, ou dans n'importe quelle extension de l'arithmétique qui a $(\mathbb{N}, 0, x \mapsto x + 1, +, \times, =)$ pour modèle, est indécidable.

Démonstration. On pose $N = \top$, $Null = (x = 0)$, $Succ = (y = S(x))$, $Plus = (z = x + y)$, $Mult = (z = x \times y)$ et $Eq = (x = y)$. Le modèle $\mathbb{N}$ est un $\mathbb{N}$-modèle. On peut donc appliquer la proposition 5.7.

On peut généraliser ce théorème à toutes les extensions cohérentes de l'arithmétique, c'est-à-dire à toutes les extensions de l'arithmétique qui ont un modèle, que ce modèle soit $\mathbb{N}$ ou non, ce qui permet de montrer également l'indécidabilité d'extensions exotiques de l'arithmétique qui, comme celles que l'on a construites dans la démonstration du théorème de Löwenheim-Skolem, sont cohérentes sans avoir $\mathbb{N}$ pour modèle. Mais on ne le fera pas ici. Il est cependant important de remarquer que ce résultat ne s'étend pas aux extensions contradictoires de l'arithmétique. Si on ajoute l'axiome $\bot$ par exemple, alors toutes les propositions sont démontrables et la théorie devient alors trivialement décidable.

Théorème 5.2 (Church)

Soit un langage contenant au moins un symbole de prédicat binaire, alors l'ensemble des propositions closes démontrables dans la théorie vide dans ce langage est indécidable.

Démonstration. On démontre que l'on peut définir des propositions N, *Null*, *Succ*, *Plus*, *Mult* et Eq et un $\mathbb{N}$-modèle $\mathcal{M}$ de ce langage et on conclut avec la proposition 5.8. Soit $\mathcal{M} = \mathbb{N} \uplus (\mathbb{N} \times \mathbb{N})$. On pose $\hat{R} = \{(a,(a,b)) \mid a \in \mathbb{N}, b \in \mathbb{N}\} \cup \{((a,b),b) \mid a \in \mathbb{N}, b \in \mathbb{N}\} \cup \{((a,b),(a+b,a\times b)) \mid a \in \mathbb{N}, b \in \mathbb{N}\}$. On définit les propositions

$$Eq = \forall z\ (x\ R\ z \Leftrightarrow y\ R\ z)$$

$$N = \exists y_1 \exists y_2 \exists y_3\ (\neg Eq[y_1,y_2] \wedge \neg Eq[y_1,y_3] \wedge \neg Eq[y_2,y_2] \wedge x\ R\ y_1 \wedge x\ R\ y_2 \wedge x\ R\ y_3)$$

$$Plus = N[x] \wedge N[y] \wedge N[z] \wedge \exists w \exists w'\ (x\ R\ w \wedge w\ R\ y \wedge w\ R\ w' \wedge z\ R\ w')$$

$$Mult = N[x] \wedge N[y] \wedge N[z] \wedge \exists w \exists w'\ (x\ R\ w \wedge w\ R\ y \wedge w\ R\ w' \wedge w'\ R\ z)$$

$$Null = N[x] \wedge \forall y\ (N[y] \Rightarrow Plus[x,y,y])$$

$$Un = N[x] \wedge \forall y\ (N[y] \Rightarrow Mult[x,y,y])$$

$$Succ = \exists u\ (Un[u] \wedge Plus[x,u,y])$$

Il n'est pas difficile de montrer que $[\![Eq[x,y]]\!]_{x=u,y=v} = 1$ si et seulement si $u = v$ en distinguant successivement le cas où u est un entier et celui ou c'est un couple, que $[\![N[x]]\!]_{x=u} = 1$ si et seulement si u est un entier, que $[\![Plus[x,y,z]]\!]_{x=p,y=q,z=r} = 1$ si et seulement si p, q et r sont des entiers et $p+q=r$, que $[\![Mult[x,y,z]]\!]_{x=p,y=q,z=r} = 1$ si et seulement si p, q et r sont des entiers et $p \times q = r$, que $[\![Null[x]]\!]_{x=p} = 1$ si et seulement si $p = 0$, et que $[\![Succ[x,y]]\!]_{x=p,y=q} = 1$ si et seulement si p et q sont des entiers et $p+1=q$.

Si le langage $\mathcal{L}$ contient un prédicat d'arité n pour $n \geq 2$, la construction d'un $\mathbb{N}$-modèle se généralise simplement. Un langage qui contient au moins un symbole de prédicat unaire P et un symbole de fonction f d'arité $n \geq 2$ est également indécidable, car avec un symbole de prédicat unaire P et un symbole de prédicat n-aire f on peut construire la proposition $P(f(x_1,\ldots,x_n))$ qui simule un symbole de prédicat n-aire.

Restent le cas où tous les symboles de prédicat sont d'arité nulle — auquel cas peu importent les symboles de fonction qui ne peuvent pas être utilisés dans les propositions — et celui où tous les symboles de fonction et de prédicat sont au plus unaire. On peut montrer que la démontrabilité est décidable dans ces deux cas.

Le théorème de Church permet d'apprécier la révolution qu'a constituée l'introduction par G. Frege de prédicats binaires. Toutes les logiques antérieures

— la logique des syllogismes d'Aristote, ... — qui ne comportent que des symboles au plus unaires — Homme, Mortel, ... — sont décidables.

Il faut prendre garde au fait que, bien que la démontrabilité dans la logique des prédicats dans un langage contenant au moins un symbole de prédicat binaire soit indécidable, en ajoutant des axiomes, on peut rendre la démontrabilité décidable. C'est naturellement le cas si on ajoute l'axiome $\bot$, puisqu'une théorie contradictoire est trivialement décidable, mais c'est par exemple aussi le cas si on ajoute l'axiome $\forall x \forall y\ (x\ R\ y)$ qui est cohérent. Le théorème de Church ne condamne donc pas *a priori* la recherche d'algorithmes pour des théories particulières, à condition que celles-ci n'aient pas de $\mathbb{N}$-modèle, même si elles utilisent un symbole de prédicat binaire. Ainsi, A. Tarski a démontré que la géométrie élémentaire est décidable, bien qu'elle utilise de nombreux prédicats binaires. Nous verrons, au chapitre 7, un autre exemple de théorie décidable.

Pour terminer cette section, mentionnons une autre généralisation du théorème d'indécidabilité de l'arithmétique. Nous avons vu que, quand nous nous donnons un programme f et des entiers $p_1, \ldots, p_n$, nous pouvons construire une proposition close de l'arithmétique qui est démontrable si et seulement si le programme f termine en $p_1, \ldots, p_n$. En 1970, Y. Matiyasevich a montré qu'il est possible de construire une telle proposition de la forme $\exists z_1 \ldots \exists z_m\ (t = u)$. De ce fait, l'ensemble des propositions de la forme $\exists z_1 \ldots \exists z_m\ (t = u)$ démontrables dans l'arithmétique est indécidable. Or, une proposition de cette forme exprime simplement l'existence d'une solution dans le domaine des entiers pour l'équation polynomiale à coefficients entiers $t = u$. De ce fait, il n'existe pas d'algorithme permettant de décider si une équation polynomiale à coefficients entiers a une solution dans le domaine des entiers ou non. Ce théorème résout, par la négative, un problème posé par D. Hilbert en 1900 (*le dixième problème de Hilbert*) : trouver un algorithme qui indique si une équation polynomiale à plusieurs variables et à coefficients entiers a une solution dans le domaine des entiers ou non. La construction d'une proposition de la forme $\exists z_1 \ldots \exists z_m\ (t = u)$ a cependant demandé un peu de travail, puisque si le lien entre le théorème de Church et une potentielle solution négative du dixième problème de Hilbert avait été entrevu en 1953 par M. Davis, qui avait proposé une première simplification de la forme des propositions représentant les programmes, ce n'est qu'en 1970 que la construction a été achevée par Matiyasevich.

5.4 La semi-décidabilité

Si l'ensemble des propositions démontrables dans la logique des prédicats est indécidable, on peut, en revanche, montrer que les règles de déduction sont effectives et donc en déduire, en utilisant la proposition 3.14, que l'ensemble des propositions démontrables est semi-décidable. Ce résultat s'étend à toutes les théories formées d'un nombre fini d'axiomes et même à toutes celles dont l'ensemble des axiomes est décidable.

Afin de préparer la démonstration de la proposition 5.10, nous redonnons ici la démonstration de cette proposition en partant de la proposition 3.13.

Proposition 5.9

Soit $\mathcal{T}$ une théorie dont l'ensemble des axiomes est décidable, les propositions démontrables dans $\mathcal{T}$ forment un ensemble semi-décidable.

Démonstration. Les règles de déduction étant effectives, d'après la proposition 3.13, l'ensemble des démonstrations est décidable. Et l'ensemble des axiomes de la théorie $\mathcal{T}$ étant décidable, on peut construire une fonction calculable g qui prend en argument le numéro d'un arbre π et le numéro d'une proposition A, vérifie que π est une démonstration bien formée, que sa racine est un séquent $\Gamma \vdash B$ tel que $B = A$ et tel que tous les éléments de Γ soient des axiomes de $\mathcal{T}$. La fonction h définie par $h(A)$ est le plus petit entier π tel que $1 \dot{-} g(\pi, A) = 0$ composée avec la fonction constante égale à 1 est un algorithme de semi-décision pour l'ensemble des propositions démontrables dans la théorie $\mathcal{T}$. Si A est démontrable dans $\mathcal{T}$, alors $h(A) = 1$, sinon h n'est pas définie en A.

L'ensemble des démonstrations est donc un ensemble décidable et celui des propositions démontrables un ensemble semi-décidable. Ces deux résultats sont à l'origine de la conception de deux types de programmes informatiques : les *programmes de vérification de démonstrations* et les *programmes de démonstration automatique*. Les programmes de la première catégorie prennent en entrée un arbre π, ils terminent toujours et indiquent si π est une démonstration bien formée ou non. Les programmes de la seconde catégorie, dont nous verrons un exemple au chapitre 6, prennent en entrée une proposition A et recherchent une démonstration π de cette proposition. Quand la proposition n'est pas démontrable, cette recherche se poursuit à l'infini.

5.5 Le premier théorème d'incomplétude de Gödel

La construction des fonctions g et h dans la démonstration de la proposition 5.9 mène à se demander ce qu'il se passe si on modifie la définition de la fonction h, en une fonction h', de manière à rechercher simultanément une démonstration de la proposition A et de la proposition $\neg A$ dans la théorie $\mathcal{T}$, et à retourner 1 ou 0 selon que l'on a trouvé une démonstration de l'une ou l'autre proposition. Chacune des propositions A et $\neg A$ pouvant être démontrable ou non, quatre cas peuvent se produire

1. les proposition A et $\neg A$ sont toutes les deux démontrables,
2. la proposition A est démontrable, mais pas $\neg A$,
3. la proposition $\neg A$ est démontrable, mais pas A,
4. ni la proposition A ni la proposition $\neg A$ ne sont démontrables.

Si on suppose la théorie $\mathcal{T}$ cohérente, aucune proposition n'est dans le cas (1.), dans le cas numéro (2.), $h'(A) = 1$, dans le cas numéro (3.), $h'(A) = 0$ et dans le cas numéro (4.), h' n'est pas définie en A.

Il est facile de donner des exemples de propositions qui sont dans le cas (2.) et de propositions qui sont dans le cas (3.), mais on peut s'interroger sur l'existence de propositions qui sont dans le cas (4.). Existe-t-il des propositions A telles que ni A ni $\neg A$ ne soient démontrables dans la théorie $\mathcal{T}$?

On montre par l'absurde que la réponse est positive dans toutes les théories dans lesquelles l'ensemble des propositions démontrables est indécidable : s'il n'existait pas de propositions dans le cas (4.), la fonction h' serait un algorithme de décision pour la démontrabilité dans la théorie $\mathcal{T}$, or, par hypothèse, un tel algorithme n'existe pas.

Définition 5.5 (Théorie complète)

Soit $\mathcal{L}$ un langage. Une théorie $\mathcal{T}$, exprimée dans $\mathcal{L}$, est dite *complète* si pour toute proposition close A de $\mathcal{L}$, A est démontrable dans $\mathcal{T}$ ou $\neg A$ est démontrable dans $\mathcal{T}$.

Proposition 5.10

Soit $\mathcal{L}$ un langage, N, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dont l'ensemble des axiomes est décidable et qui a un $\mathbb{N}$-modèle. Alors, la théorie $\mathcal{T}$ est incomplète : il existe une proposition close G telle que ni G ni $\neg G$ ne soient démontrables dans cette théorie.

Démonstration. Soit g la fonction calculable qui au numéro d'un arbre π et au numéro d'une proposition close A associe la valeur 1 si π est une démonstration bien formée dont la racine est un séquent $\Gamma \vdash B$ tel que $B = A$ et tous les éléments de Γ sont des axiomes de $\mathcal{T}$, et la valeur 0 sinon. Soit r la fonction calculable qui à une démonstration, dont la racine est un séquent $\Gamma \vdash B$, associe la proposition B. Soit $\hat{\neg}$ la fonction qui associe le numéro de la proposition $\neg A$ à celui de la proposition A : $\hat{\neg}(x) = \ulcorner\neg\urcorner; (x; 0)$. Soit $\mid$ la fonction *ou* sur les booléens : $x \mid y = x + y \dot{-} (x \times y)$ et $\chi_=$ la fonction caractéristique de l'égalité. Soit h_1 la fonction calculable définie par $h_1(A)$ est le plus petit entier π tel que $1 \dot{-} (g(\pi, A) \mid g(\pi, \hat{\neg}(A))) = 0$. Soit h' la fonction $h'(A) = \chi_=(r(h_1(A)), A)$.

Si la théorie $\mathcal{T}$ était complète, h' serait un algorithme de décision pour la démontrabilité dans $\mathcal{T}$ en contradiction avec la proposition 5.7.

Théorème 5.3 (Le premier théorème d'incomplétude de Gödel)

L'arithmétique et toutes ses extensions qui ont $(\mathbb{N}, 0, x \mapsto x+1, +, \times, =)$ comme modèle et dont l'ensemble des axiomes est décidable sont incomplètes.

Démonstration. On pose $N = \top$, $Null = (x = 0)$, $Succ = (y = S(x))$, $Plus = (z = x + y)$, $Mult = (z = x \times y)$ et $Eq = (x = y)$. L'ensemble des axiomes de la théorie est décidable et le modèle $\mathbb{N}$ est un $\mathbb{N}$-modèle de la théorie. On peut donc appliquer la proposition 5.10.

On peut généraliser ce théorème à toutes les extensions cohérente de l'arithmétique, c'est-à-dire à toutes les extensions de l'arithmétique dont l'ensemble des axiomes est décidable et qui ont un modèle, que ce modèle soit $\mathbb{N}$ ou non, ce qui permet de montrer également l'incomplétude d'extensions exotiques de l'arithmétique, qui sont cohérentes sans avoir $\mathbb{N}$ pour modèle. Mais on ne le fera pas ici.

Il est cependant important de remarquer que ce résultat ne s'étend pas aux extensions contradictoires de l'arithmétique. Si on ajoute l'axiome $\bot$, par exemple, alors toutes les propositions sont démontrables et la théorie est alors trivialement complète. Ce résultat ne s'étend pas non plus aux extensions de l'arithmétique dont l'ensemble d'axiomes est indécidable. Ainsi, la théorie dont les axiomes sont toutes les propositions valides dans le modèle $\mathbb{N}$ est un exemple d'extension de l'arithmétique cohérente et complète. Mais le théorème 5.3 montre que l'ensemble des axiomes de cette théorie est indécidable. De même, dans la démonstration de la proposition 2.5, on montre que toute théorie $\mathcal{T}$ a une extension cohérente et complète $\mathcal{U}$, mais, en général, l'ensemble des axiomes de cette théorie n'est pas décidable.

Exercice 5.1 (Un exemple de proposition indéterminée)

La proposition 5.10 montre qu'il existe une proposition close G, telle que ni G ni $\neg G$ ne soient démontrables dans la théorie $\mathcal{T}$, mais ne donne pas d'exemple d'une telle proposition. On montre dans cet exercice que l'on peut la modifier de manière à construire une telle proposition.

Soit $\mathcal{L}$ un langage, N, *Null*, *Succ*, *Plus*, *Mult* et *Eq* des propositions de ce langage et $\mathcal{T}$ une théorie dont l'ensemble des axiomes est décidable et qui a un $\mathbb{N}$-modèle $\mathcal{M}$. Soit $\mathcal{T}'$ la théorie $\mathcal{T} \cup \mathcal{T}_0$.

Soit f la fonction calculable telle que $f(n,p,q) = 1$ si $n = \ulcorner \pi \urcorner$, $p = \ulcorner A \urcorner$ et l'arbre π est une démonstration dans $\mathcal{T}'$ de la proposition $\forall w\ (N_q[w] \Rightarrow A)$ et $f(n,p,q) = 0$ sinon.

Soit F la proposition représentant un programme exprimant cette fonction. On écrit $F[t_1,t_2,t_3,u]$ la proposition $(t_1/x_1, t_2/x_2, t_3/x_3, u/y)F$.

D'après la proposition 5.3, les trois propositions suivantes sont équivalentes

- $f(n,p,q) = r$,
- la proposition

$$\forall x_1 \forall x_2 \forall x_3 \forall y\ (N_n[x_1] \wedge N_p[x_2] \wedge N_q[x_3] \wedge N_r[y] \Rightarrow F[x_1,x_2,x_3,y])$$

  est démontrable dans $\mathcal{T}'$,
- dans le modèle $\mathcal{M}$, $[\![F]\!]_{x_1=n,x_2=p,x_3=q,x_4=r} = 1$.

Soit T la proposition

$$\forall x \forall y\ ((N[x] \wedge N_1[y]) \Rightarrow \neg F[x,w,w,y])$$

$m = \ulcorner T \urcorner$ et G la proposition close

$$\forall w\ (N_m[w] \Rightarrow T)$$

Montrer que si G est démontrable dans $\mathcal{T}'$ alors

1. $[\![G]\!] = 1$,
2. pour tout entier n, $[\![F]\!]_{x_1=n,x_2=m,x_3=m,y=1} = 0$,
3. pour tout n, $f(n,m,m) = 0$,
4. la proposition $\forall w\ (N_m[w] \Rightarrow T)$ n'est pas démontrable dans $\mathcal{T}'$,
5. la proposition G n'est pas démontrable dans $\mathcal{T}'$.

En déduire que la proposition G n'est pas démontrable dans $\mathcal{T}'$.
En déduire que la proposition G n'est pas démontrable dans $\mathcal{T}$.
Montrer que si $\neg G$ est démontrable dans $\mathcal{T}'$ alors

1. $[\![\neg G]\!] = 1$,
2. il existe un entier n tel que $[\![F]\!]_{x_1=n,x_2=m,x_3=m,y=1} = 1$,
3. il existe un entier n tel que $f(n,m,m) = 1$,

4. la proposition $\forall w\ (N_m[w] \Rightarrow T)$ est démontrable dans $\mathcal{T}'$,
5. la proposition G est démontrable dans $\mathcal{T}'$,
6. la théorie $\mathcal{T}'$ est contradictoire.

En déduire que la proposition $\neg G$ n'est pas démontrable dans $\mathcal{T}'$.
En déduire que la proposition $\neg G$ n'est pas démontrable dans $\mathcal{T}$.

Exercice 5.2

Dans cet exercice on admettra le théorème de Matiyasevich, c'est-à-dire que l'ensemble des propositions démontrables dans l'arithmétique de la forme $\exists x_1 \ldots \exists x_m\ (t = u)$ est indécidable.

1. Montrer qu'il existe une proposition close A de la forme $\exists x_1 \ldots \exists x_m\ (t = u)$ telle que ni A ni $\neg A$ ne soient démontrables dans l'arithmétique.
2. Montrer que la proposition $\forall x_1 \ldots \forall x_m\ \neg(t = u)$ n'est pas démontrable.
3. Montrer que si a est un terme clos de l'arithmétique, alors il existe un entier n tel que la proposition $a = \underline{n}$ soit démontrable. Montrer que si n et p sont deux entiers, alors ou bien la proposition $\underline{n} = \underline{p}$ est démontrable ou bien la proposition $\neg(\underline{n} = \underline{p})$ est démontrable. Montrer que si a et b sont deux termes clos de l'arithmétique, alors la proposition $a = b$ est démontrable ou la proposition $\neg(a = b)$ est démontrable.
4. Soit une équation $t = u$ dont les variables sont parmi $x_1, \ldots, x_m$ et $p_1, \ldots, p_m$ des entiers tels que la proposition $(\underline{p_1}/x_1, \ldots, \underline{p_m}/x_m)(t = u)$ soit démontrable. Montrer que la proposition $\exists x_1 \ldots \exists x_m\ (t = u)$ est démontrable. Montrer que si la proposition $\exists x_1 \ldots \exists x_m\ (t = u)$ n'est pas démontrable, alors pour tout $p_1, \ldots, p_m$, la proposition $(\underline{p_1}/x_1, \ldots, \underline{p_m}/x_m)(t = u)$ n'est pas démontrable.
5. Montrer que si la proposition $\exists x_1 \ldots \exists x_m\ (t = u)$ n'est pas démontrable, alors pour tout $p_1, \ldots, p_m$, la proposition $(\underline{p_1}/x_1, \ldots, \underline{p_m}/x_m)\neg(t = u)$ est démontrable.
6. Montrer qu'il existe une proposition A de la forme $\neg(t = u)$ dont les variables sont parmi $x_1, \ldots, x_m$ et telle que
   - pour tous $p_1, \ldots, p_m$ la proposition $(\underline{p_1}/x_1, \ldots, \underline{p_m}/x_m)A$ est démontrable,
   - la proposition $\forall x_1 \ldots \forall x_m\ A$ n'est pas démontrable.

6
La démonstration automatique

Nous avons vu, au chapitre 5, que la démontrabilité en logique des prédicats est indécidable, mais semi-décidable, c'est-à-dire qu'il existe une fonction calculable f telle que $f(\ulcorner \Gamma \vdash \Delta \urcorner) = 1$ si le séquent $\Gamma \vdash \Delta$ est démontrable et f n'est pas définie en $\ulcorner \Gamma \vdash \Delta \urcorner$ sinon. Cette fonction énumère les entiers et teste si l'un d'eux est le numéro d'une démonstration de $\Gamma \vdash \Delta$. Si une telle démonstration existe, son numéro apparaîtra au cours de l'énumération. Sinon, celle-ci se poursuivra à l'infini.

Cette méthode permet de montrer que la démontrabilité en logique des prédicats est semi-décidable, mais elle est sans intérêt pratique. Toutefois, l'idée d'énumération et de test sur laquelle elle repose peut aussi mener à des méthodes moins inefficaces.

6.1 Le calcul des séquents

6.1.1 La recherche de démonstrations en déduction naturelle

Une méthode de recherche de démonstrations consiste à énumérer les règles qui peuvent s'appliquer à chaque nœud d'une démonstration, en procédant du bas vers le haut. Si on cherche, par exemple, une démonstration du séquent $P \vdash Q \Rightarrow (P \wedge Q)$, on commence par énumérer les différentes règles qui peuvent

être utilisées comme dernière règle d'une telle démonstration. Parmi d'autres possibilités, cette dernière règle peut être $\Rightarrow$-intro

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta} \Rightarrow\text{-intro}$$

et, dans ce cas, la seule possibilité est d'avoir $\Gamma = [P], A = Q, B = P \wedge Q$ et $\Delta = [\,]$. La prémisse de cette règle est donc le séquent $P, Q \vdash P \wedge Q$ et on énumère les différentes règles qui peuvent être utilisées comme dernière règle d'une démonstration de ce séquent. Parmi d'autres possibilités, cette dernière règle peut être $\wedge$-intro

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \wedge\text{-intro}$$

et, dans ce cas, la seule possibilité est d'avoir $\Gamma = [P, Q], A = P, B = Q$ et $\Delta = [\,]$. Les prémisses de cette règle sont donc les séquents $P, Q \vdash P$ et $P, Q \vdash Q$. On cherche d'abord une démonstration du premier de ces séquents et on énumère les différentes règles qui peuvent être utilisées comme dernière règle d'un telle démonstration. Parmi d'autres possibilités, cette dernière règle peut être la règle *axiome*, qui permet de conclure. Cette même règle permet de démontrer également le second séquent et on aboutit à la démonstration

$$\dfrac{\dfrac{\dfrac{}{P, Q \vdash P}\,\text{axiome} \quad \dfrac{}{P, Q \vdash Q}\,\text{axiome}}{P, Q \vdash P \wedge Q}\,\wedge\text{-intro}}{P \vdash Q \Rightarrow (P \wedge Q)}\,\Rightarrow\text{-intro}$$

Quand on énumère les règles permettant de démontrer le séquent $P \vdash Q \Rightarrow (P \wedge Q)$, la seule règle d'introduction possible est la règle $\Rightarrow$-intro. En effet, la règle $\vee$-intro, par exemple, ne permet de démontrer que des propositions de la forme $A \vee B$ et elle ne peut pas être utilisée pour démontrer une implication. En revanche, pour démontrer ce séquent, on peut utiliser toutes les règles d'élimination, par exemple, la règle $\wedge$-élim

$$\frac{\Gamma \vdash A \wedge B, \Delta}{\Gamma \vdash A, \Delta} \wedge\text{-élim}$$

De plus, quand on utilise cette règle, le séquent à démontrer $P \vdash Q \Rightarrow (P \wedge Q)$, nous suggère de prendre $\Gamma = [P]$, $A = Q \Rightarrow (P \wedge Q)$ et $\Delta = [\,]$, mais il ne suggère rien pour B qui n'apparaît pas dans la conclusion de la règle. Il faut donc choisir une proposition B et si la proposition choisie n'est pas la bonne, il faut essayer d'autres possibilités. Autrement dit, il est nécessaire d'énumérer toutes les propositions B possibles.

Ainsi, quand on cherche à démontrer le séquent $P \wedge Q \vdash P$, rien n'indique qu'il faut utiliser la règle $\wedge$-élim ni qu'il faut choisir $B = Q$ pour obtenir la démonstration

$$\dfrac{\dfrac{}{P \wedge Q \vdash P \wedge Q}\,\text{axiome}}{P \wedge Q \vdash P}\,\wedge\text{-élim}$$

6.1.2 Les règles du calcul des séquents

La déduction naturelle permet d'exploiter la forme de la conclusion du séquent pour guider le choix des règles d'introduction. En revanche, elle ne permet pas d'utiliser celle des hypothèses pour guider le choix des règles d'élimination. L'idée du *calcul des séquents* est de conserver les règles d'introduction de la déduction naturelle, désormais appelées *règles droites*, et de remplacer les règles d'élimination par des règles d'introduction sur les hypothèses du séquent : les *règles gauches*. Par exemple, la règle $\wedge$-élim est remplacée par la règle

$$\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\,\wedge\text{-gauche}$$

Ainsi, le séquent $P \wedge Q \vdash P$ a une démonstration tout à fait différente de sa démonstration en déduction naturelle

$$\dfrac{\dfrac{}{P, Q \vdash P}\,\text{axiome}}{P \wedge Q \vdash P}\,\wedge\text{-gauche}$$

et lors de la recherche d'une démonstration de ce séquent, la forme de l'hypothèse $P \wedge Q$ permet de guider le choix de la règle gauche. Chaque règle d'élimination de la déduction naturelle peut, de même, être remplacée par une règle gauche

$$\dfrac{}{\Gamma, \bot \vdash \Delta}\,\bot\text{-gauche}$$

$$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\,\vee\text{-gauche}$$

$$\dfrac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}\,\Rightarrow\text{-gauche}$$

$$\dfrac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\,\neg\text{-gauche}$$

$$\dfrac{\Gamma, (t/x)A \vdash \Delta}{\Gamma, \forall x\ A \vdash \Delta}\,\forall\text{-gauche}$$

$$\dfrac{\Gamma, A \vdash \Delta}{\Gamma, \exists x\ A \vdash \Delta}\,\exists\text{-gauche } x \text{ non libre dans } \Gamma, \Delta$$

Nous devons également ajouter une règle de contraction à gauche pour pouvoir utiliser les hypothèses plusieurs fois.

Comme en déduction naturelle, le tiers exclu peut être exprimé par une règle particulière ou, comme nous le faisons ici, par le fait d'utiliser des séquents à plusieurs conclusions.

Enfin, pour montrer l'équivalence du calcul des séquents et de la déduction naturelle nous aurons besoin d'une règle supplémentaire : la règle de *coupure*.

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\ \text{coupure}$$

dont nous démontrerons ensuite qu'elle est superflue.

Cela mène à la définition suivante.

Définition 6.1 (Les règles du calcul des séquents)

$$\frac{}{\Gamma, A \vdash A, \Delta}\ \text{axiome}$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\ \text{coupure}$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{contraction-gauche}$$

$$\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \text{contraction-droite}$$

$$\frac{}{\Gamma \vdash \top, \Delta}\ \top\text{-droite}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot\text{-gauche}$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge\text{-gauche}$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \wedge\text{-droite}$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee\text{-gauche}$$

$$\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee\text{-droite}$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}\ \Rightarrow\text{-gauche}$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta}\ \Rightarrow\text{-droite}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ \neg\text{-gauche}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg\text{-droite}$$

$$\frac{\Gamma, (t/x)A \vdash \Delta}{\Gamma, \forall x\ A \vdash \Delta}\ \forall\text{-gauche}$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x\ A, \Delta}\ \forall\text{-droite } x \text{ non libre dans } \Gamma, \Delta$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists x\ A \vdash \Delta}\ \exists\text{-gauche } x \text{ non libre dans } \Gamma, \Delta$$

$$\frac{\Gamma \vdash (t/x)A, \Delta}{\Gamma \vdash \exists x\ A, \Delta}\ \exists\text{-droite}$$

En déduction naturelle, un séquent a toujours une conclusion unique. Dans le système D', un séquent peut avoir une ou plusieurs conclusions, mais il ne peut pas en avoir aucune. En effet, les règles du système D', comme celles de la déduction naturelle, transforment la conclusion des séquents : il faut qu'il y ait quelque chose à transformer. En calcul des séquents, les hypothèses et les conclusions d'un séquent jouent des rôles symétriques et, de même qu'un séquent peut n'avoir aucune hypothèse, il peut n'avoir aucune conclusion. Intuitivement, le séquent $\Gamma \vdash$ est une variante du séquent $\Gamma \vdash \bot$. En effet, de manière générale, dans le système D' comme en calcul des séquents, le séquent $\Gamma \vdash \Delta$ est démontrable si et seulement si le séquent $\Gamma \vdash \bot, \Delta$ est démontrable.

Cela explique la différence entre la règle $\neg$-droite du calcul des séquents

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg\text{-droite}$$

dont la conclusion de la prémisse peut être vide, quand Δ est vide, et la règle homologue du système D'

$$\frac{\Gamma, A \vdash \bot, \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg\text{-intro}$$

6.1.3 L'équivalence avec la déduction naturelle

Nous voulons, à présent, montrer qu'un séquent $\Gamma \vdash A$ est démontrable en déduction naturelle si et seulement s'il est démontrable en calcul des séquents. Comme nous avons choisi une formulation du calcul des séquents avec des séquents à plusieurs conclusions, nous montrons l'équivalence avec la formulation homologue de la déduction naturelle, c'est-à-dire le système D'.

Nous commençons par montrer la proposition suivante qui est l'analogue, pour le calcul des séquents, des propositions 1.6 et 1.13.

Proposition 6.1 (L'affaiblissement)

Si le séquent $\Gamma \vdash \Delta$ est démontrable en calcul des séquents, alors c'est également le cas des séquents $\Gamma, A \vdash \Delta$ et $\Gamma \vdash A, \Delta$.

Démonstration. Par récurrence sur la structure de la démonstration de $\Gamma \vdash \Delta$.

Proposition 6.2

Si le séquent $\Gamma \vdash A$ est démontrable dans le système D', alors il est démontrable en calcul des séquents.

Démonstration. On montre, plus généralement, que si le séquent $\Gamma \vdash \Delta$ est démontrable dans le système D', alors il est démontrable en calcul des séquents. Par récurrence sur la structure de la démonstration de ce séquent dans le système D'.

- Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \bot, \Delta'}}{\Gamma \vdash A, \Delta'}\ \bot\text{-élim}$$

  alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe une démonstration en calcul des séquents π' du séquents $\Gamma \vdash \bot, A, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'}{\Gamma \vdash \bot, A, \Delta'} \quad \dfrac{}{\Gamma, \bot \vdash A, \Delta'}\ \bot\text{-gauche}}{\Gamma \vdash A, \Delta'}\ \text{coupure}$$

- Si la démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A \wedge B, \Delta'}}{\Gamma \vdash A, \Delta'}\ \wedge\text{-élim}$$

  alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe une démonstrations en calcul des séquents π' du séquent $\Gamma \vdash A \wedge B, A, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'}{\Gamma \vdash A \wedge B, A, \Delta'} \quad \dfrac{\dfrac{}{\Gamma, A, B \vdash A, \Delta'}\ \text{axiome}}{\Gamma, A \wedge B \vdash A, \Delta'}\ \wedge\text{-gauche}}{\Gamma \vdash A, \Delta'}\ \text{coupure}$$

  On procède de même pour l'autre règle $\wedge$-élim.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A \vee B, \Delta'} \quad \dfrac{\pi_2}{\Gamma, A \vdash C, \Delta'} \quad \dfrac{\pi_3}{\Gamma, B \vdash C, \Delta'}}{\Gamma \vdash C, \Delta'} \vee\text{-élim}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe des démonstrations en calcul des séquents π'_1, π'_2 et π'_3 de $\Gamma \vdash A \vee B, C, \Delta'$, de $\Gamma, A \vdash C, \Delta'$ et de $\Gamma, B \vdash C, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'_1}{\Gamma \vdash A \vee B, C, \Delta'} \quad \dfrac{\dfrac{\pi'_2}{\Gamma, A \vdash C, \Delta'} \quad \dfrac{\pi'_3}{\Gamma, B \vdash C, \Delta'}}{\Gamma, A \vee B \vdash C, \Delta'} \vee\text{-gauche}}{\Gamma \vdash C, \Delta'} \text{coupure}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A \Rightarrow B, \Delta'} \quad \dfrac{\pi_2}{\Gamma \vdash A, \Delta'}}{\Gamma \vdash B, \Delta'} \Rightarrow\text{-élim}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe des démonstrations en calcul des séquents π'_1 et π'_2 des séquents $\Gamma \vdash A \Rightarrow B, B, \Delta'$ et $\Gamma \vdash A, B, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'_1}{\Gamma \vdash A \Rightarrow B, B, \Delta'} \quad \dfrac{\dfrac{\pi'_2}{\Gamma \vdash A, B, \Delta'} \quad \dfrac{}{\Gamma, B \vdash B, \Delta'}\text{axiome}}{\Gamma, A \Rightarrow B \vdash B, \Delta'} \Rightarrow\text{-gauche}}{\Gamma \vdash B, \Delta'} \text{coupure}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash \neg A, \Delta'} \quad \dfrac{\pi_2}{\Gamma \vdash A, \Delta'}}{\Gamma \vdash \bot, \Delta'} \neg\text{-élim}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe des démonstrations en calcul des séquents π'_1 et π'_2 des séquents $\Gamma \vdash \neg A, \bot, \Delta'$ et $\Gamma \vdash A, \bot, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'_1}{\Gamma \vdash \neg A, \bot, \Delta'} \quad \dfrac{\dfrac{\pi'_2}{\Gamma \vdash A, \bot, \Delta'}}{\Gamma, \neg A \vdash \bot, \Delta'} \neg\text{-gauche}}{\Gamma \vdash \bot, \Delta'} \text{coupure}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \forall x\ A, \Delta'}}{\Gamma \vdash (t/x)A, \Delta'}\ \forall\text{-élim}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe une démonstration en calcul des séquents π' du séquent $\Gamma \vdash \forall x\ A, (t/x)A, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'}{\Gamma \vdash \forall x\ A, (t/x)A, \Delta'} \quad \dfrac{\overline{\Gamma, (t/x)A \vdash (t/x)A, \Delta'}}{\Gamma, \forall x\ A \vdash (t/x)A, \Delta'}\ \forall\text{-gauche}}{\Gamma \vdash (t/x)A, \Delta'}\ \text{coupure}\quad \text{axiome}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash \exists x\ A, \Delta'} \quad \dfrac{\pi_2}{\Gamma, A \vdash B, \Delta'}}{\Gamma \vdash B, \Delta'}\ \exists\text{-élim}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe des démonstrations en calcul des séquents π'_1 et π'_2 des séquents $\Gamma \vdash \exists x\ A, B, \Delta'$ et $\Gamma, A \vdash B, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'_1}{\Gamma \vdash \exists x\ A, B, \Delta'} \quad \dfrac{\dfrac{\pi'_2}{\Gamma, A \vdash B, \Delta'}}{\Gamma, \exists x\ A \vdash B}\ \exists\text{-gauche}}{\Gamma \vdash B, \Delta'}\ \text{coupure}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma, A \vdash \bot, \Delta'}}{\Gamma \vdash \neg A, \Delta'}\ \neg\text{-intro}$$

alors, par hypothèse de récurrence, il existe une démonstrations en calcul des séquents π' du séquent $\Gamma, A \vdash \bot, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\dfrac{\pi'}{\Gamma, A \vdash \bot, \Delta'} \quad \overline{\Gamma, A, \bot \vdash \Delta'}\ \bot\text{-gauche}}{\Gamma, A \vdash \Delta'}\ \text{coupure}}{\Gamma \vdash \neg A, \Delta'}\ \neg\text{-droite}$$

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A, \Delta'}}{\Gamma \vdash A \vee B, \Delta'} \vee\text{-intro}$$

alors, par hypothèse de récurrence et d'après la proposition 6.1, il existe une démonstration en calcul des séquents π' du séquent $\Gamma \vdash A, B, \Delta'$. On construit la démonstration

$$\dfrac{\dfrac{\pi'}{\Gamma \vdash A, B, \Delta'}}{\Gamma \vdash A \vee B, \Delta'} \vee\text{-droite}$$

On procède de même pour l'autre règle $\vee$-intro.

– Les autres règles du système D' sont aussi des règles du calcul des séquents, leur cas est donc trivial.

Pour montrer que, réciproquement, si le séquent $\Gamma \vdash A$ est démontrable en calcul des séquents, il est démontrable dans le système D', nous voudrions montrer que si le séquent $\Gamma \vdash \Delta$ est démontrable en calcul des séquents, alors il est démontrable dans le système D'. Malheureusement, cela n'est pas vrai dans le cas où Δ est vide. Nous montrons donc une proposition plus faible : si le séquent $\Gamma \vdash \Delta$ est démontrable en calcul des séquents, alors le séquent $\Gamma \vdash \bot, \Delta$ est démontrable dans le système D', puis nous éliminons la proposition $\bot$, dans le cas où le multiensemble Δ est un singleton.

Nous commençons par montrer la proposition suivante.

Proposition 6.3

Si les séquents $\Gamma, A \vdash \Delta$ et $\Gamma \vdash A, \Delta$ sont démontrables dans le système D', alors le séquent $\Gamma \vdash \Delta$ aussi.

Démonstration. On montre, plus généralement, que si les séquents $\Gamma, \Sigma, A \vdash \Delta$ et $\Gamma \vdash A, \Delta$ sont démontrables dans le système D', alors le séquent $\Gamma, \Sigma \vdash \Delta$ aussi. Par récurrence sur la structure de la démonstration de $\Gamma, \Sigma, A \vdash \Delta$. Tous les cas sont triviaux, sauf celui de la règle *axiome*. Dans ce cas, si la proposition commune à Γ, Σ, A et Δ est un élément de Γ, Σ, le séquent $\Gamma, \Sigma \vdash \Delta$ est démontrable avec la règle *axiome*. Si c'est A, alors, d'après la proposition 1.13, le séquent $\Gamma, \Sigma \vdash A, \Delta$ est démontrable et la proposition A étant un élément de Δ, le séquent $\Gamma, \Sigma \vdash \Delta$ est démontrable avec la règle *contraction*.

Proposition 6.4

Si le séquent $\Gamma \vdash A$ est démontrable en calcul des séquents, alors il est démontrable dans le système D'.

Démonstration. On montre que si le séquent $\Gamma \vdash \Delta$ est démontrable en calcul des séquents, alors le séquent $\Gamma \vdash \bot, \Delta$ est démontrable dans le système D'. Le résultat en découle car si le séquent $\Gamma \vdash \bot, A$ a une démonstration π dans le système D', alors le séquent $\Gamma \vdash A$ a la démonstration

$$\dfrac{\dfrac{\dfrac{\pi}{\Gamma \vdash \bot, A}}{\Gamma \vdash A, A}\,\bot\text{-élim}}{\Gamma \vdash A}\,\text{contraction}$$

La démonstration procède par récurrence sur la structure de la démonstration du séquent $\Gamma \vdash \Delta$ dans le calcul des séquents.

- Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma \vdash A, \Delta} \quad \dfrac{\pi_2}{\Gamma, A \vdash \Delta}}{\Gamma \vdash \Delta}\,\text{coupure}$$

  alors, par hypothèse de récurrence, les séquents $\Gamma \vdash \bot, A, \Delta$ et $\Gamma, A \vdash \bot, \Delta$ sont démontrables dans le système D'. Le séquent $\Gamma \vdash \bot, \Delta$ est donc démontrable d'après la proposition 6.3.
- Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma', A, A \vdash \Delta}}{\Gamma', A \vdash \Delta}\,\text{contraction-gauche}$$

  alors, par hypothèse de récurrence, le séquent $\Gamma', A, A \vdash \bot, \Delta$ a une démonstration dans le système D'. On montre par récurrence sur la structure de cette démonstration que le séquent $\Gamma', A \vdash \bot, \Delta$ a une démonstration dans le système D'.
- Si cette démonstration a la forme

$$\dfrac{}{\Gamma', \bot \vdash \Delta}\,\bot\text{-gauche}$$

  alors, le séquent $\Gamma', \bot \vdash \bot, \Delta$ est démontrable dans le système D' avec la règle *axiome*.
- Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma', A, B \vdash \Delta}}{\Gamma', A \wedge B \vdash \Delta}\,\wedge\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, le séquent $\Gamma', A \wedge B, A, B \vdash \bot, \Delta$ est démontrable dans le système D'. Les séquents $\Gamma', A \wedge B, B \vdash A, \bot, \Delta$ et $\Gamma', A \wedge B \vdash B, \bot, \Delta$ sont démontrables avec les règles *axiome* et $\wedge$-élim. Le séquent $\Gamma', A \wedge B \vdash \bot, \Delta$ est donc démontrable d'après la proposition 6.3.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma', A \vdash \Delta} \quad \dfrac{\pi_2}{\Gamma', B \vdash \Delta}}{\Gamma', A \vee B \vdash \Delta} \vee\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, les séquents $\Gamma', A \vee B, A \vdash \bot, \Delta$ et $\Gamma', A \vee B, B \vdash \bot, \Delta$ sont démontrables dans le système D'. Le séquent $\Gamma', A \vee B \vdash \bot, \Delta$ est donc démontrable avec les règles *axiome* et $\vee$-élim.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi_1}{\Gamma' \vdash A, \Delta} \quad \dfrac{\pi_2}{\Gamma', B \vdash \Delta}}{\Gamma', A \Rightarrow B \vdash \Delta} \Rightarrow\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, les séquents $\Gamma', A \Rightarrow B \vdash \bot, A, B, \Delta$ et $\Gamma', A \Rightarrow B, B \vdash \bot, \Delta$ sont démontrables dans le système D'. Le séquent $\Gamma', A, A \Rightarrow B \vdash B, \bot, \Delta$ est démontrable avec les règles *axiome* et $\Rightarrow$-élim. Le séquent $\Gamma', A \Rightarrow B \vdash \bot, \Delta$ est donc démontrable d'après la proposition 6.3.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma' \vdash A, \Delta}}{\Gamma', \neg A \vdash \Delta} \neg\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, le séquent $\Gamma', \neg A \vdash \bot, A, \Delta$ est démontrable dans le système D'. Le séquent $\Gamma', \neg A, A \vdash \bot, \Delta$ est démontrable avec les règles *axiome* et $\neg$-élim. Le séquent $\Gamma', \neg A \vdash \bot, \Delta$ est donc démontrable d'après la proposition 6.3.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma', (t/x)A \vdash \Delta}}{\Gamma', \forall x\ A \vdash \Delta} \forall\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, le séquent $\Gamma', \forall x\ A, (t/x)A \vdash \bot, \Delta$ est démontrable dans le système D'. Le séquent $\Gamma', \forall x\ A \vdash (t/x)A, \bot, \Delta$ est démontrable avec les règles *axiome* et $\forall$-élim. Le séquent $\Gamma', \forall x\ A \vdash \bot, \Delta$ est donc démontrable d'après la proposition 6.3.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma', A \vdash \Delta}}{\Gamma', \exists x\ A \vdash \Delta}\ \exists\text{-gauche}$$

alors, par hypothèse de récurrence et d'après la proposition 1.13, le séquent $\Gamma', \exists x\ A, A \vdash \bot, \Delta$ est démontrable dans le système D'. Le séquent $\Gamma', \exists x\ A \vdash \bot, \Delta$ est donc démontrable avec les règles *axiome* et $\exists$-élim.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma, A \vdash \Delta'}}{\Gamma \vdash \neg A, \Delta'}\ \neg\text{-droite}$$

alors, par hypothèse de récurrence, le séquent $\Gamma, A \vdash \bot, \Delta'$ est démontrable dans le système D'. Le séquent $\Gamma \vdash \neg A, \Delta'$ est donc démontrable avec la règle $\neg$-intro et le séquent $\Gamma \vdash \bot, \neg A, \Delta'$ d'après la proposition 1.13.

– Si cette démonstration a la forme

$$\dfrac{\dfrac{\pi}{\Gamma \vdash A, B, \Delta'}}{\Gamma \vdash A \vee B, \Delta'}\ \vee\text{-droite}$$

alors, par hypothèse de récurrence, le séquent $\Gamma \vdash \bot, A, B, \Delta'$ est démontrable dans le système D'. Le séquent $\Gamma \vdash \bot, A \vee B, \Delta'$ est donc démontrable avec les règles *contraction* et $\vee$-intro.

– Les autres règles du calcul des séquents sont aussi des règles du système D', leur cas est donc trivial.

Théorème 6.1

Le séquent $\Gamma \vdash A$ est démontrable en calcul des séquents si et seulement s'il est démontrable dans le système D' si et seulement s'il est démontrable en déduction naturelle.

Démonstration. D'après les propositions 6.2, 6.4 et 1.12.

6.1.4 L'élimination des coupures

Toutes les propositions apparaissant dans les prémisses d'une règle gauche ou d'une règle droite du calcul des séquents apparaissent également dans la

conclusion de cette règle. De ce fait, l'application d'une telle règle, lors de la recherche d'une démonstration, ne demande pas de choisir une proposition. Cependant, la règle de coupure

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\ \text{coupure}$$

que nous avons utilisée pour démontrer l'équivalence du calcul des séquents avec le système D', ne vérifie pas cette propriété, puisque la proposition A apparaît dans les prémisses de cette règle, mais pas dans sa conclusion. L'application de cette règle demande donc de choisir une proposition A.

Toutefois, comme nous allons le voir, cette règle est redondante et peut être supprimée du calcul des séquents.

Définition 6.2 (Le calcul des séquents sans coupures)

Le *calcul des séquents sans coupures* est formé des règles de la définition 6.1 sauf la règle *coupure*.

De manière évidente si un séquent $\Gamma \vdash \Delta$ est démontrable dans le calcul des séquents sans coupures il est démontrable dans le calcul des séquents. Nous voulons montrer que, réciproquement, si un séquent $\Gamma \vdash \Delta$ est démontrable dans le calcul des séquents, alors il est démontrable dans le calcul des séquents sans coupures. Pour cela, il suffit de montrer la proposition suivante.

Proposition 6.5

Si les séquents $\Gamma, A \vdash \Delta$ et $\Gamma \vdash A, \Delta$ sont démontrables dans le calcul des séquents sans coupures, alors le séquent $\Gamma \vdash \Delta$ est démontrable dans le calcul des séquents sans coupures.

Démonstration. On montre plus généralement que si les séquents $\Gamma, A^n \vdash \Delta$ et $\Gamma' \vdash A^m, \Delta'$ ont des démonstrations π et π' dans le calcul des séquents sans coupures, alors le séquent $\Gamma, \Gamma' \vdash \Delta, \Delta'$ a une démonstration dans le calcul des séquents sans coupures. La proposition découle du cas $n = m = 1$, en utilisant les règles de contraction.

La démonstration procède par une récurrence double d'abord sur le nombre de connecteurs et quantificateurs de la proposition A, puis sur la somme des tailles des démonstrations π et π'.

On considère les dernières règles des démonstrations π et π'. Dans une première série de cas, ces deux règles sont appliquées à la proposition A, ce qui implique $n \geq 1$ et $m \geq 1$.

- Si la dernière règle de π est la règle *axiome*, alors le multiensemble Δ contient la proposition A. Le séquent $\Gamma' \vdash A^m, \Delta'$ a une démonstration et, d'après la proposition 6.1, le séquent $\Gamma, \Gamma' \vdash A^m, \Delta, \Delta'$ également. À partir de cette démonstration, on construit une démonstration du séquent $\Gamma, \Gamma' \vdash \Delta, \Delta'$ avec la règle *contraction-droite*. On procède de même si la dernière règle de π' est la règle *axiome*.
- Si la dernière règle de π ou celle de π' est une règle de contraction, on applique l'hypothèse de récurrence.
- Dans les autres cas, si la dernière règle de π est la règle $\wedge$-gauche, alors $A = (B \wedge C)$ et la dernière règle de π' est la règle $\wedge$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho}{\Gamma, A^{n-1}, B, C \vdash \Delta}}{\Gamma, A^{n-1}, B \wedge C \vdash \Delta} \wedge\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'_1}{\Gamma' \vdash B, A^{m-1}, \Delta'} \quad \dfrac{\rho'_2}{\Gamma' \vdash C, A^{m-1}, \Delta'}}{\Gamma' \vdash B \wedge C, A^{m-1}, \Delta'} \wedge\text{-droite}$$

L'hypothèse de récurrence, appliquée à π et ρ'_1, puis à π et ρ'_2 et enfin à ρ et π' donne une démonstration de $\Gamma, \Gamma' \vdash B, \Delta, \Delta'$, de $\Gamma, \Gamma' \vdash C, \Delta, \Delta'$ et de $\Gamma, \Gamma', B, C \vdash \Delta, \Delta'$. L'hypothèse de récurrence appliquée à B et C et les règles de contraction donnent une démonstration de $\Gamma, \Gamma', C \vdash \Delta, \Delta'$ puis de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

- Si la dernière règle de π est la règle $\vee$-gauche, alors $A = (B \vee C)$ et la dernière règle de π' est la règle $\vee$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho_1}{\Gamma, A^{n-1}, B \vdash \Delta} \quad \dfrac{\rho_2}{\Gamma, A^{n-1}, C \vdash \Delta}}{\Gamma, A^{n-1}, B \vee C \vdash \Delta} \vee\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'}{\Gamma' \vdash B, C, A^{m-1}, \Delta'}}{\Gamma' \vdash B \vee C, A^{m-1}, \Delta'} \vee\text{-droite}$$

L'hypothèse de récurrence, appliquée à π et ρ', puis à ρ_1 et π' et enfin à ρ_2 et π' donne une démonstration de $\Gamma, \Gamma' \vdash B, C, \Delta, \Delta'$, de $\Gamma, \Gamma', B \vdash \Delta, \Delta'$ et de $\Gamma, \Gamma', C \vdash \Delta, \Delta'$. L'hypothèse de récurrence appliquée à B et C et les règles de contraction donnent une démonstration de $\Gamma, \Gamma' \vdash C, \Delta, \Delta'$ puis de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

– Si la dernière règle de π est la règle $\Rightarrow$-gauche, alors $A = (B \Rightarrow C)$ et la dernière règle de π' est la règle $\Rightarrow$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho_1}{\Gamma, A^{n-1} \vdash B, \Delta} \quad \dfrac{\rho_2}{\Gamma, A^{n-1}, C \vdash \Delta}}{\Gamma, A^{n-1}, B \Rightarrow C \vdash \Delta} \Rightarrow\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'}{\Gamma', B \vdash C, A^{m-1}, \Delta'}}{\Gamma' \vdash B \Rightarrow C, A^{m-1}, \Delta'} \Rightarrow\text{-droite}$$

L'hypothèse de récurrence, appliquée à π et ρ', puis à ρ_1 et π' et enfin à ρ_2 et π' donne une démonstration de $\Gamma, \Gamma', B \vdash C, \Delta, \Delta'$, de $\Gamma, \Gamma' \vdash B, \Delta, \Delta'$ et de $\Gamma, \Gamma', C \vdash \Delta, \Delta'$. L'hypothèse de récurrence appliquée à B et C et les règles de contraction donnent une démonstration de $\Gamma, \Gamma' \vdash C, \Delta, \Delta'$ puis de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

– Si la dernière règle de π est la règle $\neg$-gauche, alors $A = \neg B$ et la dernière règle de π' est la règle $\neg$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho}{\Gamma, A^{n-1} \vdash B, \Delta}}{\Gamma, A^{n-1}, \neg B \vdash \Delta} \neg\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'}{\Gamma', B \vdash A^{m-1}, \Delta'}}{\Gamma' \vdash \neg B, A^{m-1}, \Delta'} \neg\text{-droite}$$

L'hypothèse de récurrence, appliquée à π et ρ', puis à ρ et π' donne une démonstration de $\Gamma, \Gamma', B \vdash \Delta, \Delta'$ et de $\Gamma, \Gamma' \vdash B, \Delta, \Delta'$. L'hypothèse de récurrence appliquée à B et les règles de contraction donnent une démonstration de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

– Si la dernière règle de π est la règle $\forall$-gauche, alors $A = \forall x\ B$ et la dernière règle de π' est la règle $\forall$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho}{\Gamma, A^{n-1}, (t/x)B \vdash \Delta}}{\Gamma, A^{n-1}, \forall x\ B \vdash \Delta} \forall\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'}{\Gamma' \vdash B, A^{m-1}, \Delta'}}{\Gamma' \vdash \forall x\ B, A^{m-1}, \Delta'} \forall\text{-droite}$$

Comme x n'est pas une variable libre de Γ', de A ou de Δ', en substituant la variable x par le terme t dans la démonstration ρ', on obtient une

démonstration ρ'_1 de $\Gamma' \vdash (t/x)B, A^{m-1}, \Delta'$. L'hypothèse de récurrence, appliquée à π et ρ'_1, puis à ρ et π' donne une démonstration de $\Gamma, \Gamma' \vdash (t/x)B, \Delta, \Delta'$, et de $\Gamma, \Gamma', (t/x)B \vdash \Delta, \Delta'$. L'hypothèse de récurrence appliquée à $(t/x)B$ et les règles de contraction donnent une démonstration de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

– Si la dernière règle de π est la règle $\exists$-gauche, alors $A = \exists x\ B$ et la dernière règle de π' est la règle $\exists$-droite. Donc π a la forme

$$\dfrac{\dfrac{\rho}{\Gamma, A^{n-1}, B \vdash \Delta}}{\Gamma, A^{n-1}, \exists x\ B \vdash \Delta}\ \exists\text{-gauche}$$

et π' la forme

$$\dfrac{\dfrac{\rho'}{\Gamma' \vdash (t/x)B, A^{m-1}, \Delta'}}{\Gamma' \vdash \exists x\ B, A^{m-1}, \Delta'}\ \exists\text{-droite}$$

Comme x n'est pas une variable libre de Γ, de A ou de Δ, en substituant la variable x par le terme t dans la démonstration ρ, on obtient une démonstration ρ_1 de $\Gamma, A^{n-1}, (t/x)B \vdash \Delta$. L'hypothèse de récurrence, appliquée à π et ρ', puis à ρ_1 et π' donne une démonstration de $\Gamma, \Gamma' \vdash (t/x)B, \Delta, \Delta'$, et de $\Gamma, \Gamma', (t/x)B \vdash \Delta, \Delta'$. L'hypothèse de récurrence appliquée à $(t/x)B$ et les règles de contraction donnent une démonstration de $\Gamma, \Gamma' \vdash \Delta, \Delta'$.

Dans une seconde série de cas, la dernière règle de π ou celle de π' concerne une autre proposition que A. Par exemple, si la dernière règle de π est la règle $\wedge$-gauche, alors $\Gamma = \Gamma_1, B \wedge C$ et π a la forme

$$\dfrac{\dfrac{\rho}{\Gamma_1, A^n, B, C \vdash \Delta}}{\Gamma_1, A^n, B \wedge C \vdash \Delta}\ \wedge\text{-gauche}$$

on applique l'hypothèse de récurrence à ρ et π' et on obtient une démonstration de $\Gamma_1, \Gamma', B, C \vdash \Delta, \Delta'$. En utilisant cette même règle $\wedge$-gauche, on obtient une démonstration de $\Gamma_1, \Gamma', B \wedge C \vdash \Delta, \Delta'$, c'est-à-dire $\Gamma, \Gamma' \vdash \Delta, \Delta'$. On procède de même dans les autres cas.

Exercice 6.1

Montrer que le séquent $P(c) \vee Q(c) \vdash P(c)$ n'a pas de démonstration dans le calcul des séquents sans coupures. En déduire qu'il n'a pas de démonstration en déduction naturelle.

Proposition 6.6

On obtient un système équivalent au calcul des séquents sans coupures, si on restreint la règle *axiome* aux cas dans lesquels toutes les propositions du séquent démontré sont atomiques.

Démonstration. On montre, par récurrence sur le nombre de connecteurs et quantificateurs de $\Gamma \vdash \Delta$, que, dans le système restreint, les séquents de la forme $\Gamma \vdash \Delta$, où Γ et Δ ont une proposition en commun, sont démontrables.

6.2 La recherche de démonstrations dans le calcul des séquents sans coupures

Puisque toutes les propositions apparaissant dans les prémisses d'une règle du calcul des séquents sans coupures sont obtenues à partir de propositions qui apparaissent dans la conclusion de cette règle, la recherche d'une démonstration dans le calcul des séquents sans coupures ne demande pas de choisir une proposition dans l'ensemble infini des propositions du langage. Cependant, au cours d'une telle recherche, tous les choix n'ont pas disparu.

6.2.1 Les choix

Si on cherche, par exemple, une démonstration du séquent $P \wedge Q(c) \vdash P \wedge \exists x\ Q(x)$, on peut tout d'abord appliquer une règle à la proposition $P \wedge Q(c)$ ou à la proposition $P \wedge \exists x\ Q(x)$: c'est le *choix de la proposition* dans le séquent. Si on choisit la proposition $P \wedge \exists x\ Q(x)$, on peut appliquer ou bien la règle $\wedge$-droite ou bien la règle *contraction-droite* à cette proposition, c'est le *choix de la règle*. Si on choisit d'appliquer la règle $\wedge$-droite, on obtient deux séquents à démontrer $P \wedge Q(c) \vdash P$ et $P \wedge Q(c) \vdash \exists x\ Q(x)$. On doit alors choisir de chercher d'abord une démonstration du premier ou alors une démonstration du second, c'est *le choix du séquent*. Enfin, quand on cherche à démontrer le séquent $P \wedge Q(c) \vdash \exists x\ Q(x)$, si on applique la règle $\exists$-droite, on doit choisir le terme à substituer à la variable x, c'est le *choix du terme*.

Dans le cas général, à chaque étape de la recherche, on a un ensemble de séquents à démontrer, il faut donc choisir le séquent à traiter, ensuite il faut choisir la proposition à traiter dans ce séquent. Une fois cette proposition choisie, deux ou trois règles peuvent s'appliquer : la règle correspondant à son connecteur ou quantificateur principal et à sa latéralité, la règle de contraction

correspondant à sa latéralité et, parfois, la règle *axiome*. Enfin, si la règle choisie est la règle $\exists$-droite ou la règle $\forall$-gauche, il faut encore choisir le terme à substituer.

6.2.2 Les choix arborescents et les choix indifférents

Dans l'organisation d'une recherche, quand on se trouve face à un choix : explorer la voie A ou explorer la voie B et que l'on choisit la voie A, deux situations peuvent se produire. Dans certains cas, si la voie A mène à une impasse, il faut revenir en arrière et explorer la voie B. C'est par exemple ainsi que l'on explore les couloirs d'un labyrinthe à la recherche d'une sortie. Il se peut même qu'il faille commencer à explorer la voie B avant d'avoir abouti à un constat d'échec dans l'exploration de la voie A, qui peut ne pas terminer. On parle alors de *choix arborescent*. Dans d'autres cas, si la voie A mène à un échec, on sait que la voie B mènera aussi à un échec, on parle alors de choix *indifférent*. Ainsi, pour faire des œufs mimosa, on peut commencer par faire la mayonnaise ou par faire cuire les œufs. Si on commence par essayer de faire la mayonnaise et que l'on échoue, il est inutile d'essayer de faire cuire les œufs d'abord, cela ne changera rien pour la mayonnaise.

Dans l'organisation de la recherche d'une démonstration dans le calcul des séquents, le choix du séquent à traiter en premier est bien entendu indifférent : l'ordre dans lequel on cherche les démonstrations de $P \wedge Q(c) \vdash P$ et $P \wedge Q(c) \vdash \exists x\ Q(x)$ n'a pas d'importance, car chercher une démonstration d'un séquent ne change pas l'autre.

En revanche, les trois autres choix sont arborescents. L'arborescence du choix de la proposition est illustré par l'exemple suivant. On cherche, une démonstration du séquent $\exists x\ P(x), \forall y\ (P(y) \Rightarrow Q) \vdash Q$. Si on commence par appliquer la règle $\exists$-gauche à la proposition $\exists x\ P(x)$, on obtient le séquent $P(x), \forall y\ (P(y) \Rightarrow Q) \vdash Q$ et on peut appliquer la règle $\forall$-gauche à la proposition $\forall y\ (P(y) \Rightarrow Q)$, en choisissant le terme x, et conclure. En revanche, si on applique d'abord la règle $\forall$-gauche à la proposition $\forall y\ (P(y) \Rightarrow Q)$, avec ce même terme x, on obtient le séquent $\exists x\ P(x), P(x) \Rightarrow Q \vdash Q$, qui n'est pas démontrable. En particulier, la variable x apparaissant libre dans le séquent, appliquer la règle $\exists$-gauche à la proposition $\exists x\ P(x)$ demande de renommer la variable liée x en x' et on obtient alors le séquent $P(x'), P(x) \Rightarrow Q \vdash Q$ qui n'est pas démontrable.

L'arborescence du choix de la règle est illustré par l'exemple suivant. On cherche une démonstration du séquent $\vdash \exists x\ (P(x) \Rightarrow P(f(x)))$. Si on commence par appliquer la règle *contraction-droite* puis deux fois la règle $\exists$-droite avec les termes c et $f(c)$ on obtient le séquent $\vdash P(c) \Rightarrow P(f(c)), P(f(c)) \Rightarrow P(f(f(c)))$

qui se démontre sans peine. En revanche, si on applique la règle $\exists$-droite, on obtient un séquent de la forme $\vdash P(t) \Rightarrow P(f(t))$ qui n'est pas démontrable.

L'arborescence du choix du terme est illustré par l'exemple suivant. Si on veut démontrer le séquent $P(f(f(c))) \vdash \exists x\ P(f(x))$, il faut appliquer la règle $\exists$-droite avec le terme $f(c)$ et non avec le terme c.

6.2.3 Restreindre les choix

Un des trois types de choix arborescents, le choix du terme, est un choix parmi un ensemble infini de possibilités. Les deux autres en revanche, le choix de la proposition et le choix de la règle, sont des choix parmi un ensemble fini de possibilités. C'est donc le choix du terme qu'il faut chercher à restreindre en premier.

Quand on cherche une démonstration du séquent $P(f(f(c))) \vdash \exists x\ P(f(x))$, si on applique la règle $\exists$-droite à la proposition $\exists x\ P(f(x))$, on peut substituer la variable x par différents termes : c, $f(c)$, $f(f(c))$, $f(f(f(c)))$, ... Bien entendu, seul le terme $f(c)$ permettra de conclure. Au lieu d'énumérer tous les termes que l'on peut substituer et de les essayer les uns après les autres, on peut retarder le choix du terme en substituant x par une variable spéciale X, que l'on substituera à son tour, dans une seconde étape, au cours de laquelle une comparaison entre les propositions du séquent obtenu, $P(f(f(c)))$ et $P(f(X))$, suggérera la substitution $f(c)/X$.

Nous partitionnons donc les variables en deux ensembles infinis : les *variables ordinaires* que nous continuons de noter par une lettre minuscule et les *métavariables* que nous notons par une majuscule.

Définition 6.3 (Schéma de démonstration)

Un *schéma de démonstration* est une démonstration construite dans une variante du calcul des séquents sans coupures dans laquelle

- les règles $\exists$-droite et $\forall$-gauche sont restreintes au cas où le terme substitué t est une métavariable,
- la règle *axiome* est remplacée par une règle *axiome'* qui permet de démontrer n'importe quel séquent dont les propositions sont toute atomiques

$$\frac{}{\Gamma \vdash \Delta}\ \text{axiome'}\ \Gamma, \Delta \text{ atomiques}$$

Par exemple, l'arbre

$$\dfrac{\dfrac{}{P(f(f(c))) \vdash P(f(X))}\ \text{axiome'}}{P(f(f(c))) \vdash \exists x\ P(f(x))}\ \exists\text{-droite}$$

est un schéma de démonstration.

Proposition 6.7

Soit $\Gamma \vdash \Delta$ un séquent et h un entier. Le séquent $\Gamma \vdash \Delta$ a un nombre fini de schémas de démonstration de hauteur inférieure à h.

Démonstration. Par récurrence sur h.

Définition 6.4

Si σ est une substitution et Γ un multiensemble de propositions, le multiensemble $\sigma\Gamma$ est obtenu en appliquant la substitution σ à chaque élément de Γ.

Si σ est une substitution et π est une démonstration ou un schéma de démonstration, la démonstration ou le schéma de démonstration $\sigma\pi$ est obtenu en appliquant la substitution σ à chaque nœud de π.

Définition 6.5

Une substitution σ qui associe des termes $t_1, \ldots, t_n$ à des métavariables $X_1, \ldots, X_n$ *parfait* un schéma de démonstration π si l'arbre $\sigma\pi$ est une démonstration dans le calcul des séquents sans coupures et dans laquelle la règle *axiome* est restreinte aux séquents dont toutes les propositions sont atomiques, c'est-à-dire si

- pour chaque séquent $\Gamma \vdash \Delta$, démontré à l'aide de la règle *axiome'*, les multiensembles $\sigma\Gamma$ et $\sigma\Delta$ ont une proposition en commun
- et pour chaque utilisation des règle $\exists$-gauche ou $\forall$-droite dans π, la condition de fraîcheur des variables est vérifiée dans $\sigma\pi$, cela signifie que quand un nœud de π a la forme

$$\dfrac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x\ A, \Delta}\ \forall\text{-droite}$$

ou

$$\dfrac{\Gamma, A \vdash \Delta}{\Gamma, \exists x\ A \vdash \Delta}\ \exists\text{-gauche}$$

alors la variable x n'est libre ni dans $\sigma\Gamma$ ni dans $\sigma\Delta$. Autrement dit, x n'est pas libre dans Γ, Δ et si Y est libre dans Γ, Δ, alors x n'est pas libre dans σY.

Par exemple, la substitution $f(c)/X$ parfait le schéma de démonstration ci-avant et donne la démonstration

$$\dfrac{\dfrac{}{P(f(f(c))) \vdash P(f(f(c)))}\ \text{axiome}}{P(f(f(c))) \vdash \exists x\ P(f(x))}\ \exists\text{-droite}$$

Si on se donne un schéma de démonstration π, on peut décider s'il existe ou non une substitution qui le parfait.

En effet, pour cela, on doit trouver une substitution σ et, pour chaque séquent $\Gamma \vdash \Delta$ démontré par la règle *axiome'*, un couple formé d'une proposition atomique A de Γ et d'une proposition atomique B de Δ tel que $\sigma A = \sigma B$. Comme il y a, dans π, un nombre fini de séquents démontrés par la règle *axiome'*, et pour chacun d'eux un nombre fini de tels couples, on peut énumérer tous les choix possibles d'un couple de propositions atomiques par séquent. Il faut ensuite déterminer, pour chacun de ces choix, s'il existe une substitution σ telle que pour chaque couple (A, B), $\sigma A = \sigma B$.

Par exemple, dans le schéma ci-avant, on a un seul séquent démontré par la règle *axiome'*, et un seul choix pour le couple formé d'une proposition atomique de Γ et d'une proposition atomique de Δ. On doit donc trouver une substitution σ telle que $\sigma(P(f(X))) = \sigma(P(f(f(c))))$, c'est-à-dire une substitution σ qui soit une solution de l'équation

$$P(f(X)) = P(f(f(c)))$$

Cette équation s'appelle un *problème d'unification*. Pour résoudre ce problème, on procède de la manière suivante. Les solutions du problème

$$P(f(X)) = P(f(f(c)))$$

sont les mêmes que celles du problème

$$f(X) = f(f(c))$$

qui sont les mêmes que celles du problème

$$X = f(c)$$

et ce problème a une solution qui est la substitution $f(c)/X$.

Définition 6.6

Un *problème d'unification* est un système d'équations de la forme $t = u$. Une solution d'un tel problème est une substitution σ telle que pour toute équation $t = u$ du problème, les termes σt et σu sont identiques.

On résout les problèmes d'unification en utilisant l'algorithme d'unification de Robinson, qui rappelle, par certains aspects, l'algorithme du pivot de Gauss.

Définition 6.7 (L'algorithme d'unification de Robinson)

On choisit une équation dans le système.

- Si cette équation a la forme $f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$ où f est un symbole de prédicat, un symbole de fonction ou une variable, on la remplace par les équations $t_1 = u_1$, $\ldots$, $t_n = u_n$ et on résout le système obtenu.
- Si cette équation a la forme $f(t_1, \ldots, t_n) = g(u_1, \ldots, u_m)$ où f et g sont des symboles différents, on échoue.
- Si cette équation a la forme $X = X$, on la supprime du système et on résout le système obtenu.
- Si cette équation a la forme $X = t$ ou $t = X$, où X apparaît dans t et est distinct de t, on échoue.
- Si cette équation a la forme $X = t$ ou $t = X$ et que X n'apparaît pas dans t, on substitue X par t dans le reste du système, on résout le système obtenu, ce qui donne une substitution σ et on retourne la substitution $\sigma \cup \{\sigma t/X\}$.

La seule subtilité est dans le quatrième cas : une équation de la forme $X = f(X)$, par exemple, ne peut pas avoir de solution. Si elle en avait une, par exemple le terme u, celui-ci devrait être égal au terme $f(u)$ et donc le nombre de symboles de ce terme devrait vérifier l'équation $n = n + 1$. Ce test s'appelle le *test d'occurrence*, il est essentiel pour assurer la terminaison de l'algorithme d'unification. En effet, dans le cinquième cas, la terminaison est assurée par le fait que la variable X disparaît quand on substitue X par t et donc que l'algorithme est récursivement appelé sur un système qui comporte moins de variables. Cet algorithme d'unification termine donc toujours. Il échoue si le système d'équations n'a pas de solution et il retourne une solution si le système en a une.

Il se peut qu'un problème d'unification ait plusieurs solutions. Par exemple, l'équation $X = f(Y)$ a, entre autres substitutions, les solutions

$$f(c)/X, c/Y$$

$$f(f(c))/X, f(c)/Y$$
$$f(Z)/X, Z/Y$$
$$f(Y)/X$$

Une solution σ est dite *principale* si pour chaque solution τ il existe une substitution η telle que $\tau = \eta \circ \sigma$. Par exemple, les deux dernières substitutions ci-avant sont principales, mais pas les deux premières. On peut démontrer qu'un problème d'unification qui a une solution a toujours une solution principale, et que la solution calculée par l'algorithme d'unification est principale.

On peut également démontrer que si σ est une solution principale d'un problème d'unification et τ une solution quelconque, alors l'ensemble des variables ordinaires de τX est inclus dans celui des variables ordinaires de σX. Pour décider s'il existe une solution τ d'un problème d'unification qui vérifie des contraintes d'occurrence de la forme « x n'apparaît pas dans τY », il suffit donc d'observer si la solution principale σ, calculée par l'algorithme d'unification, vérifie ces contraintes.

Ainsi, on peut décider de l'existence d'une substitution qui parfait un schéma de démonstration. Ce qui montre que l'ensemble des séquents, qui ont une démonstration de hauteur inférieure à h, dans le calcul des séquents sans coupures, est décidable.

Il y a également de nombreuses manières de restreindre le choix de la proposition et le choix de la règle, on donne quelques exemples à l'exercice 6.4.

Définition 6.8

Une proposition *prénexe* est une proposition de la forme $\mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n\ C$ où $\mathcal{Q}_1, \dots, \mathcal{Q}_n$ sont des quantificateurs, $\forall$ ou $\exists$, et C est une proposition sans quantificateurs. Une proposition *existentielle* est une proposition de la forme $\exists x_1 \dots \exists x_n\ C$ où C est une proposition sans quantificateurs. Une proposition *universelle* est une proposition de la forme $\forall x_1 \dots \forall x_n\ C$ où C est une proposition sans quantificateurs.

Une proposition sans quantificateurs est une proposition *normale conjonctive* si elle est de la forme $\top$ ou $C_1 \wedge (\dots \wedge C_n)$ où chaque proposition C_i est de la forme $\bot$ ou $D_1 \vee (\dots \vee D_m)$ où chaque D_i est une proposition atomique ou la négation d'une proposition atomique.

Exercice 6.2 (Transformer le séquent : les quantificateurs)

Cet exercice demande d'avoir fait l'exercice 1.5.

1. Montrer que si le séquent $\vdash A \Leftrightarrow A'$ est démontrable alors le séquent $\Gamma, A \vdash \Delta$ est démontrable si et seulement si le séquent $\Gamma, A' \vdash \Delta$ est

démontrable et le séquent $\Gamma \vdash A, \Delta$ est démontrable si et seulement si le séquent $\Gamma \vdash A', \Delta$ est démontrable.

2. Montrer que, si x n'est pas une variable libre de B, les propositions $((\forall x\ A) \wedge B) \Leftrightarrow \forall x\ (A \wedge B)$, $(B \wedge (\forall x\ A)) \Leftrightarrow \forall x\ (B \wedge A)$, $((\exists x\ A) \wedge B) \Leftrightarrow \exists x\ (A \wedge B)$, $(B \wedge (\exists x\ A)) \Leftrightarrow \exists x\ (B \wedge A)$, $((\forall x\ A) \vee B) \Leftrightarrow \forall x\ (A \vee B)$, $(B \vee (\forall x\ A)) \Leftrightarrow \forall x\ (B \vee A)$, $((\exists x\ A) \vee B) \Leftrightarrow \exists x\ (A \vee B)$, $(B \vee (\exists x\ A)) \Leftrightarrow \exists x\ (B \vee A)$, $((\forall x\ A) \Rightarrow B) \Leftrightarrow \exists x\ (A \Rightarrow B)$, $(B \Rightarrow (\forall x\ A)) \Leftrightarrow \forall x\ (B \Rightarrow A)$, $((\exists x\ A) \Rightarrow B) \Leftrightarrow \forall x\ (A \Rightarrow B)$, $(B \Rightarrow (\exists x\ A)) \Leftrightarrow \exists x\ (B \Rightarrow A)$, $(\neg(\forall x\ A)) \Leftrightarrow \exists x\ \neg A$ et $(\neg(\exists x\ A)) \Leftrightarrow \forall x\ \neg A$ sont démontrables.

   Montrer que, pour toute proposition A, il existe une proposition prénexe A', telle que la proposition $A \Leftrightarrow A'$ soit démontrable.

   Montrer que le séquent $\vdash A$ est démontrable si et seulement si le séquent $\vdash A'$ est démontrable.

3. On rappelle que, d'après la proposition 1.7, le séquent $\vdash A$ est démontrable si et seulement si le séquent $\vdash \neg\neg A$ est démontrable.

   Montrer que le séquent $\vdash A$ est démontrable si et seulement si le séquent $\neg A \vdash$ est démontrable.

   Montrer que, pour toute proposition A, il existe une proposition prénexe A', telle que le séquent $A' \vdash$ soit démontrable si et seulement si le séquent $\vdash A$ est démontrable.

4. Montrer que, pour toute proposition A, il existe une proposition universelle A', telle que le séquent $A' \vdash$ soit démontrable si et seulement si le séquent $\vdash A$ est démontrable. Indice : utiliser le théorème de Skolem 2.3.

   Montrer que, pour toute proposition A, il existe une proposition existentielle A', telle que le séquent $\vdash A'$ soit démontrable si et seulement si le séquent $\vdash A$ est démontrable.

Exercice 6.3 (Transformer le séquent : les connecteurs)

Cet exercice demande d'avoir fait l'exercice 6.2.

1. Montrer que les propositions $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$, $(\neg\top) \Leftrightarrow \bot$, $(\neg\bot) \Leftrightarrow \top$, $(\neg(A \wedge B)) \Leftrightarrow ((\neg A) \vee (\neg B))$, $(\neg(A \vee B)) \Leftrightarrow ((\neg A) \wedge (\neg B))$ et $(\neg\neg A) \Leftrightarrow A$ sont démontrables.

   Montrer que, pour toute proposition sans quantificateurs A, il existe une proposition A' sans quantificateurs, qui ne contient pas le symbole $\Rightarrow$ et dans laquelle la négation n'est appliquée qu'à des propositions atomiques, telle que la proposition $A \Leftrightarrow A'$ soit démontrable.

2. Montrer que les propositions $((A \wedge B) \wedge C) \Leftrightarrow (A \wedge (B \wedge C))$, $((A \vee B) \vee C) \Leftrightarrow (A \vee (B \vee C))$, $(A \vee (B \wedge C)) \Leftrightarrow (A \vee B) \wedge (A \vee C)$, $((A \wedge B) \vee C) \Leftrightarrow$

$(A \vee C) \wedge (B \vee C)$, $(\top \vee A) \Leftrightarrow \top$, $(A \vee \top) \Leftrightarrow \top$, $(\bot \vee A) \Leftrightarrow A$ et $(A \vee \bot) \Leftrightarrow A$ sont démontrables.

Montrer que, pour toute proposition sans quantificateurs A, il existe une proposition normale conjonctive A', telle que la proposition $A \Leftrightarrow A'$ soit démontrable.

3. Montrer que, pour toute proposition A, il existe une proposition universelle A' de la forme $\forall x_1 \ldots \forall x_n\ C$, où C est une proposition normale conjonctive telle que le séquent $A' \vdash$ soit démontrable si et seulement si le séquent $\vdash A$ est démontrable.

   Montrer que la proposition $(\forall x\ (A \wedge B)) \Leftrightarrow ((\forall x\ A) \wedge (\forall x\ B))$ est démontrable. Montrer que le séquent $\Gamma, A \wedge B \vdash \Delta$ est démontrable si et seulement si le séquent $\Gamma, A, B \vdash \Delta$ est démontrable.

   Montrer que, pour toute proposition A, il existe des propositions $C_1, \ldots, C_p$ de la forme $\bot$ ou $\forall x_1 \ldots \forall x_n\ (D_1 \vee (\ldots \vee D_m))$, où chaque D_i est une proposition atomique ou la négation d'une proposition atomique, telles que le séquent $\vdash A$ soit démontrable si et seulement si le séquent $C_1, \ldots, C_p \vdash$ est démontrable.

Exercice 6.4

Cet exercice demande d'avoir fait l'exercice 6.2.

1. Montrer que l'on obtient un système équivalent au calcul des séquents sans coupures si on restreint la règle *contraction-gauche* aux propositions de la forme $\forall x\ A$ et la règle *contraction-droite* aux propositions de la forme $\exists x\ A$.
2. Montrer que la démonstration d'une proposition existentielle dans le calcul des séquents sans coupures n'utilise jamais les règles $\exists$-gauche et $\forall$-droite. Montrer que dans le calcul des séquents sans coupures, privé des règles $\exists$-gauche et $\forall$-droite, le choix de la proposition est indifférent.
3. Écrire un programme de recherche de démonstrations dans le calcul des séquents.

Exercice 6.5 (Le théorème de Herbrand)

Soit A une proposition prénexe close de la forme $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n\ C$. On appelle *instance close* de A, une proposition close de la forme σC, où σ est une substitution de domaine $x_1, \ldots, x_n$.

Soient $A_1, \ldots, A_n$ des propositions existentielles closes et Γ et Δ des multiensembles de propositions closes sans quantificateurs. Montrer que si le langage contient au moins une constante, alors le séquent $\Gamma \vdash A_1, \ldots, A_n, \Delta$ est démontrable dans le calcul des séquents sans coupures si et seulement s'il existe

des instances closes $A_1^1, \ldots, A_1^{p_1}$ de A_1, $\ldots$, $A_n^1, \ldots, A_n^{p_n}$ de A_n, telles que le séquent sans quantificateurs $\Gamma \vdash A_1^1, \ldots, A_1^{p_1}, \ldots, A_n^1, \ldots, A_n^{p_n}, \Delta$ soit démontrable dans le calcul des séquents sans coupures.

Soient $A_1, \ldots, A_n$ des propositions existentielles closes. Montrer que si le langage contient au moins une constante, alors le séquent $\vdash A_1, \ldots, A_n$ est démontrable dans le calcul des séquents sans coupures si et seulement s'il existe des instances closes $A_1^1, \ldots, A_1^{p_1}$ de A_1, $\ldots$, $A_n^1, \ldots, A_n^{p_n}$ de A_n, telles que le séquent sans quantificateurs $\vdash A_1^1, \ldots, A_1^{p_1}, \ldots, A_n^1, \ldots, A_n^{p_n}$ soit démontrable dans le calcul des séquents sans coupures.

Soient $A_1, \ldots, A_n$ des propositions universelles closes. Montrer, de même, que si le langage contient au moins une constante, le séquent $A_1, \ldots, A_n \vdash$ est démontrable dans le calcul des séquents sans coupures si et seulement s'il existe des instances closes $A_1^1, \ldots, A_1^{p_1}$ de A_1, $\ldots$, $A_n^1, \ldots, A_n^{p_n}$ de A_n, telles que le séquent sans quantificateurs $A_1^1, \ldots, A_1^{p_1}, \ldots, A_n^1, \ldots, A_n^{p_n} \vdash$ soit démontrable dans le calcul des séquents sans coupures.

Exercice 6.6 (La résolution)

Cet exercice demande d'avoir fait les exercices 6.2, 6.3 et 6.5.

Une *clause* est un ensemble fini de propositions, dans lequel chaque proposition est une proposition atomique ou la négation d'une proposition atomique.

Si $C = \{A_1, \ldots, A_n\}$ est une clause, on écrit $\overline{\forall} C$ la proposition $\forall x_1 \ldots \forall x_p$ $(A_1 \vee \ldots \vee A_n)$, où $x_1, \ldots, x_p$ sont les variables libres de $A_1, \ldots, A_n$ et $\overline{\forall} \varnothing = \bot$ par convention.

Soit E un ensemble de clauses, on considère l'ensemble de clauses G inductivement défini par les trois règles suivantes

- si C appartient à E, alors C appartient à G,
- C appartient à G, alors $(t/x)C$ appartient à G,
- si $C_1 \cup \{A\}$ et $C_2 \cup \{\neg A\}$ appartiennent à G, alors $C_1 \cup C_2$ appartient à G.

On écrit $E \rightsquigarrow C$ pour exprimer que la clause C appartient à l'ensemble G.

1. Soit E l'ensemble formé des quatre clauses

$$P(a, b)$$

$$P(b, c)$$

$$\neg P(x, y), \neg P(y, z), G(x, z)$$

$$\neg G(a, c)$$

Donner une dérivation de $E \rightsquigarrow \varnothing$.

2. Montrer que pour toute proposition A, il existe un ensemble de clauses $C_1, \ldots, C_n$, tel que le séquent $\vdash A$ soit démontrable si et seulement si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable. Quel est l'ensemble de clauses associé à la proposition suivante ?

$$(P(a,b) \wedge P(b,c) \wedge \forall x \forall y \forall z \ ((P(x,y) \wedge P(y,z)) \Rightarrow G(x,z))) \Rightarrow G(a,c)$$

3. On veut montrer que si $C_1, \ldots, C_n \rightsquigarrow \varnothing$, alors le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable. On montre plus généralement que si $C_1, \ldots, C_n \rightsquigarrow D$, alors le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash \overline{\forall} D$ est démontrable.

   Montrer que si $D = (t/x)C$, alors le séquent $\overline{\forall} C \vdash \overline{\forall} D$ est démontrable.

   Montrer que si $C_1 = C_1' \cup \{A\}$ et $C_2 = C_2' \cup \{\neg A\}$ et $D = C_1' \cup C_2'$, alors le séquent $\overline{\forall} C_1, \overline{\forall} C_2 \vdash \overline{\forall} D$ est démontrable.

   Soit $C_1, \ldots, C_n$ un ensemble de clauses, montrer que si $C_1, \ldots, C_n \rightsquigarrow D$, alors le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash \overline{\forall} D$ est démontrable.

   Montrer que si le séquent $\Gamma \vdash \bot$ est démontrable, alors le séquent $\Gamma \vdash$ l'est aussi.

   Montrer que si $C_1, \ldots, C_n \rightsquigarrow \varnothing$, alors le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable.

4. On veut maintenant montrer la réciproque, c'est-à-dire que si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable, alors $C_1, \ldots, C_n \rightsquigarrow \varnothing$.

   Soit E un ensemble de clauses et C et D deux clauses. Montrer que si $E \rightsquigarrow C$ et $E \cup \{C\} \rightsquigarrow D$, alors $E \rightsquigarrow D$.

   Soit D est une clause close, $E = \{C_1, \ldots, C_n\}$ et $E' = \{C_1', \ldots, C_n'\}$ deux ensembles de clauses closes, tels que pour tout i, C_i' est ou bien la clause C_i ou bien la clause $C_i \cup D$ et C une clause close. Montrer que si $E \rightsquigarrow C$, alors ou bien $E' \rightsquigarrow C$ ou bien $E' \rightsquigarrow C \cup D$. Montrer que si $E \rightsquigarrow \varnothing$, alors ou bien $E' \rightsquigarrow \varnothing$ ou bien $E' \rightsquigarrow D$. Montrer que si C et C' sont deux clauses closes et $E \cup \{C\} \rightsquigarrow \varnothing$ et $E \cup \{C'\} \rightsquigarrow \varnothing$, alors $E, (C \cup C') \rightsquigarrow \varnothing$.

   Soient $C_1, \ldots, C_n$ des clauses closes et $P_1, \ldots, P_m$ des propositions atomiques closes. Montrer que si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash P_1, \ldots, P_m$ est démontrable, alors $C_1, \ldots, C_n, \neg P_1, \ldots, \neg P_m \rightsquigarrow \varnothing$. Montrer que si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable, alors $C_1, \ldots, C_n \rightsquigarrow \varnothing$.

   Soient $C_1, \ldots, C_n$ des clauses quelconques. Montrer que si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable, alors $C_1, \ldots, C_n \rightsquigarrow \varnothing$. Indice : utiliser le théorème de Herbrand.

   Soient $C_1, \ldots, C_n$ des clauses quelconques. Montrer que le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable si et seulement si $C_1, \ldots, C_n \rightsquigarrow \varnothing$.

5. Les trois règles ci-avant ne peuvent pas encore être utilisées comme un algorithme de recherche de démonstrations, car la deuxième demande de choisir

un terme dans un ensemble infini. On introduit donc un autre ensemble de règles appelé *règles de résolution*.

Soit E un ensemble de clauses, on considère l'ensemble de clauses G inductivement défini par les deux règles suivantes

- si C appartient à E, alors C appartient à G,
- si $C \cup \{A_1, \ldots, A_n\}$ et $C' \cup \{\neg B_1, \ldots, \neg B_m\}$ sont deux clauses de G dans lesquelles on a renommé les variables de manière à ce qu'elles ne partagent pas de variables, et σ est une solution principale du problème d'unification $A_1 = \ldots = A_n = B_1 = \ldots = B_m$, alors la clause $\sigma(C \cup C')$ appartient à G.

On écrit $E \hookrightarrow C$ pour exprimer que C appartient à l'ensemble G.

Soit E l'ensemble de clause de la question (1.). Donner une dérivation de $E \hookrightarrow \varnothing$.

Soit E l'ensemble de clauses. Montrer que si $E \hookrightarrow D$, alors $E \rightsquigarrow D$. Montrer que si $E \hookrightarrow \varnothing$, alors $E \rightsquigarrow \varnothing$.

Montrer que s'il existe un ensemble E' contenant des clauses de la forme σC, où C est une clause de E et σ une substitution, tel que $E' \rightsquigarrow D'$, alors il existe une clause D et une substitution τ telle que $E \hookrightarrow D$ et $D' = \tau D$. Montrer que si $E \rightsquigarrow \varnothing$, alors $E \hookrightarrow \varnothing$.

Soit E un ensemble de clauses. Montrer que $E \rightsquigarrow \varnothing$ si et seulement si $E \hookrightarrow \varnothing$.

Soit A une proposition et $C_1, \ldots, C_n$ un ensemble de clauses, tel que le séquent $\vdash A$ soit démontrable si et seulement si le séquent $\overline{\forall} C_1, \ldots, \overline{\forall} C_n \vdash$ est démontrable. Montrer que le séquent $\vdash A$ est démontrable si et seulement si $C_1, \ldots, C_n \hookrightarrow \varnothing$.

6. Écrire un programme de recherche de démonstrations utilisant la résolution.

7
Des théories décidables

Nous avons vu, au chapitre 5, que la logique des prédicats était indécidable mais, d'une part, qu'elle était semi-décidable et, d'autre part, qu'en ajoutant des axiomes, on rendait parfois la démontrabilité décidable. La semi-décidabilité nous a mené, au chapitre 6, à développer des algorithmes de recherche de démonstrations, qui ne terminent pas quand la proposition dont on recherche une démonstration n'est pas démontrable. Nous allons voir, dans ce chapitre, un exemple d'algorithme qui permet de décider la démontrabilité dans une théorie particulière.

Diverses méthodes peuvent être employées pour décider la démontrabilité dans une théorie. L'une d'elle consiste à montrer que si une proposition n'est pas démontrable dans cette théorie, alors il existe un modèle fini de la théorie dans laquelle cette proposition n'est pas valide. Ainsi, en énumérant d'une part les démonstrations, d'autre part les modèles finis, on finit par trouver une démonstration dans le cas où la proposition est démontrable et un modèle dans le cas où elle ne l'est pas.

Une autre méthode, la méthode d'*élimination des quantificateurs*, consiste à montrer, d'une part, que la démontrabilité d'une proposition close sans quantificateurs peut se déterminer par une simple analyse de cette proposition et, d'autre part, que toute proposition close peut se transformer en une proposition close sans quantificateurs qui lui est équivalente. Nous utilisons cette méthode pour démontrer la décidabilité de l'ensemble des propositions du langage $0, 1, +, -, \leq$ valides dans $\mathbb{Z}$. Cet ensemble étant décidable, on peut poser chacun de ses éléments en axiome. On obtient ainsi une théorie cohérente, complète et décidable.

Pour démontrer ce théorème, on doit étendre le langage ci-avant en ajoutant pour chaque entier naturel n non nul, un prédicat unaire $Mult_n$, qui caractérise les multiples de n.

Définition 7.1

Soit $\mathcal{L}$ le langage constitué des constantes 0 et 1, des symboles de fonction binaire $+$ et $-$, d'un symbole de prédicat binaire $\leq$ et, pour chaque entier naturel non nul n, d'un symbole de prédicat unaire $Mult_n$.

Si n est un entier positif, on écrit n l'entier $1+1+\ldots+1$ avec n occurrences du symbole 1 et si n est négatif, on écrit n l'entier $0-1-1-\ldots-1$ avec $-n$ occurrences du symbole 1. De même, on écrit $1.x$ le terme x, $2.x$ le terme $x+x$, $3.x$ le terme $x+x+x$, $\ldots$, $(-1).x$ le terme $0-x$, $(-2).x$ le terme $0-x-x$, $\ldots$ et $0.x$ le terme 0.

Définition 7.2 (Le modèle $\mathbb{Z}$)

Le modèle $\mathbb{Z}$ est formé de l'ensemble $\mathbb{Z}$, des entiers 0 et 1, de l'addition et de la soustraction de $\mathbb{Z}$ de la relation d'ordre sur $\mathbb{Z}$ et, pour tout entier naturel non nul n, de la fonction caractéristique de l'ensemble des multiples de n.

Commençons par un exemple. Soit A la proposition $1 \leq 3.x \wedge x \leq 7-x$, qui contient une unique variable libre x. On veut décider si la proposition $\exists x\ A$ est valide ou non dans $\mathbb{Z}$. Pour cela, dans chaque inéquation, on commence par rassembler les x d'un côté du signe $\leq$ et les autres termes de l'autre. Puis on multiplie la première inéquation par 2 et la seconde par 3 de manière à obtenir le même coefficient pour x dans toutes les inéquations. On obtient alors la proposition équivalente $\exists x\ (2 \leq 6.x \wedge 6.x \leq 21)$. On fait alors un changement de variable, qui donne la proposition équivalente à $\exists x'\ (2 \leq x' \wedge x' \leq 21 \wedge Mult_6(x'))$. On doit donc décider de l'existence d'un multiple de 6 dans l'intervalle des nombres compris entre 2 et 21 et la réponse est positive.

Si, maintenant, la proposition A contient d'autres variables que x, on ne veut plus simplement décider si la proposition $\exists x\ A$, qui contient des variables libres, est valide ou non, mais on veut la transformer en une proposition équivalente, sans quantificateurs et qui contient ces mêmes variables. Considérons, par exemple, la proposition $1 \leq 3.x \wedge x \leq y-x$. On commence, comme ci-avant, par rassembler les x d'un côté du signe $\leq$ et les autres termes de l'autre puis on multiplie chaque inégalité de manière à obtenir le même coefficient pour x dans chaque inéquation et on effectue un changement de variable. On obtient

la proposition, équivalente à $\exists x\ A$, suivante $\exists x'\ (2 \leq x' \wedge x' \leq 3.y \wedge \mathit{Mult}_6(x'))$. Appelons A' la proposition $2 \leq x' \wedge x' \leq 3.y \wedge \mathit{Mult}_6(x')$.

Fixons une valeur q pour y et supposons qu'il existe un entier p qui vérifie cette proposition. Deux cas peuvent alors se produire. Ou bien, tous les nombres supérieurs à p et congrus à p modulo 6 la vérifient également, ou bien, comme dans cet exemple, ce n'est pas le cas. Dans ce cas, il existe un entier p' qui vérifie la proposition et tel que $p'+6$ ne la vérifie pas. Cela signifie qu'il existe une inéquation, ici $x' \leq 3.y$, qui a changé d'avis entre p' et $p'+6$. Donc $p' \leq 3.q < p'+6$. Il existe donc un entier j compris entre 0 et 5 tel que $3.q = p' + j$ et donc $p' = 3.q - j$ vérifie la proposition ci-avant. Autrement dit, la proposition A' dans laquelle on substitue y par q et x' par $3.q - j$ pour un certain j compris entre 0 et 5 est valide dans $\mathbb{Z}$.

Soit B la proposition sans quantificateurs $(3.y/x')A' \vee ((3.y-1)/x')A' \vee ((3.y-2)/x')A' \vee ((3.y-3)/x')A' \vee ((3.y-4)/x')A' \vee ((3.y-5)/x')A'$. La proposition B, dans laquelle on substitue y par q est valide dans $\mathbb{Z}$. Réciproquement, si cette proposition est valide dans $\mathbb{Z}$, alors $\exists x'\ A'$ est également valide.

La proposition ci-après généralise cette construction.

Proposition 7.1

Soit A une proposition sans quantificateurs dans le langage $\mathcal{L}$, il existe une proposition B sans quantificateurs telle que la proposition $(\exists x\ A) \Leftrightarrow B$ soit valide dans le modèle $\mathbb{Z}$.

Démonstration. On commence par supprimer dans A les implications en remplaçant les propositions de la forme $C \Rightarrow D$ par la proposition équivalente $\neg C \vee D$. On supprime ensuite les négations en remplaçant les propositions de la forme $\neg\top$ par $\bot$, $\neg\bot$ par $\top$, $\neg(C \wedge D)$ par $\neg C \vee \neg D$, $\neg(C \vee D)$ par $\neg C \wedge \neg D$, $\neg\neg C$ par C, $\neg\ t \leq u$ par $u+1 \leq t$ et $\neg\ \mathit{Mult}_n(t)$ par $\mathit{Mult}_n(t+1) \vee \ldots \vee \mathit{Mult}_n(t+n-1)$ jusqu'à la disparition complète du symbole $\neg$. On obtient alors une proposition formée avec les connecteurs $\top$, $\bot$, $\wedge$, $\vee$ à partir de propositions atomiques de la forme $t \leq u$ ou $\mathit{Mult}_n(t)$.

On rassemble ensuite, dans chaque inéquation, les x d'un côté du signe $\leq$ et les autres termes de l'autre. Puis on remplace chaque proposition de la forme $t \leq u$ par la proposition équivalente $k.t \leq k.u$ pour un entier k strictement positif et chaque proposition de la forme $\mathit{Mult}_n(t)$ par $\mathit{Mult}_{kn}(k.t)$ de manière à obtenir le même coefficient s pour x dans toutes les propositions atomiques. On remplace ensuite tous les termes de la forme $s.x$ par une variable x' et on ajoute la proposition atomique $\mathit{Mult}_s(x')$. On obtient alors une proposition, équivalente à la proposition $\exists x\ A$ de la forme $\exists x'\ A'$ où A' est formée avec les connecteurs $\top$, $\bot$, $\wedge$, $\vee$ à partir de propositions atomiques de la forme $x' \leq t$,

$t \leq x'$, $0 \leq t$, $Mult_n(x'+t)$ et $Mult_n(t)$ où t est un terme qui ne contient pas la variable x'.

Soit r un multiple commun de tous les entiers n tels que la proposition atomique $Mult_n(x'+t)$ apparaisse dans A'.

Si on fixe la valeur des variables distinctes de x', la valeur de vérité d'une telle proposition est une fonction de la valeur associée à x' qui est périodique de période r à partir d'un certain rang. En effet, au-delà d'une certaine valeur, les propositions de la forme $x' \leq t$ sont toujours fausses et celles de la forme $t \leq x'$ toujours vraies, seules celles de la forme $Mult_n(x'+t)$ changent de valeur selon une période r.

Soit E l'ensemble de tous les termes t tels que la proposition atomique $x' \leq t$ apparaisse dans A'. Soit A'' la proposition obtenue en remplaçant dans A' les propositions de la forme $x' \leq t$ par $\bot$ et les propositions de la forme $t \leq x'$ par $\top$ et soit B la disjonction de toutes les propositions de la forme

- $(i/x')A''$ où i est un entier compris entre 0 et $r-1$,
- $((t-j)/x')A'$ où t est un terme de E et j un entier compris entre 0 et $r-1$.

Montrons que la proposition $(\exists x'\ A') \Leftrightarrow B$ est valide dans $\mathbb{Z}$.

Soient $y_1, \ldots, y_n$ les variables de A' distinctes de x'. On écrit $A'[p, q_1, \ldots, q_n]$ la proposition $(p/x', q_1/y_1, \ldots, q_n/y_n)A'$, $A''[p, q_1, \ldots, q_n]$ la proposition $(p/x', q_1/y_1, \ldots, q_n/y_n)A''$ et $B[q_1, \ldots, q_n]$ la proposition $(q_1/y_1, \ldots, q_n/y_n)B$. On veut montrer que pour tout $q_1, \ldots, q_n$, il existe un entier p tel que $A'[p, q_1, \ldots, q_n]$ soit valide si et seulement si $B[q_1, \ldots, q_n]$ est valide.

Supposons qu'il existe un entier p tel que $A'[p, q_1, \ldots, q_n]$ soit valide, dans ce cas, ou bien, pour tout v, $A'[p+vr, q_1, \ldots, q_n]$ est valide, ou bien non.

Dans le premier cas, il existe des entiers p' arbitrairement grands tels que $A'[p', q_1, \ldots, q_n]$ soit valide, et pour p' suffisamment grand $A'[p', q_1, \ldots, q_n]$ est équivalent à $A''[p', q_1, \ldots, q_n]$. Il existe donc un entier p' tel que $A''[p', q_1, \ldots, q_n]$ soit valide. Or, pour tout p, $A''[p, q_1, \ldots, q_n]$ est équivalent à $A''[p-r, q_1, \ldots, q_n]$. Il existe donc un entier i compris entre 0 et $r-1$ tel que $A''[i, q_1, \ldots, q_n]$ soit valide. La proposition $B[q_1, \ldots, q_n]$ est donc valide.

Dans le second cas, il existe un entier p' tel que $A'[p', q_1, \ldots, q_n]$ soit valide, mais pas $A'[p'+r, q_1, \ldots, q_n]$. Il existe donc une proposition atomique de la forme $x' \leq t$ qui est vérifiée en p' mais pas en $p'+r$. Écrivons $t[q_1, \ldots, q_n]$ le terme $(q_1/y_1, \ldots, q_n/y_n)t$. On a $p' \leq t[q_1, \ldots, q_n]$, mais $t[q_1, \ldots, q_n] < p'+r$, il existe donc un entier j compris entre 0 et $r-1$ tel que $p' = t[q_1, \ldots, q_n] - j$. La proposition $B[q_1, \ldots, q_n]$ est donc valide.

Réciproquement, si $B[q_1, \ldots, q_n]$ est valide, alors, ou bien il existe un entier i tel que $A''[i, q_1, \ldots, q_n]$ soit valide ou bien il existe un élément t de E et un entier j tels que $A'[t[q_1, \ldots, q_m] - j, q_1, \ldots, q_n]$ soit valide. Dans ce second cas, il existe un entier p tel que $A'[p, q_1, \ldots, q_n]$ soit valide. Dans le premier, comme

pour tout p, tel que $A''[p, q_1, \ldots, q_n]$ est équivalent à $A''[p + r, q_1, \ldots, q_n]$, il existe des entiers p arbitrairement grands tels que $A''[p, q_1, \ldots, q_n]$ soit valide, et pour p assez grand $A''[p, q_1, \ldots, q_n]$ est équivalent à $A'[p, q_1, \ldots, q_n]$. Il existe donc un entier p tel que $A'[p, q_1, \ldots, q_n]$ soit valide.

Proposition 7.2

Soit A une proposition du langage $\mathcal{L}$, alors il existe une proposition B sans quantificateurs telle que $A \Leftrightarrow B$ soit valide dans $\mathbb{Z}$.

Démonstration. On remplace les propositions de la forme $\forall x\ C$ par la proposition équivalente $\neg\exists x \neg C$ et on conclut avec une démonstration par récurrence sur la structure de la proposition ainsi obtenue, en utilisant la proposition 7.1 dans le cas du quantificateur existentiel.

Théorème 7.1

L'ensemble des propositions formées dans le langage $0, 1, +, -, \leq$ et valide dans $\mathbb{Z}$ est décidable.

Démonstration. La validité des propositions closes et sans quantificateurs est évidemment décidable, celle des propositions quelconques s'en déduit par la proposition 7.2.

On peut en déduire un résultat similaire pour les entiers naturels.

Théorème 7.2 (Presburger)

L'ensemble des propositions formées dans le langage $0, S, +, =$ et valide dans $\mathbb{N}$ est décidable.

Démonstration. À chaque proposition A du langage $0, S, +, =$ on associe une proposition $|A|$ du langage $0, 1, +, -, \leq$ telle que pour toute proposition close A, A est valide dans $\mathbb{N}$ si et seulement si $|A|$ est valide dans $\mathbb{Z}$.

- $|0| = 0$, $|x| = x$, $|S(t)| = |t| + 1$, $|t + u| = |t| + |u|$,
- $|t = u| = |t| \leq |u| \wedge |u| \leq |t|$,
- $|\top| = \top$, $|\bot| = \bot$, $|\neg A| = \neg|A|$, $|A \wedge B| = |A| \wedge |B|$, $|A \vee B| = |A| \vee |B|$, $|A \Rightarrow B| = |A| \Rightarrow |B|$,
- $|\forall x\ A| = \forall x\ (0 \leq x \Rightarrow |A|)$, $|\exists x\ A| = \exists x\ (0 \leq x \wedge |A|)$.

8
La constructivité

Si un ensemble contient l'entier 0, mais pas l'entier 2, on peut montrer qu'il existe un entier qui appartient à cet ensemble, mais dont le successeur ne lui appartient pas : il faut bien, en effet, qu'à un moment ou a un autre, la suite des entiers sorte de l'ensemble. On peut même montrer que ce nombre est égal ou bien à 0 ou bien à 1. Mais, on ne peut pas montrer que ce nombre est égal à 0 ni qu'il est égal à 1, car il faudrait pour cela savoir si le nombre 1 appartient à l'ensemble ou non.

En logique des prédicats cela se traduit par le fait que le séquent

$$\Gamma \vdash \exists x\ (P(x) \wedge \neg P(S(x)))$$

où $\Gamma = P(0), \neg P(2)$ est démontrable

$$\dfrac{\dfrac{}{\Gamma \vdash P(1) \vee \neg P(1)} \quad \dfrac{\dfrac{\cdots}{\Gamma, P(1) \vdash P(1) \wedge \neg P(2)}}{\Gamma, P(1) \vdash \exists x\ (P(x) \wedge \neg P(S(x)))} \quad \dfrac{\dfrac{\cdots}{\Gamma, \neg P(1) \vdash P(0) \wedge \neg P(1)}}{\Gamma, \neg P(1) \vdash \exists x\ (P(x) \wedge \neg P(S(x)))}}{\Gamma \vdash \exists x\ (P(x) \wedge \neg P(S(x)))}$$

En revanche, pour chaque terme t, le séquent $P(0), \neg P(2) \vdash P(t) \wedge \neg P(S(t))$ n'est pas démontrable. En effet, en prenant $\mathcal{M} = \mathbb{N}$, en interprétant 0 et S de manière évidente et P successivement par la fonction caractéristique de la paire $\{0, 1\}$ et du singleton $\{0\}$, on obtient deux modèles dans lesquels le terme t a la même dénotation et qui réfutent le séquent ci-avant, le premier dans le cas où la dénotation de t est nulle et le second dans le cas où elle est non nulle.

Définition 8.1 (La propriété du témoin)

On dit qu'un ensemble de propositions a la *propriété du témoin* si chaque fois

qu'il contient une proposition de la forme $\exists x\ A$, il contient la proposition $(t/x)A$ pour un certain terme t.

L'ensemble des propositions démontrables dans la théorie $P(0), \neg P(2)$ n'a donc pas la propriété du témoin.

On peut montrer, de même, que l'ensemble des propositions démontrables dans la théorie vide n'a pas la propriété du témoin. Il suffit pour cela de considérer la proposition

$$\exists x\ ((P(0) \wedge \neg P(2)) \Rightarrow (P(x) \wedge \neg P(S(x))))$$

Exercice 8.1

Montrer que la proposition $\exists x\ (P(x) \vee \neg P(S(x)))$ est démontrable, mais qu'il n'existe pas de terme t tel que $P(t) \vee \neg P(S(t))$ soit démontrable.

Dans la démonstration du séquent $P(0), \neg P(2) \vdash \exists x\ (P(x) \wedge \neg P(S(x)))$, l'utilisation du tiers exclu pour montrer la proposition $P(1) \vee \neg P(1)$ semble essentielle. On peut donc se demander si ce séquent peut être démontré sans utiliser le tiers exclu. Comme nous allons le voir, ce n'est pas le cas, car l'ensemble des propositions démontrables en logique des prédicats sans utiliser le tiers exclu a la propriété du témoin.

Définition 8.2 (Démonstration constructive)

Une démonstration en déduction naturelle est *constructive* si elle n'utilise pas la règle *tiers exclu*. Une démonstration dans le système D' est *constructive* si elle ne contient que des séquents de la forme $\Gamma \vdash \Delta$ où Δ est un singleton. Une démonstration en calcul des séquents est *constructive* si elle ne contient que des séquents de la forme $\Gamma \vdash \Delta$ où Δ est un singleton ou le multiensemble vide. En calcul des séquents, on supprime la règle *contraction-droite* et on modifie quelques règles : la règle $\vee$-droite est remplacée par les deux règles

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\text{-droite}$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\text{-droite}$$

la règle $\Rightarrow$-gauche par

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \Rightarrow B \vdash C}\ \Rightarrow\text{-gauche}$$

la règle $\neg$-gauche par

$$\frac{\Gamma \vdash A}{\Gamma, \neg A \vdash B}\ \neg\text{-gauche}$$

et la règle *coupure* par la règle

$$\frac{\Gamma \vdash A \quad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{coupure}$$

On peut démontrer, en s'inspirant des démonstrations du chapitre 6, qu'un séquent $\Gamma \vdash A$ a une démonstration constructive en déduction naturelle si et seulement s'il a une démonstration constructive en calcul des séquents.

De même on peut montrer qu'en calcul des séquents, un séquent $\Gamma \vdash A$ a une démonstration constructive si et seulement s'il a une démonstration constructive et sans coupures.

Proposition 8.1

Si, en calcul des séquents, un séquent $\vdash A$ a une démonstration sans coupures, alors la dernière règle de cette démonstration est une règle droite.

Démonstration. Comme la partie gauche du séquent est vide, cette règle ne peut pas être une règle gauche, ni la règle *axiome*. Comme la démonstration est sans coupures, ce ne peut pas être la règle *coupure*. C'est donc une règle droite.

Proposition 8.2

L'ensemble des propositions qui ont une démonstration constructive a la propriété du témoin.

Démonstration. Si le séquent $\vdash \exists x\ A$ a une démonstration constructive dans le calcul des séquents, il a aussi une démonstration constructive et sans coupures. La dernière règle de cette démonstration est une règle droite, et comme il n'y a pas de règle *contraction-droite* dans le calcul des séquents constructif, c'est la règle $\exists$-droite. La démonstration a donc la forme

$$\frac{\dfrac{\pi}{\vdash (t/x)A}}{\vdash \exists x\ A}\ \exists\text{-droite}$$

et la proposition $(t/x)A$ a une démonstration constructive.

Dans la démonstration ci-avant, le fait que la partie gauche du séquent soit vide est essentiel. Le théorème ne s'étend pas à une théorie quelconque. Par

exemple, l'ensemble des propositions qui ont une démonstration constructive dans la théorie $\exists x\ P(x)$ n'a bien entendu pas la propriété du témoin. Toutefois, on peut montrer que ce théorème s'étend à l'arithmétique et à certaines versions de la théorie des ensembles.

Cette propriété du témoin permet d'utiliser les démonstrations constructives comme des programmes. Par exemple, la proposition

$$\forall x \exists y\ (x = 2 \times y \vee x = 2 \times y + 1)$$

a une démonstration constructive π dans l'arithmétique. À partir de cette démonstration, il n'est pas difficile d'en construire une de la proposition

$$\exists y\ (25 = 2 \times y \vee 25 = 2 \times y + 1)$$

$$\dfrac{\dfrac{\pi}{\Gamma \vdash \forall x \exists y\ A[x,y]} \quad \dfrac{\dfrac{}{\Gamma, \exists y\ A[25,y] \vdash \exists y\ A[25,y]}\ \text{axiome}}{\Gamma, \forall x \exists y\ A[x,y] \vdash \exists y\ A[25,y]}\ \forall\text{-gauche}}{\Gamma \vdash \exists y\ A[25,y]}\ \text{coupure}$$

où A est la proposition $x = 2 \times y \vee x = 2 \times y + 1$ et où on note $A[t,u]$ la proposition $(t/x, u/y)A$. En éliminant les coupures dans cette démonstration, on obtient un témoin : 12.

La démonstration π est donc un programme qui divise son entrée 25 par 2. Le mécanisme d'exécution de ce programme est l'élimination des coupures. Par construction, ce programme est correct vis à vis de la spécification

$$x = 2 \times y \vee x = 2 \times y + 1$$

Exercice 8.2

On associe un type à chaque terme du lambda-calcul. Les types sont les expressions closes d'un langage formé d'un ensemble infini de constantes $\rho_0, \rho_1, \rho_2, \ldots$ et d'un symbole binaire $\rightarrow$. Un *contexte de typage* est un ensemble fini de déclarations de la forme $x : \alpha$ où x est une variable et α un type, tel que si $x : \alpha$ et $x : \beta$ appartiennent tous les deux à l'ensemble, alors $\alpha = \beta$. Un jugement de typage est un triplet formé d'un contexte de typage Γ, d'un terme t et d'un type α. Le jugement $\Gamma \vdash t : \alpha$ exprime que le terme t a le type α dans le contexte Γ, par exemple le terme $\mathit{fun}\ x \rightarrow (f\ x\ x)$ a le type $\rho_0 \rightarrow \rho_0$ dans le contexte $f : \rho_0 \rightarrow \rho_0 \rightarrow \rho_0$. Les jugements *dérivables* sont inductivement définis par les règles suivantes

$$\dfrac{}{\Gamma \vdash x : \alpha}\ \text{si } x : \alpha \text{ est un élément de } \Gamma$$

$$\dfrac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash (\mathit{fun}\ x \rightarrow t) : \alpha \rightarrow \beta}$$

$$\dfrac{\Gamma \vdash t : \alpha \rightarrow \beta \quad \Gamma \vdash u : \alpha}{\Gamma \vdash (t\ u) : \beta}$$

1. Donner un terme de type $\rho_0 \to \rho_1 \to \rho_0$ dans le contexte vide. Et un terme de type $\rho_0 \to \rho_1 \to \rho_1$.
   La logique propositionnelle minimale est le fragment de la logique des prédicats formé des symboles de proposition $P_0, \ldots, P_n$ et de l'implication.
2. Quelles sont les règles de la déduction naturelle que l'on peut utiliser pour démontrer une proposition dans ce fragment ? Donner une démonstration de la proposition $P_0 \Rightarrow P_1 \Rightarrow P_0$. Et de la proposition $P_0 \Rightarrow P_1 \Rightarrow P_1$.
   On considère une application ϕ qui associe un type du lambda-calcul à chaque proposition de la logique propositionnelle minimale.
   $$\phi P_i = \rho_i$$
   $$\phi(A \Rightarrow B) = (\phi A) \to (\phi B)$$
3. Quelle est le type associé à la proposition $P_0 \Rightarrow P_1 \Rightarrow P_0$?
4. Montrer qu'il existe une démonstration π du séquent $A_1, \ldots, A_p \vdash B$ si et seulement s'il existe un terme t de type ϕB dans le contexte $x_1 : \phi A_1, \ldots, x_p : \phi A_p$.
5. Soit Γ le contexte $A, A \Rightarrow B, B \Rightarrow C, C \Rightarrow D$. Quel est le terme associé à la démonstration
   $$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, B \vdash C \Rightarrow D}\,\text{axiome} \quad \dfrac{\dfrac{}{\Gamma, B \vdash B \Rightarrow C}\,\text{axiome} \quad \dfrac{}{\Gamma, B \vdash B}\,\text{axiome}}{\Gamma, B \vdash C}\,\Rightarrow\text{-élim}}{\Gamma, B \vdash D}\,\Rightarrow\text{-élim}}{\Gamma \vdash B \Rightarrow D}\,\Rightarrow\text{-intro} \quad \dfrac{\dfrac{}{\Gamma \vdash A \Rightarrow B}\,\text{axiome} \quad \dfrac{}{\Gamma \vdash A}\,\text{axiome}}{\Gamma \vdash B}\,\Rightarrow\text{-élim}}{\Gamma \vdash D}\,\Rightarrow\text{-élim}$$
   ? Ce terme termine-t-il ? Quelle est sa forme irréductible ? Quelle est la démonstration associée à cette forme irréductible ?
6. Quelle est la forme des démonstrations qui se traduisent sur un radical ? Quelle est la démonstration associée au terme obtenu en réduisant ce radical ?

Exercice 8.3

Cet exercice demande d'avoir fait l'exercice 1.5.

1. Soit A une proposition quelconque, donner une démonstration — non nécessairement constructive —, dans le calcul des séquents, de la proposition
   $$A \vee \neg A$$
   Donner une démonstration constructive de la proposition
   $$\neg\neg(A \vee \neg A)$$
   On associe à chaque proposition A de la logique des prédicats une proposition $|A|$ définie par récurrence sur la structure de A de la manière suivante

- $|P| = \neg\neg P$
- $|\top| = \neg\neg\top$
- $|\bot| = \neg\neg\bot$
- $|A \wedge B| = \neg\neg(|A| \wedge |B|)$
- $|A \vee B| = \neg\neg(|A| \vee |B|)$
- $|A \Rightarrow B| = \neg\neg(|A| \Rightarrow |B|)$
- $|\neg A| = \neg\neg\neg|A|$
- $|\forall x\ A| = \neg\neg\forall x\ |A|$
- $|\exists x\ A| = \neg\neg\exists x\ |A|$

2. Quelle est la proposition $|\exists x\ (P(x) \wedge \neg P(S(x)))|$?
3. On associe à chaque proposition A de la logique des prédicats une proposition $\|A\|$ similaire à $|A|$, sauf que l'on enlève une négation à la racine
   - $\|P\| = \neg P$
   - $\|\top\| = \neg\top$
   - $\|\bot\| = \neg\bot$
   - $\|A \wedge B\| = \neg(|A| \wedge |B|)$
   - $\|A \vee B\| = \neg(|A| \vee |B|)$
   - $\|A \Rightarrow B\| = \neg(|A| \Rightarrow |B|)$
   - $\|\neg A\| = \neg\neg|A|$
   - $\|\forall x\ A\| = \neg\forall x\ |A|$
   - $\|\exists x\ A\| = \neg\exists x\ |A|$

   Montrer que si le séquent $\Gamma \vdash \Delta$ a une démonstration — non nécessairement constructive —, dans le calcul des séquents sans coupures, alors le séquent $|\Gamma|\|\Delta\| \vdash$ a une démonstration sans coupures et constructive.
4. Soit A une proposition. Montrer que si A a une démonstration — non nécessairement constructive —, alors $|A|$ a une démonstration constructive.
5. Montrer que pour toute proposition B, la proposition $B \Leftrightarrow \neg\neg B$ a une démonstration — non nécessairement constructive. Montrer que la proposition $A \Leftrightarrow |A|$ a une démonstration — non nécessairement constructive. Montrer que si la proposition $|A|$ a une démonstration constructive, alors la proposition A a une démonstration — non nécessairement constructive.

   Montrer que la proposition $|A|$ a une démonstration constructive si et seulement si la proposition A a une démonstration — non nécessairement constructive.
6. Donner une démonstration constructive du séquent $|P(0)|, |\neg P(2)| \vdash |\exists x\ (P(x) \wedge \neg P(S(x)))|$.
7. Donner une démonstration constructive du séquent $P(0), \neg P(2) \vdash \neg\neg\exists x\ (P(x) \wedge \neg P(S(x)))$.

Dans ce livre, nous avons exploré un certain nombre de liens entre les notions de démonstration et d'algorithme, à travers le théorème d'indécidabilité de la démontrabilité en logique des prédicats d'abord, puis à travers le résultat de décidabilité de la bonne formation d'une démonstration, qui a mené à un résultat de semi-décidabilité de la démontrabilité en logique des prédicats, à des algorithmes de vérification de démonstrations et de démonstration automatique et à travers des résultats de décidabilité pour des théories particulières. Enfin, la notion de constructivité a mis en évidence un autre lien entre les démonstrations et les algorithmes, qui a mené à une méthode, parmi d'autres, pour démontrer qu'un algorithme vérifie une spécification.

En chemin, nous avons découvert les quatre grandes notions autour desquelles la logique contemporaine est organisée : les notions de démonstration, d'algorithme, de modèle et d'ensemble. Ces quatre notions définissent les quatre branches de la logique : la théorie de la démonstration, la théorie de la calculabilité, la théorie des modèles et la théorie des ensembles. Cette classification, pour utile qu'elle soit, ne doit cependant pas occulter le fait que toutes ces notions sont utilisées, à des degrés divers, dans chacune de ces branches.

Jusqu'à la fin du XIX$^{\text{e}}$ siècle, il n'existait qu'une notion rudimentaire de démonstration, remontant à l'Antiquité, des notions informelles d'ensemble et d'algorithme et pas de notion de modèle. La logique s'est donc entièrement renouvelée quand ces quatre notions ont été dégagées entre les années soixante-dix du XIX$^{\text{e}}$ siècle et les années trente du XX$^{\text{e}}$ siècle.

Nous avons également abordé un certain nombre d'applications de la logique : en mathématiques d'abord, avec des résultats d'indépendance et de cohérence relative, mais aussi, de manière peut-être plus inattendue, avec des résultats d'algèbre, pour lesquels il n'était pas *a priori* évident que des outils logiques soient nécessaires.

Mais c'est surtout en informatique que la logique a trouvé un vaste champ d'applications : en théorie des langages de programmation, avec la conception de langages fonctionnels issus du lambda-calcul et de langages logiques issus

d'algorithmes de démonstration automatique, en architecture où certains circuits sont représentés par des propositions de la logique propositionnelle, en théorie de la complexité, avec, par exemple, la notion de machine de Turing non déterministe, en théorie des bases de données, avec la notion de langage de requête issue de la théorie des modèles finis et en sûreté, avec la conception d'outils permettant de démontrer la correction de circuits et de programmes par rapport à leur spécification logique.

Le rôle central que joue la notion d'algorithme en logique pouvait certes laisser espérer certaines applications en informatique, mais sans doute pas au point que nous connaissons aujourd'hui. La logique semble, par certains aspects, être à l'informatique ce que le calcul différentiel est à la physique.

Et il n'est pas certain que nous ayons aujourd'hui complètement compris les raisons de cette déraisonnable efficacité de la logique en informatique.

Table des matières

Index

Bibliographie

[1] René Cori et Daniel Lascar. *Logique mathématique*. Dunod, 2003.

[2] René David, Karim Nour et Christophe Raffalli. *Introduction à la logique : théorie de la démonstration*. Dunod, 2001.

[3] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and types*. Cambridge University Press, 1989.

[4] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Masson, 1990.

[5] Jean-Louis Krivine. *Théorie des ensembles*. Cassini, 1998.