Logique
M1 MFA - Semestre 2

Franck Benoist

2009-2010, mise à jour 2017

Une mise en perspective historique

Fin du $19^{\text{ème}}$: introduction de nouvelles méthodes à l'encontre de l'intuition de nombreux mathématiciens (induction transfinie de Cantor), apparition de paradoxes (paradoxe de Russell).
Volonté de donner des fondements solides aux mathématiques, en les formalisant.
Programme de Hilbert (tournant du siècle) :

- se donner des règles du jeu pour les démonstations mathématiques (objectif atteint à travers le théorème de complétude de Gödel)
- prouver la cohérence des axiomes mathématiques, en utilisant pour cela uniquement des moyens finitaires (objectif inatteignable, cf le théorème d'incomplétude de Gödel, années 30)

Les mathématiques ne peuvent pas se réduire à un déroulement mécanique des conséquences d'une liste d'axiomes.
Pour certaines branches des mathématiques, on peut toutefois bien comprendre les modèles d'un système d'axiomes intelligible (cf les théories décidables).

Quelques références

Logique mathématique. R. Cori, D. Lascar.
Mathematical Logic. J. Shoenfield.

Chapitre 1

Calcul Propositionnel

1.1 Syntaxe

On se fixe une liste de symboles :
- un ensemble de variables $\mathcal{P}$, dites variables propositionnelles. Par exemple $\mathcal{P} = \{P_1, P_2, \ldots\}$.
- des connecteurs booléens : $\neg$ (non), $\wedge$ (et)
- des parenthèses : (,)

Définition 1.1 *L'ensemble des formules $\mathcal{F} = \mathcal{F}_\mathcal{P}$ est le plus petit ensemble de suites finies de symboles tel que :*
- *tout $P \in \mathcal{P}$ est dans $\mathcal{F}$ (on identifie les éléments de $\mathcal{P}$ avec des suites à un élément)*
- *si $A \in \mathcal{F}$, alors $\neg A \in \mathcal{F}$*
- *si $A, B \in \mathcal{F}$, alors $(A \wedge B) \in \mathcal{F}$*

Remarque 1.2
- *On ne s'occuppe pour l'instant que de syntaxe, on ne s'intéresse pas ici au sens à donner aux connecteurs $\neg$ et $\wedge$*
- *Le choix des connecteurs booléens peut paraitre arbitraire. C'est encore à travers la sémantique qu'on verra qu'il n'est pas nécessaire d'ajouter $\vee, \rightarrow, \ldots$*

Théorème 1.3 (Théorème de lecture unique) *Pour tout $A \in \mathcal{F}$, on est dans une seule des trois situations suivantes :*
- *A est un élément de $\mathcal{P}$*
- *$A = \neg B$ pour un $B \in \mathcal{F}$ uniquement déterminé*
- *$A = (B \wedge C)$ pour $B, C \in \mathcal{F}$ uniquement déterminés*

Preuve On est au plus dans une de ces trois situations, selon que le premier terme de la suite est un élément de $\mathcal{P}$, $\neg$ ou (.
Si $A \notin \mathcal{P}$, on montre que $A = \neg B$ ou $A = (B \wedge C)$, où $B, C \in \mathcal{F}$: si ce n'était pas le cas, $\mathcal{F} \setminus \{A\}$ serait un sous-ensemble strict de $\mathcal{F}$ satisfaisant les conditions de la définition 1.1.
Si $A = \neg B$, B est uniquement déterminé à partir de A (enlever le premier terme de la suite.
Si $A = (B \wedge C) = (B' \wedge C')$, avec $B, C, B', C' \in F$, on montre que $B = B'$: on

peut supposer que B est un segment initial de B' (quitte à échanger B et B'). Si B est un segment initial propre de B', $B \notin F$:

- les éléments de $\mathcal{F}$ ont même nombre de parenthèses ouvrantes que de parenthèses fermantes (induction sur la longueur des formules)
- pour M un segment initial d'une formule, $o(M) \geq f(M)$ (induction sur la longueur des formules)
- si $B' \in \mathcal{P}$, B' n'a pas de segment initial propre autre que vide
- Si $B' = \neg D'$, $B = \neg D$ où D est un segment initial propre de D'
- si B' commence par (, $o(B) > f(B)$ (si $B' = (D' \wedge E')$, soit $B = (D$ pour D segment initial de D', soit $B = (D' \wedge E$ pour E segment initial propre, éventuellement vide, de E'), et donc B n'est pas une formule

On a donc $B = B'$, et donc aussi $C = C'$. □

Remarque 1.4 *Les parenthèses sont indispensables pour avoir ce théorème : pas de lecture unique pour* $\neg P_1 \wedge P_2$.

Ce théorème permet des définitions par induction sur les formules.

Définition 1.5 *On définit la complexité c d'une formule par induction sur les formules :*

- *pour* $P \in \mathcal{P}$, $c(P) = 0$
- *si* $A = \neg B$, $c(A) = c(B) + 1$
- *si* $A = (B \wedge C)$, $c(A) = max(c(B), c(C)) + 1$

On utilisera la notation $F[P_1, \ldots, P_n]$ pour dire que les variables utilisées dans F sont parmi les variables distinctes $P_1, \ldots, P_n$.

Définition 1.6 *Soit* $F[P_1, \ldots, P_n]$ *une formule, et* $A_1, \ldots, A_n$ *des formules. On définit* $F[A_1/P_1, \ldots, A_n/P_n]$ *la formule obtenue à partir de* F *en substituant simultanément chaque* A_i *à* P_i.
Plus précisément, on procède par induction sur les formules :

- *si* F *est une des variables* P_i, $F[A_1/P_1, \ldots, A_n/P_n] = A_i$
- *si* $F = \neg F'$, $F[A_1/P_1, \ldots, A_n/P_n] = \neg F'[A_1/P_1, \ldots, A_n/P_n]$
- *si* $F = (F' \wedge F'')$, $F[A_1/P_1, \ldots, A_n/P_n] = (F'[A_1/P_1, \ldots, A_n/P_n] \wedge F''[A_1/P_1, \ldots, A_n/P_n])$

Cette induction montre clairement que le résultat obtenu est une formule.

Remarque 1.7 *Noter que les substitutions sont simultanées : pour* $F = (P \wedge \neg Q)$, $F[Q/P, P/Q] = (Q \wedge \neg P)$ *est différent de* $F[Q/P, Q/Q][P/P, P/Q] = (P \wedge \neg P)$.

1.2 Sémantique

On se donne deux symboles de vérité $\{0, 1\}$ (ou, si on préfère, $\{F, V\}$, ou tout autre choix de deux symboles distincts). Le sens intuitif des connecteurs booléens correspond à des fonctions $\overline{\neg} : \{0,1\} \longrightarrow \{0,1\}$, avec $\overline{\neg}(0) = 1$ et $\overline{\neg}(1) = 0$, et $\overline{\wedge} : \{0,1\} \times \{0,1\} \longrightarrow \{0,1\}$, avec $\overline{\wedge}(x, y) = 1$ si $x = y = 1$, 0 sinon.

Définition 1.8 *Une évaluation est une fonction* $\phi : \mathcal{P} \longrightarrow \{0,1\}$. *Elle s'étend de manière unique (par induction sur les formules) en* $\overline{\phi} : \mathcal{F} \longrightarrow \{0,1\}$ *telle que :*

- *si* $P \in \mathcal{P}$, $\overline{\phi}(P) = \phi(P)$
- $\overline{\phi}(\neg A) = \overline{\neg}(\overline{\phi}(A)$
- $\overline{\phi}((A \wedge B)) = \overline{\wedge}(\overline{\phi}(A), \overline{\phi}(B))$

Définition 1.9 *On dit qu'une formule F est une tautologie, et on note $\vdash F$, si pour toute évaluation ϕ, $\overline{\phi}(F) = 1$.*
Soit Σ un ensemble de formules, et F une formule. On dit que F est conséquence de Σ, et on note $\Sigma \vdash F$, si pour toute évaluation ϕ qui vérifie $\overline{\phi}(A) = 1$ pour tout $A \in \Sigma$, $\overline{\phi}(F) = 1$.
On dit qu'un ensemble de formules Σ est satisfaisable s'il existe une évaluation ϕ telle que $\overline{\phi}(A) = 1$ pour tout $A \in \Sigma$. Sinon, on dit que Σ est contradictoire.

Remarque 1.10 *Une formule F est une tautologie si et seulement si $\{\neg F\}$ n'est pas satisfaisable.*
Un ensemble fini de formules $\{F_1, \ldots, F_n\}$ est satisfaisable si et seulement si $\{(F_1 \wedge (\ldots \wedge F_n \ldots))\}$ est satisfaisable.
Le fait qu'une formule $F[P_1, \ldots, P_n]$ soit une tautologie est vérifiable par un algorithme : il suffit de calculer $\overline{\phi}(F)$ pour chacune des 2^n évaluations ϕ sur les variables $P_1, \ldots, P_n$. On verra que cette question est plus compliquée pour des systèmes logiques plus riches.

Lemme 1.11 *Soient $F[P_1, \ldots, P_n]$ et $A_1, \ldots, A_n$ des formules, et ϕ une évaluation.*
On définit une évaluation ψ par :
- $\psi(P) = \phi(P)$ *si* $P \in \mathcal{P} \setminus \{P_1, \ldots, P_n\}$
- $\psi(P_i) = \overline{\phi}(A_i)$

Alors $\overline{\phi}(F[A_1/P_1, \ldots, A_n/P_n]) = \overline{\psi}(F)$.

Preuve Par induction sur les formules. □

Corollaire 1.12 *Si $F[P_1, \ldots, P_n]$ est une tautologie et $A_1, \ldots, A_n$ sont des formules, alors $F[A_1/P_1, \ldots, A_n/P_n]$ est une tautologie.*

Preuve Pour chaque évaluation ϕ, on choisit ψ donné par le lemme précédent, et alors $\overline{\phi}(F[A_1/P_1, \ldots, A_n/P_n]) = \overline{\psi}(F) = 1$. □

Proposition et définition 1.13 *On définit la relation $F \sim G$ (F est équivalente à G) sur les formules par : pour toute évaluation ϕ, $\overline{\phi}(F) = \overline{\phi}(G)$.*
Pour Σ_1 et Σ_2 des ensembles de formules, on définit $\Sigma_1 \sim \Sigma_2$ par : pour tout $F_1 \in \Sigma_1$ et $F_2 \in \Sigma_2$, $\Sigma_1 \vdash F_2$ et $\Sigma_2 \vdash F_1$. Ce sont des relations d'équivalence.
On remarque que $\{F\} \sim \{G\}$ si et seulement si $F \sim G$.

On s'attache généralement plus à la sémantique des formules qu'à leur syntaxe, et on s'autorisera souvent à identifier deux formules équivalentes. Par exemple, on identifie $(P_1 \wedge (P_2 \wedge P_3))$ et $((P_1 \wedge P_2) \wedge P_3)$, et on écrit $(P_1 \wedge P_2 \wedge P_3)$ pour cette classe d'équivalence.

On peut voir les autres connecteurs logiques habituels comme des abréviations de formules écrites uniquement avec $\neg$ et $\wedge$:

- $(F \vee G)$ pour $\neg(\neg F \wedge \neg G)$
- $(F \to G)$ pour $(\neg F \vee G)$
- $(F \leftrightarrow G)$ pour $((F \to G) \wedge (G \to F))$

On va montrer en fait un résultat plus fort : toute fonction de $\{0,1\}^n$ dans $\{0,1\}$ est donnée par une fonction de vérité d'une formule écrite avec ces connecteurs (cf Corollaire 1.19).

Proposition 1.14 *$\Sigma \vdash F$ si et seulement si $\Sigma \cup \{\neg F\}$ est contradictoire.*
$\Sigma \cup \{F\} \vdash G$ si et seulement si $\Sigma \vdash (F \to G)$.
$\Sigma \vdash (F \wedge G)$ si et seulement si $\Sigma \vdash F$ et $\Sigma \vdash G$ (attention, c'est faux pour le "ou").
$\vdash (F \leftrightarrow G)$ si et seulement si $F \sim G$.

Preuve Exercice. □

Définition 1.15 *Pour F une formule, on note*

$$\Delta(F) = \{\phi \in \{0,1\}^{\mathcal{P}}; \overline{\phi}(F) = 1\}.$$

Pour $\varepsilon \in \{0,1\}$ et $P \in \mathcal{P}$, on note εP la formule P si $\varepsilon = 1$, $\neg P$ si $\varepsilon = 0$.

Lemme 1.16 *Soient F et G deux formules. Alors $F \sim G$ si et seulement si $\Delta(F) = \Delta(G)$. En particulier, il y a au plus 2^{2^n} classes d'équivalence de formules en les variables $P_1, \ldots, P_n$.*

On va montrer maintenant qu'il y a exactement 2^{2^n} telles classes d'équivalence.

Lemme 1.17 *Soit $\mathcal{P} = \{P_1, \ldots, P_n\}$ un ensemble de variables, et $(\varepsilon_1, \ldots, \varepsilon_n) \in \{0,1\}^n$. On note $\phi_{(\varepsilon_1,\ldots,\varepsilon_n)}$ l'évaluation $P_i \mapsto \varepsilon_i$. Soit la formule*

$$F = \bigwedge_{i=1}^{n} \varepsilon_i P_i.$$

Alors $\Delta(F) = \{\phi_{(\varepsilon_1,\ldots,\varepsilon_n)}\}$.

Preuve Remarquer que $\phi_{(\varepsilon_1,\ldots,\varepsilon_n)}(\varepsilon_i P_i) = 1$. □

Corollaire 1.18 *Soit $X \subseteq \{0,1\}^n$. Il existe une formule $F \in \mathcal{F}_{\mathcal{P}}$ telle que $\Delta(F) = X$.*

Preuve Il suffit de prendre la formule

$$F = \bigvee_{(\varepsilon_1,\ldots,\varepsilon_n) \in X} \bigwedge_{i=1}^{n} \varepsilon_i P_i.$$

□

On donne maintenant deux reformulations de ce résultat.

Corollaire 1.19 *Soit une fonction $\lambda : \{0,1\}^n \longrightarrow \{0,1\}$. Alors il existe une formule $F \in \mathcal{F}_{\mathcal{P}}$ telle que pour toute évaluation ϕ, $\overline{\phi}(F) = \lambda(\phi(P_1), \ldots, \phi(P_n))$.*

Preuve C'est le résultat précédent appliqué à

$$X = \{(\varepsilon_1, \ldots, \varepsilon_n) \in \{0,1\}^n ; \lambda(\varepsilon_1, \ldots, \varepsilon_n) = 1\}.$$

□

On dit alors que $\{\wedge, \neg\}$ est un système complet de connecteurs.

Définition 1.20 *On dit qu'une formule F est sous forme normale disjonctive si elle s'écrit :*

$$F = \bigvee_k \Big(\bigwedge_i \varepsilon_{i,k} P_{i,k}\Big).$$

Corollaire 1.21 *Soit F une formule. Il existe une formule G sous forme normale disjonctive telle que $F \sim G$.*

Preuve On considère l'ensemble $\Delta(F)$ et on constate dans la preuve du corollaire 1.18 qu'on peut trouver une formule G sous forme normale disjonctive telle que $\Delta(G) = \Delta(F)$. □

1.3 Le théorème de compacité

On dit qu'un ensemble de formules Σ est finiment satisfaisable si tout sous-ensemble fini de Σ est satisfaisable. Le but est de montrer le théorème suivant.

Théorème 1.22 (Théorème de compacité) *Soit Σ un ensemble de formules. Alors Σ est satisfaisable si et seulement si Σ est finiment satisfaisable.*

La direction "satisfaisable" implique "finiment satisfaisable" est triviale. Pour la réciproque, on utilise le lemme suivant.

Lemme 1.23 *Soit Σ un ensemble de formules finiment satisfaisable, et F une formule. Alors $\Sigma \cup \{F\}$ ou $\Sigma \cup \{\neg F\}$ est finiment satisfaisable.*

Preuve Dans le cas contraire, on trouve $F_1, \ldots, F_r$ et $G_1, \ldots, G_s$ dans Σ telles que $F_1, \ldots, F_r, F$ et $G_1, \ldots, G_s, \neg F$ ne sont pas satisfaisables. Comme Σ est finiment satisfaisable, il existe une évaluation ϕ telle que qui satisfait $\{F_1, \ldots, F_r, G_1, \ldots, G_s\}$. Mais alors soit $\overline{\phi}(F) = 1$, et $F_1, \ldots, F_r, F$ est satisfaisable, soit $\overline{\phi}(\neg F) = 1$ et $G_1, \ldots, G_s, \neg F$ est satisfaisable : contradiction. □

Corollaire 1.24 *Soit Σ' un ensemble finiment satisfaisable, maximal pour l'inclusion. Alors pour toute formule F, $F \in \Sigma'$ ou $\neg F \in \Sigma'$.*
De plus, $(F \wedge G) \in \Sigma'$ si et seulement si $F \in \Sigma'$ et $G \in \Sigma'$.

Preuve Si $F \notin \Sigma'$, $\Sigma' \subsetneq \Sigma' \cup \{F\}$, donc $\Sigma' \cup \{F\}$ n'est pas finiment satisfaisable (par maximalité). Par le lemme, $\Sigma' \cup \{\neg F\}$ est finiment satisfaisable, ce qui donne par maximalité que $\neg F \in \Sigma'$.
Ensuite, si $(F \wedge G) \in \Sigma'$, et si on suppose par l'absurde que $F \notin \Sigma'$, alors $\neg F \in \Sigma'$. C'est une contradiction puisqu'alors $\{(F \wedge G), \neg F\} \subseteq \Sigma'$ est contradictoire. Réciproquement, on ne peut pas avoir $\{F, G, \neg(F \wedge G)\} \subseteq \Sigma'$ puisque

cet ensemble est contradictoire. □

Preuve (Théorème 1.22) On cherche Σ' finiment satisfaisable maximal contenant Σ.
Démontrons tout d'abord l'existence de Σ' dans le cas où l'ensemble $\mathcal{F}$ des formules est dénombrable, écrivons $\mathcal{F} = \{F_0, F_1, \ldots\}$.
On construit par récurrence :

- $\Sigma_0 = \Sigma$
- $\Sigma_{n+1} = \Sigma_n \cup \{F_n\}$ si cet ensemble est finiment satisfaisable, $\Sigma_{n+1} = \Sigma_n \cup \{\neg F_n\}$ sinon

Soit $\Sigma' = \bigcup_{n \leq 0} \Sigma_n$. On a clairement $\Sigma \subseteq \Sigma'$ et pour tout $F \in \mathcal{F}$, $F \in \Sigma'$ ou $\neg F \in \Sigma'$. De plus, par récurrence et en utilisant le lemme, pour tout n, Σ_n est finiment satisfaisable. On en déduit que Σ' est finiment satisfaisable, puisque tout sous-ensemble fini de Σ' est contenu dans un des Σ_n, et est donc satisfaisable. Et Σ' est maximal parmi les ensembles finiment satisfaisables : si $F \notin \Sigma'$, on sait que $\neg F \in \Sigma'$, donc $\Sigma' \cup \{F\}$ n'est pas finiment satisfaisable puisqu'il contient $\{F, \neg F\}$.

Dans le cas général, on trouve Σ' par le lemme de Zorn. Pour trouver Σ', il suffit de montrer que l'ensemble E des Σ'' finiment satisfaisables contenant Σ est non vide et inductif (l'ordre étant l'inclusion). Il est non vide puisqu'il contient Σ. On montre qu'il est inductif : si E' est est un sous-ensemble totalement ordonné de E, il est majoré dans E. Pour cela, il suffit de considérer le majorant $\Sigma''' = \bigcup_{\Sigma'' \in E'} \Sigma''$; c'est bien un élément de E puisqu'il contient Σ et que tout sous-ensemble fini de Σ''' est contenu dans un des Σ'' de E' (puisque E' est totalement ordonné).
Le lemme de Zorn permet donc de trouver l'élément maximal Σ'.

On montre que Σ' est satisfaisable. Pour cela, on considère l'évaluation ϕ définie par $\phi(P) = 1$ si et seulement si $P \in \Sigma'$.
Alors on montre que pour toute formule F, $\overline{\phi}(F) = 1$ si et seulement si $F \in \Sigma'$, par induction :

- c'est vrai pour les variables d'après la définition de ϕ
- si $F = \neg G$, $\overline{\phi}(F) = 1$ si et seulement si $\overline{\phi}(G) = 0$ si et seulement si $G \notin \Sigma'$ si et seulement si $F \in \Sigma'$ (par le corollaire 1.24)
- si $F = (G \wedge H)$, $\overline{\phi}(F) = 1$ si et seulement si $\overline{\phi}(G) = 1$ et $\overline{\phi}(H) = 1$, si et seulement si $G \in \Sigma'$ et $H \in \Sigma'$ par hypothèse d'induction. C'est équivalent à $(G \wedge H) \in \Sigma'$ par le corollaire 1.24.

L'ensemble Σ' est donc satisfait par ϕ, et donc Σ aussi. □

Chapitre 2

Calcul des prédicats

But : formaliser l'activité habituelle des mathématiques qui est d'étudier des structures. Contrairement au calcul propositionnel, le langage employé dépendra de la structure que l'on veut étudier (par exemple les groupes, les ensembles ordonnés,...). On ne parlera ici que de langage du premier ordre : les quantificateurs ne porteront que sur l'ensemble de base de la structure.

2.1 Syntaxe

Une partie du langage sera commune :
- un ensemble de variables $\mathcal{V}$, par exemple $\mathcal{V} = \{x_0, x_1, \ldots\}$
- les parenthèses (,)
- les connecteurs booléens $\neg$, $\wedge$ (mais comme en calcul propositionnel, on s'autorisera les abréviations $\vee$, $\rightarrow$, $\leftrightarrow$)
- un quantificateur $\forall$, dit universel. On utilisera le quantificateur existentiel $\exists x$ comme abréviation pour $\neg\forall x\neg$
- un symbole de relation binaire pour l'égalité, noté $\equiv$, afin d'éviter la confusion avec l'égalité dans le "méta-langage"

A partir de cette base commune, chaque langage sera donné par :
- une liste de symboles de relation, avec leur arité (il y aura toujours en plus la relation binaire $\equiv$)
- une liste de symboles de fonction, avec leur arité
- une liste de symboles de constante (que l'on peut parfois considérer comme des fonctions d'arité zéro)

On notera par exemple le langage $\mathcal{L} = (\{R_i\}_{i\in I}, \{f_j\}_{j\in J}, \{c_h\}_{h\in H})$, sachant qu'on devra préciser l'arité pour chaque relation et fonction, et que le langage comporte aussi la base commune.

Définition 2.1 (Termes) *L'ensemble $\mathcal{T}$ des termes pour le langage $\mathcal{L}$ est le plus petit ensemble de suites finies de symboles tel que :*
- *si u est un symbole de constante ou une variable, $u \in \mathcal{T}$*
- *si f est un symbole de fonction d'arité n, et si $t_1, \ldots, t_n$ sont des termes, alors $f(t_1)\ldots(t_n) \in \mathcal{T}$*

Remarque 2.2 *On pourrait en fait supprimer l'usage des parenthèses, et montrer un théorème de lecture unique. Avec nos conventions ici, ce théorème de*

lecture unique est à peu près évident. Dans la pratique, on utilisera l'écriture plus courante $f(t_1, \ldots, t_n)$, et $t_1 * t_2$ pour une fonction binaire

Définition 2.3 (Formules) *Les formules atomiques sont les suites finies de symboles $R(t_1)\ldots(t_n)$, où R est un symbole de relation d'arité n (y compris $\equiv$), et $t_1, \ldots, t_n$ des termes.*
L'ensemble des formules $\mathcal{F}$ est le plus ensemble de suites finies de symboles tel que :

- *les formules atomiques sont dans $\mathcal{F}$*
- *si F et G sont dans $\mathcal{F}$, $\neg F$ et $(F \wedge G)$ sont dans $\mathcal{F}$*
- *si F est dans $\mathcal{F}$, et si $x \in \mathcal{V}$, alors $\forall x F$ est dans $\mathcal{F}$*

Comme pour le calcul propositionnel, on a un théorème de lecture unique, que l'on donne sans démonstration.

Théorème 2.4 (Théorème de lecture unique) *Soit F une formule, alors on est dans un seul des cas suivants :*

- *$F = R(t_1)\ldots(t_n)$ est une formule atomique, et $R, t_1, \ldots, t_n$ sont uniquement déterminés*
- *$F = (G \wedge H)$, où G, H sont des formules, uniquement déterminées*
- *$F = \neg G$, où G est une formule, uniquement déterminée*
- *$F = \forall x G$, où x est une variable et G une formule, uniquement déterminées*

Dans une formule F, les variables peuvent intervenir dans le champ d'un quantificateur ou non, ce qui nous amène à distinguer deux types d'occurence des variables.

Définition 2.5 *Soit $x \in \mathcal{V}$. On définit par induction les occurences libres de x dans les formules :*

- *si F est une formule atomique, toute les occurences de x sont libres*
- *si $F = \neg G$, ou $F = (G \wedge H)$, ou $F = \forall y G$ avec y une variable différente de x, les occurences libres de x dans F sont celles de x dans G (et celles de x dans H dans le cas de $\wedge$)*
- *si $F = \forall x G$, x n'a pas d'occurence libre dans F*

Toutes les occurences de x qui ne sont pas libres sont dites liées.
Une formule F sans occurence libre de variables est dite close (on dit aussi que F est un énoncé).
On notera $F[x_1, \ldots, x_n]$ pour exprimer le fait que les seules variables qui ont une occurence libre dans F sont parmi $x_1, \ldots, x_n$.

Exemple 2.6 *Si F est une variable sans quantificateur (c'est-à-dire que $\forall$ n'apparait pas dans F), toutes les occurences de variables dans F sont libres.*

Définition 2.7 *On définit le rang de quantification $RQ(F)$ d'une formule F par induction sur les formules :*

- *si F est une forumule atomique, $RQ(F) = 0$*
- *si $F = \neg G$, $RQ(F) = RQ(G)$*
- *si $F = (G \wedge H)$, $RQ(F) = max(RQ(G), RQ(H))$*
- *si $F = \forall x G$, $RQ(F) = RQ(G) + 1$*

Soit $t[x_1, \ldots, x_n]$ un terme (cette notation signifie que les variables qui y apparaissent sont parmi $x_1, \ldots, x_n$). Pour des termes $t_1, \ldots, t_n$, on définit sans difficulté $t[t_1/x_1, \ldots, t_n/x_n]$ le terme obtenu en substituant simultanément chaque t_i pour x_i (induction sur les termes). Pour des formules, on ne fera les substitutions que sur les occurences libres des variables.

Définition 2.8 *Soit $F[x_1, \ldots, x_n]$ une formule et $t_1, \ldots, t_n$ des termes. Alors on définit $F[t_1/x_1, \ldots, t_n/x_n]$ par induction sur les formules :*
- *si $F = R(u_1) \ldots (u_m)$, $F[t_1/x_1, \ldots, t_n/x_n] = R(u_1[t_1/x_1, \ldots, t_n/x_n]) \ldots (u_m[t_1/x_1, \ldots, t_n/x_n])$*
- *si $F = \neg G$, $F[t_1/x_1, \ldots, t_n/x_n] = \neg G[t_1/x_1, \ldots, t_n/x_n]$*
- *si $F = (G \wedge H)$, $F[t_1/x_1, \ldots, t_n/x_n] = (G[t_1/x_1, \ldots, t_n/x_n] \wedge H[t_1/x_1, \ldots, t_n/x_n])$*
- *si $F = \forall x G$, où x est une variable distincte de $x_1, \ldots, x_n$, $F[t_1/x_1, \ldots, t_n/x_n] = \forall x G[t_1/x_1, \ldots, t_n/x_n, x/x]$*
- *si $F = \forall x_i G$, avec $1 \leq i \leq n$, $F[t_1/x_1, \ldots, t_n/x_n] = \forall x_i G[t_1/x_1, \ldots, x_i/x_i, \ldots, t_n/x_n]$*

Exemple 2.9 *Dans le langage $\mathcal{L} = (\emptyset, \{\cdot\}, \{e\})$, soit la formule $F[x] = \forall y(\forall x x.y \equiv y.x \wedge x.y \equiv y)$. Alors $F[e/x] = \forall y(\forall x x.y \equiv y.x \wedge e.y \equiv y)$.*

2.2 Structures

On se donne $\mathcal{L} = (\{R_i\}_{i \in I}, \{f_j\}_{j \in J}, \{c_h\}_{h \in H})$ un langage du premier ordre. On appelle $\mathcal{L}$-structure la donnée de $\mathcal{M} = (M, (\overline{R}_i)_{i \in I}, (\overline{f}_j)_{j \in J}, (\overline{c}_h)_{h \in H})$ où :
- M est un ensemble non-vide, appelé l'ensemble de base de $\mathcal{M}$
- si R_i est un symbole de fonction d'arité n_i, $\overline{R}_i$ est un sous-ensemble de M^{n_i} (par convention, on écrira aussi $\equiv$ pour la diagonale de M^2)
- si f_j est un symbole de fonction d'arité m_j, $\overline{f}_j$ est une fonction $M^{m_j} \to M$
- $\overline{c}_h$ est un élément de M

Les $\overline{R}_i, \overline{f}_j, \overline{c}_h$ sont les interprétations de R_i, f_j, c_h dans $\mathcal{M}$. Si on considère plusieurs $\mathcal{L}$-structures, on précisera qu'on considère les interprétations dans $\mathcal{M}$ par $\overline{R}_i^{\mathcal{M}}, \overline{f}_j^{\mathcal{M}}, \overline{c}_h^{\mathcal{M}}$.

Soient $\mathcal{M} = (M, (\overline{R}_i^{\mathcal{M}})_{i \in I}, (\overline{f}_j^{\mathcal{M}})_{j \in J}, (\overline{c}_h^{\mathcal{M}})_{h \in H})$ et $\mathcal{N} = (N, (\overline{R}_i^{\mathcal{N}})_{i \in I}, (\overline{f}_j^{\mathcal{N}})_{j \in J}, (\overline{c}_h^{\mathcal{N}})_{h \in H})$ deux $\mathcal{L}$-structure. On dit que $\mathcal{N}$ est une sous-structure de $\mathcal{M}$, ou encore que $\mathcal{M}$ est une extension de $\mathcal{N}$, et on note $\mathcal{N} \subseteq_{\mathcal{L}} \mathcal{M}$ si :
- $N \subseteq M$
- pour tout $i \in I$, $\overline{R}_i^{\mathcal{N}} = \overline{R}_i^{\mathcal{M}} \cap N^{n_i}$
- pour tout $j \in J$, $\overline{f}_j^{\mathcal{N}} = \overline{f}_{j \restriction N^{m_j}}^{\mathcal{M}}$
- pour tout $h \in H$, $\overline{c}_h^{\mathcal{N}} = \overline{c}_h^{\mathcal{M}}$

Remarque 2.10 *Ici, l'ensemble de base change mais le langage reste le même. On utilisera parfois une autre notion, à ne pas confondre, où le langage change mais l'ensemble de base reste le même : soient $\mathcal{L} \subseteq \mathcal{L}'$ deux langages et $\mathcal{M}' = (M, \overline{H})_{H \in \mathcal{L}'}$ une $\mathcal{L}'$-structure. La $\mathcal{L}$-structure $\mathcal{M} = (M, \overline{H})_{H \in \mathcal{L}}$ s'appelle la réduite de $\mathcal{M}'$ de $\mathcal{L}'$ à $\mathcal{L}$, et on dit que $\mathcal{M}'$ est un enrichissement de $\mathcal{M}$ de $\mathcal{L}$ à $\mathcal{L}'$.*

Un homomorphisme (de $\mathcal{L}$-structures) de $\mathcal{M}$ dans $\mathcal{N}$ est une application $\phi : M \to N$ telle que :

- pour tout $i \in I$, et $a_1, \ldots, a_{n_i} \in M$, si $(a_1, \ldots, a_{n_i}) \in \overline{R}_i^{\mathcal{M}}$, alors $(\phi(a_1), \ldots, \phi(a_{n_i})) \in \overline{R}_i^{\mathcal{N}}$
- pour tout $j \in J$, et $a_1, \ldots, a_{m_j} \in M$, $\phi(\overline{f}_j^{\mathcal{M}}(a_1, \ldots, a_{m_j})) = \overline{f}_j^{\mathcal{N}}(\phi(a_1), \ldots, \phi(a_{m_i}))$
- pour tout $h \in H$, $\phi(\overline{c}_h^{\mathcal{M}}) = \overline{c}_h^{\mathcal{N}}$

Exemple 2.11 *Soit le langage $\mathcal{L} = (\emptyset, \{\cdot\}, \{e\})$, et $\mathcal{G} = (G, +, 0)$, $\mathcal{H} = (H, \times, 1)$ deux $\mathcal{L}$-structures qui sont des groupes. Alors un homomorphisme de $\mathcal{L}$-structures de $\mathcal{G}$ dans $\mathcal{H}$ est simplement un homomorphisme de groupes.*

Un homomorphisme ϕ est un monomorphisme si de plus il vérifie la condition : pour toute relation R d'arité n (une relation R_i ou $\equiv$), et $a_1, \ldots, a_n \in M$, $(a_1, \ldots, a_n) \in \overline{R}^{\mathcal{M}}$ si et seulement si $(\phi(a_1), \ldots, \phi(a_n)) \in \overline{R}^{\mathcal{N}}$.

En considérat la relation $\equiv$, on constate en particulier qu'un monomorphisme est injectif.
Un homomorphisme ϕ est un isomorphisme si et seulement s'il admet un homomorphisme réciproque.

Remarque 2.12 *Pour qu'un homomorphisme soit un isomorphisme, il faut et il suffit que ce soit un monomorphisme surjectif.*

2.3 Sémantique

On se fixe un langage $\mathcal{L}$ et une $\mathcal{L}$-structure $\mathcal{M}$.
Soit $t[x_1, \ldots, x_n]$ un terme. Par composition des fonctions $\overline{f}_j$, il est clair qu'on peut associer à t une fonction $\overline{t} : M^n \to M$ (c'est encore une fois une induction sur les termes).

Définition 2.13 (Satisfaction d'une formule) *Soit $F[x_1, \ldots, x_n]$ une formule et $a_1, \ldots, a_n$ des éléments de M. On définit "$\mathcal{M}$ satisfait F en $(a_1, \ldots, a_n)$", et on note $\mathcal{M} \models F[a_1, \ldots, a_n]$, par induction sur les formules :*

- *si $F = R(t_1[x_1, \ldots, x_n]) \ldots (t_m[x_1, \ldots, x_n])$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $(\overline{t_1}[a_1, \ldots, a_n], \ldots, \overline{t_m}[a_1, \ldots, a_n]) \in \overline{R}$*
- *si $F = \neg G$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{M} \not\models G[a_1, \ldots, a_n]$*
- *si $F = (G \wedge H)$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{M} \models G[a_1, \ldots, a_n]$ et $\mathcal{M} \models H[a_1, \ldots, a_n]$*
- *si $F = \forall y G[x_1, \ldots, x_n, y]$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si pour tout $b \in M$, $\mathcal{M} \models G[a_1, \ldots, a_n, b]$*

Cette définition par induction nécessite le passage par des formules avec variables libres, toutefois, on s'intéressera surtout aux énoncés. On utilisera dans la suite les mêmes abréviations qu'en calcul propositionnel : $\vee, \rightarrow, \leftrightarrow$, ainsi que $\exists x$ pour $\neg\forall x\neg$.

Définition 2.14 *On appelle théorie (dans le langage $\mathcal{L}$) un ensemble d'énoncés.*
Soit T une théorie. On dit que $\mathcal{M}$ est modèle de T, et on note $\mathcal{M} \models T$, si $\mathcal{M} \models F$ pour tout $F \in T$.
On dit que T a pour conséquence (sémantique) un énoncé F, et on note $T \vdash F$,

si pour tout modèle $\mathcal{M}$ de T, $\mathcal{M} \models F$.
On dit que $F[x_1, \ldots, x_n]$ est une tautologie si $\emptyset \vdash \forall x_1 \ldots \forall x_n F[x_1, \ldots, x_n]$.
On dit que les formules $F[x_1, \ldots, x_n]$ et $G[x_1, \ldots, x_n]$ sont équivalentes modulo T (ou simplement équivalentes quand $T = \emptyset$) si $T \vdash \forall x_1 \ldots \forall x_n (F[x_1, \ldots, x_n] \leftrightarrow G[x_1, \ldots, x_n])$.
Une théorie est dite consistante si elle admet au moins un modèle.
On appelle théorie de $\mathcal{M}$ la théorie $Th(\mathcal{M}) := \{F \text{ énoncé} ; \mathcal{M} \models F\}$. En particulier, $\mathcal{M} \models T$ si et seulement si $T \subseteq Th(\mathcal{M})$.
Deux $\mathcal{L}$-structures $\mathcal{M}$ et $\mathcal{N}$ sont dites élémentairement équivalentes (on note $\mathcal{M} \equiv \mathcal{N}$) si elles satisfont les mêmes énoncés, c'est-à-dire si $Th(\mathcal{M}) = Th(\mathcal{N})$.

Remarque 2.15 *Si T est une théorie inconsistante, alors tout énoncé est conséquence de T.*
Soit T une théorie et T' l'ensemble des conséquences de T. Alors T et T' ont exactement les mêmes modèles. Dans la pratique, on se permettra donc souvent de passer de T à T', sauf quand il sera important de fixer une axiomatisation particulière de la théorie.
Si deux $\mathcal{L}$-structures sont isomorphes, elles sont élémentairement équivalentes (on montre en effet par une induction facile sur les formules que si $\phi : \mathcal{M} \to \mathcal{N}$ est un isomorphisme, $F[x_1, \ldots, x_n]$ une formule et $a_1, \ldots, a_n$ des éléments de M, alors $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{N} \models F[\phi(a_1) \ldots, \phi(a_n)]$). La réciproque n'est pas vraie en générale.

Proposition et définition 2.16 *Soit T une théorie consistante. Les propriétés suivantes sont équivalentes :*

1. *T est maximale parmi les théories consistantes*
2. *pour tout énoncé F, $F \in T$ ou $\neg F \in T$*
3. *il existe une $\mathcal{L}$-structure M telle que $T = Th(\mathcal{M})$*
4. *pour tous $\mathcal{M}, \mathcal{N} \models T$, $\mathcal{M} \equiv \mathcal{N}$, et T contient l'ensemble de ses conséquences*

Quand ces propriétés sont vérifiées pour $T' := \{F \text{ énoncé}; T \vdash F\}$, on dit que la théorie T est complète.

Preuve $1 \Rightarrow 2$ Soit F un énoncé. Si $F \notin T$, par maximalité de T, $T \cup \{F\}$ n'est pas consistante. Soit $\mathcal{M}$ un modèle de T, $\mathcal{M}$ ne peut pas satisfaire F, donc $\mathcal{M} \models \neg F$. Ainsi, $T \cup \{\neg F\}$ est consistante, et par maximalité, $\neg F \in T$.
$2 \Rightarrow 3$ T est consistante donc on peut considérer un modèle $\mathcal{M}$ de T. En particulier, $T \subseteq \mathrm{Th}(\mathcal{M})$. Pour l'inclusion inverse, soit $F \in \mathrm{Th}(\mathcal{M})$. Si $\neg F \in T \subseteq \mathrm{Th}(\mathcal{M})$, on a à la fois $\mathcal{M} \models F$ et $\mathcal{M} \models \neg F$, ce qui est impossible. Donc $F \in T$.
$3 \Rightarrow 4$ On choisit $\mathcal{M}$ tel que $T = \mathrm{Th}(\mathcal{M})$. Soit $\mathcal{N}$ un modèle de T. Alors $\mathrm{Th}(\mathcal{M}) = T \subseteq \mathrm{Th}(\mathcal{N})$. Or pour $F \in \mathrm{Th}(\mathcal{N})$, on sait que $\mathcal{M} \models F$ ou $\mathcal{M} \models \neg F$ (par définition de la satisfaction), or $\mathcal{M} \models \neg F$ est impossible car on aurait alors $\neg F \in \mathrm{Th}(\mathcal{N})$. Donc $\mathcal{M} \equiv \mathcal{N}$ (et le cas général s'obtient par transitivité). De plus, $T = \mathrm{Th}(\mathcal{M})$ contient toutes ses conséquences : si $T \vdash F$, alors en particulier $F \in \mathrm{Th}(\mathcal{M})$.
$4 \Rightarrow 1$ Soit F un énoncé qui n'est pas dans T, on sait par hypothèse qu'il n'est pas conséquence de T. Il existe donc un modèle $\mathcal{M}$ de T tel que $\mathcal{M} \models \neg F$. Alors, pour tout modèle $\mathcal{N}$ de T, on a aussi $\mathcal{N} \models \neg F$. Ainsi, la théorie $T \cup \{F\}$

n'est pas consistante. □

Corollaire 2.17 *Soit T une théorie consistante. Les propriétés suivantes sont équivalentes :*

1. *T est complète*
2. *pour tout énoncé F, $T \vdash F$ ou $T \vdash \neg F$*
3. *pour tous modèles $\mathcal{M}$ et $\mathcal{N}$ de T, $\mathcal{M} \equiv \mathcal{N}$*

Exemple 2.18 *La théorie T des groupes dans le langage naturel $\mathcal{L} = (\emptyset, \{\cdot\}, \{e\})$ n'est pas complète : certains modèles de T satisfont l'énoncé $\forall x \forall y\ x \cdot y \equiv y \cdot x$, et d'autres non. La théorie $Th(\mathbb{Z}, +, 0)$ est une théorie complète contenant T, on dit que c'est une complétion de T.*
Il faut en général plus de travail pour montrer qu'une théorie donnée par une liste explicite d'axiomes est complète ; exemple : la théorie des ordres totaux denses sans extrémités.

2.4 Isomorphismes locaux et va-et-vient

On suppose dans cette section que le langage ne contient que des symboles de relation. Ce n'est pas une vraie contrainte dans l'expressivité du langage : on pourra remplacer un symbole de fonction d'arité m par son graphe, qui est une relation d'arité $m + 1$; de même, un symbole de constante c (qu'on peut voir comme une fonction d'arité 0) pourra être remplacé par une relation unaire R vérifiant $R(x)$ ssi $x = c$. Le principal intérêt de cette hypothèse sur le langage est que pour toute $\mathcal{L}$-structure $\mathcal{M}$ et tout sous-ensemble N de $\mathcal{M}$, on peut voir N comme une sous-$\mathcal{L}$-structure de $\mathcal{M}$ en le munissant des restrictions des relations interprétées dans $\mathcal{M}$. Exceptionnellement pour cette section, on s'autorisera à regarder des sous-structures dont l'ensemble de base est vide.

Définition 2.19 *Soient $\mathcal{M}$ et $\mathcal{N}$ deux $\mathcal{L}$-structures. Un isomorphisme local de $\mathcal{M}$ dans $\mathcal{N}$ est un $\mathcal{L}$-isomorphisme $\mathcal{M}_0 \to \mathcal{N}_0$ pour $\mathcal{M}_0$ et $\mathcal{N}_0$ des sous-$\mathcal{L}$-structures finies (éventuellement vides) de $\mathcal{M}$ et $\mathcal{N}$ respectivement.*
On définit la notion de p-isomorphisme de $\mathcal{M}$ dans $\mathcal{N}$ par récurrence sur $p \in \mathbb{N}$:

- *les 0-isomorphismes sont exactement les isomorphismes locaux*
- *un $(p+1)$-isomorphisme est un isomorphisme local ϕ de $\mathcal{M}$ dans $\mathcal{N}$ tel que*
  - *pour tout $a \in M$, il existe $b \in N$ tel que la fonction $\tilde{\phi}$, obtenue en prolongeant ϕ par $\tilde{\phi}(a) = b$, est un p-isomorphisme (condition de va)*
  - *pour tout $b \in N$, il existe $b \in M$ tel que la fonction $\tilde{\phi}$, obtenue en prolongeant ϕ par $\tilde{\phi}(a) = b$, est un p-isomorphisme (condition de vient)*

Un isomorphisme local est un ω-isomorphisme si c'est un p-isomorphisme pour tout p.

Les propriétés suivantes sont à peu près évidentes :

Fait 2.20 *1. la restriction d'un p-isomorphisme est encore un p-isomorphisme, pour $p \in \mathbb{N} \cup \{\omega\}$ (récurrence sur p)*

2. *en particulier, s'il existe un p-isomorphisme entre $\mathcal{M}$ et $\mathcal{N}$, l'application vide en est un aussi*
3. *les $(p+1)$-isomorphismes sont des p-isomorphismes (récurrence sur p)*
4. *si les p-isomorphismes sont des $(p+1)$-isomorphismes, alors ce sont aussi des q-isomorphismes pour tout $q > p$ (récurrence sur q) ; en particulier, ce sont des ω-isomorphismes*

Exemple 2.21 *On considère le langage vide, dont le seul symbole de relation est l'égalité. Les isomorphismes (locaux) sont donc seulement des bijections.*
Si M et N sont deux ensembles infinis, tous les isomorphismes loacaux entre M et N sont des 1-isomorphismes, et donc des ω-isomorphismes.
Si M est un ensemble infini et N un ensemble de cardinal fini m, les p-isomorphismes entre M et N sont les isomorphismes locaux dont le domaine a au plus $m-p$ éléments. En particulier, il n'existe pas de $(m+1)$-isomorphisme entre M et N.

Théorème 2.22 (de Fraïssé) *Soient $\mathcal{M}$ et $\mathcal{N}$ deux $\mathcal{L}$-structures, et ϕ un p-isomorphisme de $\mathcal{M}$ dans $\mathcal{N}$, de domaine $\{a_1, \ldots, a_n\}$. Alors pour toute formule $F[x_1, \ldots, x_n]$ de rang de quantification $\leq p$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{N} \models F[\phi(a_1), \ldots, \phi(a_n)]$.*

Preuve On raisonne par récurrence sur p.
Pour $p = 0$, un isomorphisme local préserve par définition les formules atomiques ($\mathcal{M} \models R[a_1, \ldots, a_n]$ si et seulement si $\mathcal{N} \models R[\phi(a_1), \ldots, \phi(a_n)]$), la transmission à toutes les formules sans quantificateur se fait par induction sur les formules.
On suppose que l'énoncé est satisfait pour p. Si ϕ est un $(p+1)$-isomorphisme, c'est a fortiori un p-isomorphisme, donc il préserve la satisfaction des formules de rang de quantification $\leq p$ par hypothèse de récurrence. Il n'y a donc que le cas des formules de rang de quantification $p+1$ à traiter. On raisonne par induction sur les formules. Si $F = G \to H$, avec G et H de rang de quantification $\leq p+1$, la propriété pour G et H donne la propriété pour F. Idem si $F = \neg G$. Si $F[x_1, \ldots, x_n] = \forall y G[x_1, \ldots, x_n, y]$, avec G de rang de quantification p, supposons que $\mathcal{M} \not\models F[a_1, \ldots, a_n]$, alors il existe a dans M tel que $\mathcal{M} \not\models G[a_1, \ldots, a_n, a]$. Comme ϕ est un $(p+1)$-isomorphisme, on peut le prolonger en un p-isomorphisme $\tilde{\phi}$ dont le domaine contient a. Alors par hypothèse de récurrence, $\mathcal{N} \not\models G[\tilde{\phi}(a_1), \ldots, \tilde{\phi}(a_n), \tilde{\phi}(a)]$, et donc $\mathcal{N} \not\models F[\phi(a_1), \ldots, \phi(a_n)]$.
La réciproque se montre de la même façon, en utilisant cette fois la propriété de vient. □

Corollaire 2.23 *S'il existe un ω-isomorphisme entre $\mathcal{M}$ et $\mathcal{N}$, alors $\mathcal{M} \equiv \mathcal{N}$.*

Exemple 2.24 *Soit T la théorie dont les modèles sont les ensembles totalement ordonnés, denses, sans extrémité. Alors tous les isomorphismes locaux entre modèles de T sont des ω-isomorphismes, donc la théorie T est complète.*

2.5 Démonstrations et complétude

Pour montrer qu'un énoncé F est conséquence d'une théorie T, il faut a priori vérifier la satisfaction de F dans tous les modèles de T, ce qui est impossible en pratique. L'activité mathématique consiste en (grande) partie à dériver

des théorèmes à partir d'axiomes, c'est ce qu'on veut formaliser maintenant.

On fixe un langage $\mathcal{L}$, et T une théorie dans ce langage. Une démonstration d'un énoncé F à partir de T est une suite de formules $(F_0, F_1, \ldots, F_n)$, telle que $F_n = F$ et que pour tout $0 \leq i \leq n$, F_i est obtenue à partir d'une des es règles suivantes :

1. F_i est un élément de T (axiome de T)
2. F_i est l'une des formules suivantes (axiomes pour l'égalité)
   (a) $x_1 \equiv x_1, (x_1 \equiv x_2 \to x_2 \equiv x_1), (x_1 \equiv x_2 \to ((x_2 \equiv x_3) \to (x_1 \equiv x_3)))$
   (b) $(y \equiv z \to t[y/x_1, x_2, \ldots, x_r] \equiv t[z/x_1, x_2, \ldots, x_r])$, où $t[x_1, \ldots, x_r]$ est un terme
   (c) $(y \equiv z \to (A[y/x_1, x_2, \ldots, x_r] \leftrightarrow A[z/x_1, x_2, \ldots, x_r]))$, où $A[x_1, \ldots, x_r]$ est une formule
3. il existe des formules A, B, C telles que F_i s'écrive sous une des formes suivantes (tautologie propositionnelle) :
   (a) $(A \to (B \to A))$
   (b) $((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$
   (c) $((\neg A \to \neg B) \to (B \to A))$
4. il existe $j, k < i$ tels que F_k soit de la forme $(F_j \to F_i)$ (modus ponens)
5. il existe $j < i$ et x une variable tels que $F_i = \forall x F_j$ (généralisation)
6. il existe des formules A, B telles que F_i s'écrive sous une des formes suivantes (axiomes pour $\forall$) :
   (a) $(\forall x A \to A[t/x])$, où t est un terme et aucune occurence libre de x dans A ne se trouve dans le champ d'un quantificateur liant une variable de t
   (b) $(\forall x(A \to B) \to (A \to \forall x B))$, où x n'a aucune occurence libre dans A

Quand une telle démonstration existe, on dit que F est démontrable à partir de T et on note (provisoirement) $T \vdash_d F$.

Remarque 2.25 *Les règles et axiomes ci-dessus sont écrits avec le connecteur $\to$ pour une meilleure compréhension, il faudrait en fait les remplacer par les formules adaptées avec les connecteurs habituels : $(F \to G)$ à remplacer par $\neg(F \wedge \neg G)$, et aussi par $\neg(F \wedge H)$ dans le cas où $G = \neg H$.*

Définition 2.26 *On dit qu'une théorie T est cohérente si T ne démontre pas tous les énoncés.*

Lemme 2.27 *Une théorie T est incohérente si et seulement s'il existe un énoncé F telle que $T \vdash_d F$ et $T \vdash_d \neg F$.*

Preuve Une direction est facile : si T est incohérente, elle démontre tous les énoncés, a fortiori les énoncés F et $\neg F$ (F étant quelconque).
Réciproquement, on suppose que $T \vdash_d F$ et $T \vdash_d \neg F$. Soit G une énoncé. Il suffit alors de faire suivre les démonstrations de F et de $\neg F$ des formules suivantes : $(\neg F \to (\neg G \to \neg F))$ (tautologie (a)), $(\neg G \to \neg F)$ (modus ponens),

$((\neg G \to \neg F) \to (F \to G))$ (tautologie (c)), $(F \to G)$ (modus ponens), G (modus ponens), et on obtient ainsi $T \vdash_d G$. □

Lemme 2.28 (Lemme de finitude) *Soit T une théorie et F un énoncé. Alors $T \vdash_d F$ si et seulement s'il existe un sous-ensemble fini T_0 de T tel que $T_0 \vdash_d F$.*

Preuve Si $T_0 \vdash_d F$, avec $T_0 \subseteq T$, a fortiori, $T \vdash_d F$.
Réciproquement, si $(F_0, \ldots, F_n)$ est une démonstration de F à partir de T, on prend pour T_0 le nombre (fini) de ces formules qui apparaissent dans la démonstration au titre d'axiomes de T ; on a bien encore $T_0 \vdash_d F$. □

Lemme 2.29 (Lemme de déduction) *Soit T une théorie, F et G deux énoncés. Alors $T \cup \{F\} \vdash_d G$ si et seulement si $T \vdash_d (F \to G)$.*

Preuve "$\Leftarrow$" : si $(F_0, \ldots, (F \to G))$ est une preuve de $(F \to G)$ à partir de T, alors $(F_0, \ldots, (F \to G), F, G)$ est une preuve de G à partir de $T \cup \{F\}$ par modus ponens.
"$\Rightarrow$" : on se donne $(F_0, \ldots, F_n)$ une démonstration de $G = F_n$ à partir de $T \cup \{F\}$. On va insérer parmi la suite de formules $(F \to F_0), \ldots, (F \to F_n)$ des formules afin d'obtenir une démonstartion à partir de T :

- si F_i est un axiome de T, un axiome de l'égalité, un axiome pour $\forall$ ou une tautologie propositionnelle, on insère avant $(F \to F_i)$ les formules : F_i (axiome ou tautologie), $(F_i \to (F \to F_i))$ (tautologie (a)), et $(F \to F_i)$ est obtenu par modus ponens
- si $F_i = F$, on insère avant la formule $(F \to F)$ les formules : $(F \to ((F \to F) \to F))$ (tautologie (a)), $((F \to ((F \to F) \to F)) \to ((F \to (F \to F)) \to (F \to F)))$ (tautologie (b)), $((F \to (F \to F)) \to (F \to F))$ (modus ponens), $(F \to (F \to F))$ (tautologie (a)), $(F \to F)$ est obtenu par modus ponens
- si F_i est obtenu par modus ponens à partir de $F_k = (F_j \to F_i)$ pour $j, k < i$, alors, $(F \to F_k)$ et $(F \to F_j)$ apparaissent déjà, et on insère juste avant la formule $(F \to F_i)$ les formules : $((F \to (F_j \to F_i)) \to ((F \to F_j) \to (F \to F_i)))$ (tautologie (b)), $((F \to F_j) \to (F \to F_i))$ (modus ponens), $(F \to F_i)$ est obtenu par modus ponens
- si F_i est obtenu par généralisation, $F_i = \forall x F_j$ avec $j < i$, alors $(F \to F_j)$ apparait déjà, et on insère juste avant $(F \to F_i)$ les formules : $\forall x(F \to F_j)$ (généralisation), $(\forall x(F \to F_j) \to (F \to \forall x F_j))$ (axiome pour $\forall$ (b), puisque x n'a pas d'occurence libre dans F, qui est un énoncé), $(F \to \forall x F_j)$ est obtenu par modus ponens

Ainsi, $T \vdash_d (F \to F_n)$. □

Corollaire 2.30 *Soit T une théorie et F un énoncé. Alors $T \vdash_d F$ si et seulement si $T \cup \{\neg F\}$ est incohérente.*

Preuve Si $T \vdash_d F$, alors a fortiori $T \cup \{\neg F\} \vdash_d F$ et $T \cup \{\neg F\} \vdash_d \neg F$, donc $T \cup \{\neg F\}$ est incohérente.
Réciproquement, si $T \cup \{\neg F\}$ est incohérente, alors en particulier, $T \cup \{\neg F\} \vdash_d \neg(F \to (F \to F))$, et donc $T \vdash_d (\neg F \to \neg(F \to (F \to F))$ par le lemme

de déduction. On en déduit, par la tautologie (c) et le modus ponens, que $T \vdash_d ((F \to (F \to F)) \to F)$, puis, par la tautologie (a) et le modus ponens, $T \vdash_d F$. □

Le choix des règles pour les démonstrations peut paraître arbitraire (pourquoi par exemple ne pas ajouter d'autres tautologies propositionnelles ?) ; mais le point important est que ce système de démonstration est complet, dans le sens suivant.

Théorème 2.31 (Théorème de complétude de Gödel) *Soit T une théorie et F un énoncé.*
Alors $T \vdash F$ si et seulement si $T \vdash_d F$.
De manière équivalente, une théorie T est consistante si et seulement si elle est cohérente.

Vérifions tout d'abord que ces deux formulations sont bien équivalentes. Si la première formulation est vraie, alors T est incohérente si et seulement s'il existe un énoncé F tel que $T \vdash_d F$ et $T \vdash_d \neg F$, ce qui équivaut à $T \vdash F$ et $T \vdash \neg F$, qui est vrai si et seulement si T est inconsistante. Si la deuxième formulation est vraie, et si F est un énoncé, alors $T \vdash_d F$ si et seulement si $T \cup \{\neg F\}$ est incohérent, ou encore inconsistant, c'est-à-dire si aucun des modèles de T ne satisfait $\neg F$, ce qui équivaut à $T \vdash F$.
Le reste du paragraphe va être dédié à la démonstration de ce théorème. La partie difficile est de construire, *ex nihilo*, un modèle d'une théorie cohérente T. Pour cela, nous introduisons les outils suivants.

Définition 2.32 *On dit qu'une théorie T admet des témoins de Henkin si pour toute formule $F[x]$ à (au plus) une variable libre, il existe une constante c_F du langage telle que $T \vdash_d (\exists x F \to F[c_F/x])$.*

Lemme 2.33 *Soit T une théorie cohérente. Alors il existe un langage $\mathcal{L}' \supseteq \mathcal{L}$ et une théorie cohérente T' dans le langage $\mathcal{L}'$ telle que $T \subseteq T'$ et T' admet des témoins de Henkin.*

Preuve Partant de $\mathcal{L}_0 = \mathcal{L}$ et $T_0 = T$, on va construire $\mathcal{L}_{i+1} \supseteq \mathcal{L}_i$ et $T_{i+1} \supseteq T_i$ une théorie cohérente dans le langage $\mathcal{L}_{i+1}$ telle que pour toute formule $F[x]$ du langage $\mathcal{L}_i$, il existe une constante c_F du langage $\mathcal{L}_{i+1}$ telle que $T_{i+1} \vdash_d (\exists x F \to F[c_F/x])$.
Soit donc $\mathcal{L}_{i+1} := \mathcal{L}_i \dot{\cup} \{c_F; F[x] \text{ formule de } \mathcal{L}_i\}$ et $T_{i+1} = T_i \cup \{(\exists x F \to F[c_F/x]); F[x] \text{ formule de } \mathcal{L}_i\}$. Il faut vérifier que T_{i+1} est cohérente. Par le lemme de finitude, il suffit de vérifier que T_i reste cohérente quand on lui ajoute un nombre fini de formules de la forme $(\exists x F \to F[c_F/x])$. Par récurrence, il suffit de vérifier que, si T'' est cohérente et ne fait pas intervenir la constante c, alors $T'' \cup \{\exists x F \to F[c/x])\}$ est cohérente.
Remarquons tout d'abord qu'on ne peut pas avoir à la fois $T'' \vdash_d \neg F[c/x]$ et $T'' \vdash_d \exists x F$. Supposons en effet que l'on a une démonstration $(F_0, \ldots, F_n)$ de $\neg F[c/x]$ à partir de T''. Soit y une variable qui n'apparait pas dans cette démonstration. Alors on a un début de démontration à partir de T'', $(G_0, \ldots, G_n)$, où G_i est obtenu à partir de F_i en remplaçant le symbole de constante c par la variable y : si F_i est un axiome de T, $G_i = F_i \in T$ puisque c n'apparaît

pas dans T, si F_i est un autre axiome, alors G_i l'est aussi, et si F_i est obtenu par les modus ponens ou généralisation à partir de formules précédentes, G_i l'est à partir des formules correspondantes. Puis, on termine la démonstration par : $G_n = \neg F[y/x]$, $\forall y \neg F[y/x]$ (généralisation), $(\forall y \neg F[y/x] \rightarrow \neg F[y/x][x/y])$ (par l'axiome (a) pour $\forall$, sachant que les occurences libres de y dans $F[y/x]$ remplacent les occurences libres de x dans F, et qu'elles ne se trouvent donc pas dans le champ d'un quantificateur liant x ; remarquons d'autre part que $F[y/x][x/y] = F$ puisque y n'apparaît pas dans F), $\neg F$ par modus ponens, et enfin $\forall x \neg F$ par généralisation. Comme T est cohérente, et que $\exists x F$ est l'abréviation pour $\neg \forall x \neg F$, on ne peut pas avoir $T \vdash_d \exists x F$.
En utilisant le Corollaire 2.30, on a donc : ou bien $T'' \cup \{\neg \exists x F\}$ est cohérente, ou bien $T'' \cup \{\neg \neg F[c/x]\}$ est cohérente. Dans le premier cas, on démontre à partir de $T'' \cup \{\neg \exists x F\}$: $(\neg F[c/x] \rightarrow \neg \exists x F)$ (par la tautologie (a) et le modus ponens, puis $(\exists x F \rightarrow F[c/x])$ par la tautologie (c) et le modus ponens. Donc $T'' \cup \{(\exists x F \rightarrow F[c/x])\}$ est cohérente : si on pouvait démontrer tous les énoncés à partir de cette théorie, on le pourrait aussi à partir de $T'' \cup \{\neg \exists x F\}$. Dans le deuxième cas, on démontre à partir de $T'' \cup \{\neg \neg F[c/x]\} : F[c/x]$ (par le corollaire 2.30, puisque $T'' \cup \{\neg \neg F[c/x]\} \cup \{\neg F[c/x]\}$ est incohérent), puis $(\exists x F \rightarrow F[c/x])$ par la tautologie (a) et le modus ponens. Là encore, cela donne que $T'' \cup \{(\exists x F \rightarrow F[c/x])\}$ est cohérente.
On pose maintenant $\mathcal{L}' = \bigcup_{i \in \mathbb{N}} \mathcal{L}_i$ et $T' = \bigcup_{i \in \mathbb{N}} T_i$. Alors T' est cohérente (par le lemme de finitude) et admet des témoins de Henkin (car toute formule de $\mathcal{L}'$ est une formule d'un certain $\mathcal{L}_i$). □

Lemme 2.34 *Soit T une théorie cohérente admettant des témoins de Henkin. Alors T est consistante.*

Preuve On montre tout d'abord que T est incluse dans une théorie T' maximale cohérente. On utilise pour cela le lemme de Zorn : l'ensemble des théories cohérentes qui contiennent T, ordonné par l'inclusion, est inductif (c'est une conséquence du lemme de finitude), il admet donc un élément maximal T'. Bien évidemment, T' admet toujours des témoins de Henkin (puisque le langage est inchangé et que T a été augmentée).

Fait 2.35 *Soient T' une théorie maximale cohérente, et F, G des énoncés. Alors $T' \vdash_d F$ si et seulement si $F \in T'$; $T' \vdash_d F$ si et seulement si $T' \nvdash_d \neg F$; $T' \vdash_d (F \wedge G)$ si et seulement si $T' \vdash_d F$ et $T' \vdash_d G$. Preuve en exercice, en utilisant une bonne réécriture pour le connecteur $\rightarrow$.*

Construisons le modèle suivant $\mathcal{M}$ de T'. Soit $\mathcal{T}_0$ l'ensemble des termes sans variables (cet ensemble est non vide car le langage possède nécessairement des constantes, les constantes de Henkin), et $\sim$ la relation sur $\mathcal{T}_0$ définie par $t_1 \sim t_2$ si et seulement si $T' \vdash_d t_1 \equiv t_2$. On voit aisément qu'il s'agit d'une relation d'équivalence, en utilisant les axiomes pour l'égalité (a), la règle de généralisation, l'axiome pour $\forall$ (a), et le modus ponens. On prend comme base de notre modèle l'ensemble $M := \mathcal{T}_0 / \sim$. On doit encore donner les interprétations des différents symboles de $\mathcal{L}$:

- si c est un symbole de constante, son interprétation $\overline{c}$ est la classe du terme $c \in \mathcal{T}_0$

- si R est un symbole de relation d'arité n, et $m_1, \ldots, m_n$ des éléments de M, on choisit pour chaque i un représentant $t_i \in \mathcal{T}_0$ de m_i, et on pose $(m_1, \ldots, m_n) \in \overline{R}$ si et seulement si $T' \vdash_d R(t_1) \ldots (t_n)$ (c'est indépendant du choix des représentants par l'axiome (c) pour $\equiv$, la généralisation et l'axiome (a) pour $\forall$)
- si f est un symbole de fonction d'arité n, et $m_1, \ldots, m_n$ des éléments de M, on choisit pour chaque i un représentant $t_i \in \mathcal{T}_0$ de m_i, et on pose pour $\overline{f}(m_1, \ldots, m_n)$ la classe de $f(t_1) \ldots (t_n) \in \mathcal{T}_0$ (c'est indépendant du choix des représentants d'après l'axiome (b) pour $\equiv$, la généralisation et l'axiome (a) pour $\forall$)

Reste à vérifier que la $\mathcal{L}$-structure $\mathcal{M}$ ainsi obtenue est bien un modèle de T'. On va montrer plus précisément, par induction sur les formules, que si F est un énoncé, alors $\mathcal{M} \models F$ si et seulement si $F \in T'$:

- si F est une formule atomique $R(t_1) \ldots (t_n)$, c'est une conséquence directe de la définition
- si $F = \neg G$, on sait comme T' est maximale cohérente que $\neg G \in T'$ si et seulement si $G \notin T'$, ce qui équivaut à $\mathcal{M} \not\models G$ par induction, et donc à $\mathcal{M} \models F$
- si $F = G \wedge H$, on sait comme T' est maximale cohérente que $F \in T$ si et seulement si $G \in T$ et $H \in T$, ce qui équivaut à $\mathcal{M} \models G$ et $\mathcal{M} \models H$ par induction, et donc à $\mathcal{M} \models F$
- si $F = \forall x G$, puisque F n'a pas de variable libre, G a au plus pour variable libre x. Si $T' \vdash_d F$, on obtient par l'axiome pour $\forall$ (a) et par modus ponens que $T' \vdash_d G[t/x]$ pour tout terme $t \in \mathcal{T}_0$, et donc par hypothèse d'induction, $\mathcal{M} \models G[\overline{t}]$, où $\overline{t}$ est la classe de t. Comme $\overline{t}$ décrit M quand t décrit $\mathcal{T}_0$, on a bien $\mathcal{M} \models \forall x G[x]$. Si $T' \not\vdash_d F$, comme T' est maximale cohérente, $T' \vdash_d \neg \forall x G$. Il vient facilement (car $\forall x \neg\neg G \to \forall x G$ est démontrable dans toute théorie, exercice) que $T' \vdash_d \neg \forall x \neg\neg G$, c'est-à-dire $T' \vdash_d \exists x \neg G$. Comme T' admet des témoins de Henkin, on obtient par modus ponens que $T' \vdash_d \neg G[c_{\neg G}/x]$. D'après l'hypothèse d'induction, $\mathcal{M} \models \neg G[\overline{c_{\neg G}}]$, donc $\mathcal{M} \not\models \forall x G$.

A fortiori, $\mathcal{M}$ est un modèle de T. □

Preuve (Théorème de complétude) Si T est cohérente, on trouve $\mathcal{L}'$ et T' comme dans lemme 2.33. Par le lemme 2.34, il existe une $\mathcal{L}'$-structure $\mathcal{M}'$ qui est un modèle de T'. Le réduit $\mathcal{M}$ de $\mathcal{M}'$ à $\mathcal{L}$ est un modèle de T.

Si T est consistante, considérons un modèle $\mathcal{M}$ de T. Soit $(F_0, \ldots, F_n)$ une démonstration d'un énoncé F_n à partir de T, on montre par induction sur i que, si $F_i = F_i[x_1, \ldots, x_r]$, alors $\mathcal{M} \models \forall x_1 \ldots \forall x_r F_i$. Pour cela, on vérifie que toutes les tautologies propositionnelles, les axiomes de T, les axiomes pour l'égalité, les axiomes pour $\forall$, sont vrais dans $\mathcal{M}$. De plus, si $\forall x_1 \ldots \forall y_s F_j[x_1, \ldots, x_r, y_1, \ldots, y_s]$ et $\forall x_1 \ldots \forall y_s (F_j \to F_i)$ sont vrais dans $\mathcal{M}$, alors $\forall x_1 \ldots \forall x_r F_i$ aussi ; et si $\forall x_1 \ldots \forall x_r \forall x F_i[x_1, \ldots, x_r, x]$ est vrai dans $\mathcal{M}$, alors $\forall x_1 \ldots \forall x_r (\forall x F_i)[x_1, \ldots, x_r]$ aussi. Comme F_n est un énoncé, on obtient ainsi $\mathcal{M} \models F_n$. Comme $\mathcal{M} \not\models \forall x \neg x \equiv x$, T est cohérente. □

Remarque 2.36 *On peut voir le théorème de complétude comme un accomplissement partiel du théorème de Hilbert : il permet de réduire la notion de conséquence à des manipulations finitaires (suites finies de formules suivant des*

règles déterminées).

2.6 Le théorème de compacité

Nous pouvons déduire du théorème de complétude l'important théorème suivant.

Théorème 2.37 (Théorème de compacité) *Soit T une théorie. Alors T est consistante si et seulement si tout sous-ensemble fini de T est consistant.*

Preuve Comme pour le théorème de compacité du calcul propositionnel, une direction est claire : si T admet un modèle, alors ce modèle est a fortiori un modèle de chacun des sous-ensemble fini de T.
Montrons maintenant la contraposée : on suppose que T est inconsistante, c'est-à-dire, d'après le théorème de complétude, incohérente. On a vu qu'il existe alors une formule F telle que $T \vdash F$ et $T \vdash \neg F$. Par le lemme de finitude, on trouve des sous-ensembles finis T_0 et T_1 de T tels que $T_0 \vdash F$ et $T_1 \vdash \neg F$. Alors $T_0 \cup T_1$ est un sous-ensemble fini de T qui est incohérent, c'est-à-dire inconsistant : c'est ce qu'on voulait démontrer. □

Ce théorème est d'une grande importance pour la construction des modèles, c'est un des théorèmes de base de la branche de la logique qu'on appelle théorie des modèles. Citons une conséquence.

Corollaire 2.38 *Soit T une théorie. On suppose que T admet des modèles de taille arbitrairement grande. Alors T admet un modèle infini.*

Preuve Ajoutons au langage $\mathcal{L}$ considéré une infinité de symboles de constantes $\{c_i\}_{i\in\mathbb{N}}$, distincts des symboles de constantes de $\mathcal{L}$. On considère la théorie T', dans le langage $\mathcal{L}' = \mathcal{L} \cup \{c_i\}_{i\in\mathbb{N}}$:

$$T' := T \cup \{\neg c_i \equiv c_j ; i \neq j\}.$$

Montrons que T' est consistante. Pour cela, d'après le théorème de compacité, il suffit de montrer que toute partie finie T'_0 de T' est consistante. Un tel T'_0 ne fait intervenir qu'un nombre fini de symbole de constantes c_i ; il existe donc un entier N tel que $T'_0 \subseteq T \cup \{\neg c_i \equiv c_j ; i, j \leq N-1, i \neq j\}$. Construisons un modèle de T'_0 : pour cela, il suffit de considérer un modèle $\mathcal{M}$ de T de taille supérieure ou égale à N (ce qui est possible car T admet des modèles arbitrairement grands) et d'enrichir $\mathcal{M}$ en une $\mathcal{L}'$-structure en interprétant les $c_0, \ldots, c_{N-1}$ par N éléments deux à deux distincts de M. Cette $\mathcal{L}'$-structure est alors un modèle de T'_0. On peut donc trouver un modèle $\mathcal{N}'$ de T', ce modèle est infini car il contient les éléments $\{\overline{c_i}\}_{i\in\mathbb{N}}$ qui sont deux à deux disjoints. En prenant le réduit $\mathcal{N}$ de $\mathcal{N}'$ au langage $\mathcal{L}$, on obtient un modèle infini de T. □

Interprétation topologique du théorème de compacité
Comme le théorème de compacité du calcul propositionnel, ce théorème admet une interprétation topologique. Pour $\mathcal{L}$ un langage donné, on considère X l'ensemble des théories maximales consistantes (ou encore l'ensemble des théories complètes dans lequel on identifie deux théories qui ont même ensemble de

conséquences) dans le langage $\mathcal{L}$. On définit une topologie sur X en se donnant une base d'ouvert : pour tout énoncé F du langage $\mathcal{L}$, on définit l'ouvert de base

$$\langle F\rangle := \{T \in X; T \vdash F\}.$$

Notons que $\langle F\rangle \cap \langle G\rangle = \langle F \wedge G\rangle$ et $\langle F\rangle^c = \langle \neg F\rangle$ sont encore des ouverts de base ; en particulier, nous avons obtenu une base d'ouvert-fermés.
Pour cette topologie X est un ensemble séparé : soit T_0 et T_1 deux points distincts de X. Quitte à inverser le rôle de T_0 et T_1, on trouve un énoncé F tel que $F \in T_0$ et $F \notin T_1$. Comme T_1 est complète, $\neg F \in T_1$. Ainsi, $T_0 \in \langle F\rangle$ et $T_1 \in \langle \neg F\rangle$, où $\langle F\rangle$ et $\langle \neg F\rangle$ sont deux ouverts disjoints.
Ce que nous dit le théorème de compacité, c'est que X, pour cette topologie, est un ensemble compact : soit Y_i un fermé, il s'écrit comme une intersection de fermés de base, $Y_i = \bigcap_{j\in J_i}\langle F_{i,j}\rangle$. Dire que Y_i est non vide revient à dire qu'il existe une théorie complète T telle que $F_{i,j} \in T$ pour tout j, c'est-à-dire que l'ensemble $\{F_{i,j}; j \in J_i\}$ est consistant. Donc, si $\bigcap_{i\in I} Y_i = \emptyset$, la théorie $\{F_{i,j}; i \in I, j \in J_i\}$ est inconsistante. Par le théorème de compacité, on peut donc en extraire un sous-ensemble fini qui est inconsistant. Ce sous-ensemble ne fait intervenir qu'un ensemble fini I_0 d'indices i. On a a fortiori que $\{F_{i,j}; i \in I_0, j \in J_i\}$ est inconsistant, donc $\bigcap_{i\in I_0} Y_i$ est vide.

Exemple 2.39 *Soit le langage $\mathcal{L} = (\emptyset, \{+, -, \cdot\}, \{0, 1\})$, on considère dans ce langage les théories $Th(\mathbb{F}_p)$, où p est un nombre premier et $\mathbb{F}_p$ le corps à p éléments, avec l'interprétation naturelle du langage. On a alors que la suite de ces théories admet (au moins) une valeur d'adhérence T, par compacité. Montrer que les modèles de T sont des corps de caractéristique zéro. Est-ce que la théorie des corps de caractéristique zéro est complète ?*

2.7 Un peu de théorie des modèles

Théorème 2.40 (Théorème de Löwenheim-Skolem) *Soit $\mathcal{L}$ un langage fini ou dénombrable, et T une théorie consistante dans ce langage. Alors T admet un modèle de cardinalité au plus dénombrable.*
Si de plus T admet un modèle infini, alors elle admet un modèle dénombrable.

Preuve La preuve repose essentiellement sur la remarque suivante : si $\mathcal{L}$ est fini ou dénombrable, alors l'ensemble des formules dans $\mathcal{L}$ est dénombrable (puisque c'est un sous-ensemble de l'ensemble des mots, c'est-à-dire l'union dénombrable (pour $n \geq 1$) des ensembles dénombrables $\mathcal{S}^n$ (où $\mathcal{S}$ désigne l'ensemble des symboles)).
On reprend alors la démonstration du théorème des complétude : on trouve $\mathcal{L}' \supseteq \mathcal{L}$ et $T' \supseteq T$ dans le langage $\mathcal{L}'$ qui est cohérente et qui admet des témoins de Henkin (lemme 2.33), et par construction $\mathcal{L}'$ est dénombrable puisque c'est l'union dénombrable de langages $\mathcal{L}_i$ tous dénombrables (par récurrence sur i, et en utilisant la remarque préliminaire). Puis, le modèle de T que l'on a construit est un réduit d'une $\mathcal{L}'$-structure dont l'ensemble de base M est le quotient de l'ensemble des termes sans variable $\mathcal{T}_0$ par une certaine relation d'équivalence. Or, comme dans la remarque préliminaire, on constate que $\mathcal{T}_0$ est dénombrable.
Pour la deuxième partie, il suffit d'ajouter à $\mathcal{L}$ une infinité dénombrable de nouveaux symboles de constantes $(c_i)_{i\in\mathbb{N}}$, de regarder dans ce nouveau langage la

théorie $T' = T \cup \{\neg c_i \equiv c_j; i \neq j\}$, qui est consistante (prendre un modèle infini de T et l'enrichir en interprétant les symboles c_i par des éléments deux à deux distincts). Il reste à prendre alors un modèle fini ou dénombrable de T' (par la première partie), les axiomes de T' impliquent qu'il est infini. □

Pour le théorème suivant, on anticipe un peu le chapitre sur la théorie des ensembles (avec axiome du choix) : à tout ensemble X on peut associer un cardinal $card(X)$ qui est une mesure de sa "taille" (c'est-à-dire que $card(X) = card(Y)$ si et seulement s'il existe une bijection entre X et Y). L'ensembles des cardinaux est totalement ordonné ; on peut aussi noter que l'ensemble des cardianux infinis est infini.

Théorème 2.41 (Théorème de Löwenheim-Skolem généralisé) *Soit $\mathcal{L}$ un langage de cardinal κ, T une théorie dans ce langage admettant des modèles infinis et $\lambda \geq \kappa$ un cardinal infini.*
Alors T admet un modèle de cardinal λ.

La preuve de ce théorème n'est pas plus difficile que celle du théorème précédent, il s'agit de reprendre la même méthode en utilisant un peu d'arithmétique cardinale de base.

Soient $\mathcal{M} \subseteq_{\mathcal{L}} N$ des $\mathcal{L}$-structures, on sait que pour toute formule $F[x_1, \ldots, x_n]$ sans quantificateur, et pour tout $a_1, \ldots, a_n$, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{N} \models F[a_1, \ldots, a_n]$. Cela inspire la définition suivante :

Définition 2.42 *Soient $\mathcal{M} \subseteq_{\mathcal{L}} N$. On dit que $\mathcal{M}$ est une sous-structure élémentaire de $\mathcal{N}$, ou encore que $\mathcal{N}$ est une extension élémentaire de $\mathcal{M}$, et on note $\mathcal{M} \preceq_{\mathcal{L}} \mathcal{N}$ (éventuellement sans $\mathcal{L}$ si le langage est sous-entendu), si pour toute formule $F[x_1, \ldots, x_n]$ et tous $a_1, \ldots, a_n$ dans M, $\mathcal{M} \models F[a_1, \ldots, a_n]$ si et seulement si $\mathcal{N} \models F[a_1, \ldots, a_n]$ (noter que contrairement à la situation de la remarque précédente, F peut avoir des quantificateurs).*
On dit que qu'une $\mathcal{L}$-structure $\mathcal{M}_0$ se plonge (élémentairement) dans une $\mathcal{L}$-structure $\mathcal{M}_1$ s'il existe un isomorphisme de $\mathcal{M}_0$ vers une sous-structure (élémentaire) de $\mathcal{M}_1$.

Définition 2.43 *Soit $\mathcal{M}$ une $\mathcal{L}$-structure. On pose $\mathcal{L}_{\mathcal{M}}$ le langage obtenu en ajoutant à $\mathcal{L}$ un nouveau symbole de constante pour chaque élément de M : $\mathcal{L}_{\mathcal{M}} = \mathcal{L} \cup \{c_m; m \in M\}$. On note $\mathcal{M}^*$ l'enrichissement de $\mathcal{M}$ en une $\mathcal{L}_{\mathcal{M}}$-structure donné par $\overline{c_m}^{\mathcal{M}^*} = m$.*

Remarque 2.44 *Pour toute formule $F[x_1, \ldots, x_n]$ et $m_1, \ldots, m_n$ dans M, $\mathcal{M} \models F[m_1, \ldots, m_n]$ si et seulement si $\mathcal{M}^* \models F[c_{m_1}/x_1, \ldots, c_{m_n}/x_n]$ (la première satisfaction est celle d'une formule avec variables libres, avec des assignations pour ces variables, alors que la deuxième est la satisfaction d'un énoncé du langage $\mathcal{L}_{\mathcal{M}}$).*

Définition 2.45 (Diagrammes) *Le diagramme de $\mathcal{M}$, noté $D(\mathcal{M})$, est l'ensemble des énoncés sans quantificateur F dans le langage $\mathcal{L}_{\mathcal{M}}$ tel que $\mathcal{M}^* \models F$.*
Le diagramme élémentaire de $\mathcal{M}$, noté $D_e(\mathcal{M})$, est l'ensemble des énoncés F

dans le langage $\mathcal{L}_\mathcal{M}$ tel que $\mathcal{M}^* \models F$ (clairement, $D(\mathcal{M}) \subseteq D_e(\mathcal{M})$).

Proposition 2.46 *Soient $\mathcal{M}$ et $\mathcal{N}$ deux $\mathcal{L}$-structures, alors $\mathcal{M}$ se plonge (resp. élémentairement) dans $\mathcal{N}$ si et seulement s'il existe un enrichissement $\mathcal{N}'$ de $\mathcal{N}$ de $\mathcal{L}$ à $\mathcal{L}_\mathcal{M}$ tel que $\mathcal{N}' \models D(\mathcal{M})$ (resp. $\mathcal{N}' \models D_e(\mathcal{M})$).*
Ou encore : $\mathcal{M}$ se plonge (resp. élémentairement) dans un réduit de $\mathcal{N}'$ de $\mathcal{L}_\mathcal{M}$ à $\mathcal{L}$ si et seulement si $\mathcal{N}' \models D(\mathcal{M})$ (resp. $\mathcal{N}' \models D_e(\mathcal{M})$).

Preuve Soit $\phi : \mathcal{M} \to \mathcal{N}$ un plongement, éventuellement élémentaire. On enrichit $\mathcal{N}$ en posant, pour tout $m \in M$, $\overline{c_m}^{\mathcal{N}'} = \phi(m)$. Comme ϕ est un isomorphisme de $\mathcal{M}$ vers une sous-structure (éventuellement élémentaire) $\mathcal{M}_0$ de $\mathcal{N}$, on a pour toute formule $F[x_1, \ldots, x_n]$ (sans quantificateur si le plongement est simple, ou éventuellement avec quantificateur si le plongement est élémentaire), et pour tous $m_1, \ldots, m_n$ dans M :
$F[c_{m_1}, \ldots, c_{m_n}] \in D(\mathcal{M})$ (ou éventuellement $\mathcal{D}_e(\mathcal{M})$) ssi $\mathcal{M} \models F[m_1, \ldots, m_n]$ ssi $\mathcal{M}_0 \models F[\phi(m_1), \ldots, \phi(m_n)]$ ssi $\mathcal{N} \models F[\phi(m_1), \ldots, \phi(m_n)]$ (par sous-structure, éventuellement élémentaire) ssi $\mathcal{N}' \models F[c_{m_1}, \ldots, c_{m_n}]$ (car $\overline{c_{m_i}}^{\mathcal{N}'} = \phi(m_i)$).
Cela donne la première implication.
Pour la réciproque, soit $\mathcal{N}' \models D(\mathcal{M})$ et $\mathcal{N}$ le réduit de $\mathcal{N}'$ de $\mathcal{L}_\mathcal{M}$ à $\mathcal{L}$. On construit $\phi : M \to N$ en posant $\phi(m) = \overline{c_m}^{\mathcal{N}'}$, et on pose pour M_0 l'image de ϕ. Alors M_0 est stable par les interprétations $\overline{f}^{\mathcal{N}}$ pour f une fonction dans $\mathcal{L}$ (on la suppose unaire pour ne pas alourdir l'écriture) : pour $\phi(m) \in M_0$, $\mathcal{M}^* \models f(c_m) \equiv c_{\overline{f}^{\mathcal{M}}(m)}$, donc $\mathcal{N}'$ satisfait aussi cet énoncé de $D(\mathcal{M})$, ce qui peut encore s'écrire $\mathcal{N} \models f(\phi(m)) = \phi(\overline{f}^{\mathcal{M}}(m))$: on a bien $\overline{f}^{\mathcal{N}}(\phi(m)) \in M_0$, et on voit même au passage que ϕ respecte l'interprétation de f. On peut faire la même chose pour les symboles de constantes d de $\mathcal{L}$, ce qui permet d'avoir $\overline{d}^{\mathcal{N}} = \phi(\overline{d}^{\mathcal{M}})$. On peut alors munir M_0 de la $\mathcal{L}$-structure induite par $\mathcal{N}$, c'est-à-dire telle que $\mathcal{M}_0 \subseteq_\mathcal{L} \mathcal{N}$. Reste à voir que $\phi : \mathcal{M} \to \mathcal{M}_0$ est un isomorphisme. Il est clair que ϕ est surjective sur M_0, et on a déjà remarqué que ϕ préserve les fonctions et les constantes de $\mathcal{L}$; enfin si R est un symbole de relation (unaire pour simplifier), et m dans M, alors :
$m \in \overline{R}^{\mathcal{M}}$ ssi $\mathcal{M}^* \models R[c_m]$ ssi $\mathcal{N}' \models R[c_m]$ ssi $\phi(m) \in \overline{R}^{\mathcal{N}}$ (car $\overline{c_m}^{\mathcal{N}'} = \phi(m)$) ssi $\phi(m) \in \overline{R}^{\mathcal{M}_0}$ car $\overline{R}^{\mathcal{M}_0} = \overline{R}^{\mathcal{N}} \cap M_0$ par construction de la structure induite.
Ainsi, ϕ est bien un plongement $\mathcal{M} \to \mathcal{N}$.
Enfin, si $\mathcal{N}' \models D_e(\mathcal{M}) \supseteq D(\mathcal{M})$, on peut a fortiori faire la construction du plongement ϕ précédent, et c'est un plongement élémentaire puisque pour toute formule $F[x]$ (une seule variable libre pour simplifier), et tout $\phi(m)$ dans M_0 :
$\mathcal{M}_0 \models F[\phi(m)]$ ssi $\mathcal{M} \models F[m]$ (isomorphisme) ssi $F[c_m] \in D_e(\mathcal{M})$ (regarder $\neg F[c_m]$ pour la réciproque) ssi $\mathcal{N}' \models F[c_m]$ (idem) ssi $\mathcal{N} \models F[\phi(m)]$ (car $\overline{c_m}^{\mathcal{N}'} = \phi(m)$). □

Exemple 2.47 *Soit $\mathcal{M}$ une $\mathcal{L}$-structure infinie. Alors pour tout cardinal $\lambda \geq card(\mathcal{L}_\mathcal{M})$, $\mathcal{M}$ se plonge (élémentairement si on veut) dans une $\mathcal{L}$-structure de cardinal λ : il suffit d'utiliser la caractérisation précédente, et d'appliquer le théorème de Löwenheim-Skolem généralisé.*

Corollaire 2.48 *Soient deux $\mathcal{L}$-structures $\mathcal{M}_1$ et $\mathcal{M}_2$. Alors $\mathcal{M}_1$ et $\mathcal{M}_2$ se plongent élémentairement dans une même $\mathcal{L}$-structure $\mathcal{N}$ si et seulement si $\mathcal{M}_1 \equiv \mathcal{M}_2$.*

Preuve Pour le sens direct : dans une sous-structure élémentaire, la satisfaction des formules à paramètres dans la petite structure est préservée, et donc a fortiori celle des énoncés (c'est-à-dire si $\mathcal{M} \preceq \mathcal{N}$, alors $\mathcal{M} \equiv \mathcal{N}$). La conclusion vient donc par transitivité $\mathcal{M}_1 \equiv \mathcal{N} \equiv \mathcal{M}_2$ (on a aussi utilisé la préservation de la satisfaction des énoncés par isomorphisme).
Pour la réciproque, on suppose $\mathcal{M}_1 \equiv \mathcal{M}_2$, et par la proposition précédente, il suffit de montrer que $D_e(\mathcal{M}_1) \cup D_e(\mathcal{M}_2)$ est consistante (remarque : on a pris $\mathcal{L}_{\mathcal{M}_1} = \mathcal{L} \cup \{c_m; m \in M_1\}$ et $\mathcal{L}_{\mathcal{M}_2} = \mathcal{L} \cup \{d_m; m \in M_2\}$ pour des ensembles disjoints de nouvelles variables). D'après le théorème de compacité, il suffit de vérifier que $X \cup D_e(\mathcal{M}_2)$ est consistante pour toute partie finie $X \subseteq D_e(\mathcal{M}_1)$, et comme une conjonction finie d'énoncés dans $D_e(\mathcal{M}_1)$ est encore dans $D_e(\mathcal{M}_1)$ par construction, il suffit de vérifier que $D_e(\mathcal{M}_2) \cup \{F\}$ est consistante pour tout $F \in D_e(\mathcal{M}_1)$. Supposons le contraire, on a donc $F \in D_e(\mathcal{M}_1)$ tel que $D_e(\mathcal{M}_2) \cup \{F\}$ est inconsistant, c'est-à-dire tel que $D_e(\mathcal{M}_2) \vdash \neg F$. Ecrivons F sous la forme $F = G[c_{m_1}, \ldots, c_{m_r}]$ pour $G[x_1, \ldots, x_r]$ une formule dans le langage $\mathcal{L}$. A partir d'une démonstration formelle de $\neg G[c_{m_1}, \ldots, c_{m_r}]$ partant de $D_e(\mathcal{M}_2)$, on construit, comme dans le lemme 2.33, une démonstration formelle de $\forall x_1 \ldots \forall x_r \neg G[x_1, \ldots, x_r]$ (on utilise le fait que les constantes $c_{m_1}, \ldots, c_{m_r}$ n'apparaissent pas dans la théorie $D_e(\mathcal{M}_2)$). Ainsi $\mathcal{M}_2$ satisfait l'énoncé (dans le langage $\mathcal{L}$) $\forall x_1 \ldots \forall x_r \neg G[x_1, \ldots, x_r]$, mais comme $\mathcal{M}_1 \equiv \mathcal{M}_2$, on a aussi $\mathcal{M}_1 \models \forall x_1 \ldots \forall x_r \neg G[x_1, \ldots, x_r]$. Cela contredit le fait que $\mathcal{M}_1 \models G[m_1, \ldots, m_r]$. □

Chapitre 3

Théorie des ensembles

But : construire une théorie du premier ordre dans laquelle on pourra introduire la quasi-totalité des constructions mathématiques.

3.1 Axiomatisations de Zermelo et de Zermelo-Fraenkel

Langage $\mathcal{L} = \{\in\}$. Dans toute la suite, on se place dans une $\mathcal{L}$-structure $\mathcal{U} = (U, \in)$. On parlera d'ensembles pour désigner les éléments de U. Quand on parlera de parties de U qui ne correspondent pas à des ensembles, on parlera de collections, ou de classe si cette collection est définie par une formule.
Notations : $\forall x \in y\ F$ pour $\forall x(x \in y \to F)$ et $\exists x \in y\ F$ pour $\exists x(x \in y \wedge F)$.

Axiome d'extensionnalité : $\forall x \forall y(\forall z(z \in x \leftrightarrow z \in y) \to x \equiv y)$

L'implication réciproque est une conséquence des axiomes pour l'égalité.
On notera $x \subseteq y$ pour $\forall z \in x\ z \in y$, l'axiome d'extensionnalité nous dit que $x \subseteq y$ et $y \subseteq x$ impliquent $x = y$.
Les axiomes suivants exprimeront l'existence d'ensembles dont les éléments vérifient certaines propriétés ; à chaque fois, l'axiome d'extensionnalité nous donnera l'unicité de ces ensembles dont on a caractérisé les éléments.

Axiome de la paire : $\forall x \forall y \exists z \forall w(w \in z \leftrightarrow (w = x \vee w = y))$

Pour deux éléments a, b de U, on note $\{a, b\}$ l'unique ensemble dont les éléments sont exactement a et b, ou $\{a\}$ si $a = b$.

Axiome de la réunion : $\forall x \exists y \forall z(z \in y \leftrightarrow (\exists w \in x\ z \in w))$

On note $\bigcup x$, ou $\bigcup_{w \in x} w$, l'ensemble donné par cet axiome.
Premières constructions :

- pour a, b deux éléments de U, on peut construire leur union : $a \cup b = \bigcup\{a, b\}$.
- pour un nombre fini $a_1, \ldots, a_n$ d'éléments de U, on peut construire un ensemble qui contient exactement ces éléments, par récurrence sur n :

$$\{a_1, \ldots, a_n\} = \{a_1, \ldots, a_{n-1}\} \cup \{a_n\}$$

Axiome des parties : $\forall x \exists y \forall z (z \in y \leftrightarrow z \subseteq x)$

On note $\mathcal{P}(x)$.

Schéma d'axiomes de compréhension : pour toute formule $F[x, x_1, \ldots, x_n]$, on a l'axiome

$$\forall x_1 \ldots \forall x_n \forall y \exists z \forall x (x \in z \leftrightarrow (x \in y \wedge F[x, x_1, \ldots, x_n]))$$

Pour $a_1, \ldots, a_n, b$ dans U, on note cet ensemble $\{x \in b; F[x, a_1, \ldots, a_n]\}$. Ce schéma d'axiome résulte d'un compromis : on veut pouvoir parler d'ensemble défini par une propriété, en espérant ne pas introduire de contradiction dans notre théorie. A ce titre, la partie « $x \in y$ » est nécessaire. Cf le paradoxe de Russell : l'existence de l'ensemble $\{x; x \notin x\}$ est contradictoire.
Quelques constructions supplémentaires :

- Pour a un élément de U, on construit $\{x \in a; x \neq x\}$. Par l'axiome d'extensionnalité, cet ensemble ne dépend pas u choix de a, on le note $\emptyset$ (ou parfois 0, et 1 pour $\mathcal{P}(0)$).
- Pour a, b dans U, on a l'ensemble $a \cap b = \{x \in a; x \in b\}$. Pour c un ensemble non vide, on peut choisir $d \in c$ et considérer l'ensemble $\{x \in d; \forall y \in c \; x \in y\}$. Cet ensemble ne dépend pas du choix de $d \in c$, on le note $\bigcap c$ ou $\bigcap_{y \in c} y$.
- Pour a, b dans U, on construit le couple $(a, b) = \{\{a\}, \{a, b\}\}$. Exercice : $(a, b) = (c, d)$ si et seulement si $a = b$ et $c = d$. Remarque : si $a \in c$ et $b \in d$, $(a, b) \in \mathcal{P}(\mathcal{P}(c \cup d))$. On peut aussi définir par récurrence sur n : $(a_1, \ldots, a_n) = (a_1, (a_2, \ldots, a_n))$.
- Produit cartésien : pour a, b dans U, on définit

  $$a \times b = \{x \in \mathcal{P}(\mathcal{P}(a \cup b)); \exists y \in a \exists z \in b \; x = (y, z)\}$$

  Et par induction, pour tout entier $n \geq 1$, $a_1 \times \ldots \times a_n$ et a^n.
- Union disjointe : pour a, b dans U, on construit

  $$a \dot{\bigcup} b = a \times \{0\} \cup b \times \{1\}$$

Schéma d'axiomes de remplacement :
Soit $F[x, y, x_1, \ldots, x_n]$ une formule, et $a_1, \ldots, a_n$ des éléments de U. On dit que $F[x, y, a_1, \ldots, a_n]$ est une fonctionnelle si $\forall x \forall y \forall z ((F[x, y, \overline{a}] \wedge F[x, z, \overline{a}]) \rightarrow y \equiv z)$.
Le domaine de la fonctionnelle $F[x, y, \overline{a}]$ est la classe des b dans U tels qu'il existe c tel que $F[b, c, \overline{a}]$. L'image la fonctionnelle est la classe des c dans U tels qu'il existe b tel que $F[b, c, \overline{a}]$.
On appelle fonction (partielle) de a dans b un sous-ensemble $f \subseteq a \times b$ tel que la formule $(x, y) \in f$ soit une fonctionnelle. Si le domaine de f est a, la fonction est dite totale. Pour chaque formule $F[x, y, x_1, \ldots, x_n]$, on considère l'axiome :

$$\forall x_1 \ldots \forall x_n (F[x, y, x_1, \ldots, x_n] \text{ est une fonctionnelle} \rightarrow \forall z \exists w \forall y (y \in w \leftrightarrow \exists x \in z \; F[x, y, x_1, \ldots, x_n]))$$

On dira aussi que l'ensemble w est l'image ensembliste de l'ensemble z par la fonctionnelle F.
Remarque : le schéma de remplacement implique le schéma de compréhension : l'ensemble $\{x \in b; F[x, \overline{a}]\}$ est l'image de l'ensemble b par la fonctionnelle $G[x, y, \overline{a}] = (y \equiv x \wedge F[x, \overline{a}])$.

Corollaire 3.1 *Soit F une fonctionelle de domaine un ensemble a. Alors il existe une fonction f telle que $(x, y) \in f \leftrightarrow F[x, y]$.*

Preuve Par le schéma de remplacement, b, l'image ensembliste de a par F, est un ensemble. Il suffit alors de définir f en utilisant le schéma de compréhension :

$$f = \{z \in a \times b; \exists x \in a \; \exists y \in b \; F[x, y]\}.$$

□.

La théorie Z^- est formée des axiomes d'extensionalité, de la paire, de la réunion, des parties et de compréhension.
La théorie ZF^- est formée des axiomes d'extensionalité, de la paire, de la réunion, des parties et de remplacement.
Pour obtenir les théories Z et ZF, on ajoutera l'axiome de l'infini.

3.2 Ordinaux

On se place désormais dans un modèle $\mathcal{U} \models ZF^-$.

Définition 3.2
- *Relation d'ordre strict : c'est une classe de couples $R[x, y]$ (éventuellement avec paramètres) vérifiant les propriétés d'irreflexivité $\forall x \neg R[x, x]$ et de transitivité $\forall x \forall y \forall z((R[x, y] \wedge R[y, z]) \rightarrow R[x, z])$. Notation : $x <_R Y$ et $x \leq_R y$ pour l'ordre large associé. L'ordre est total si $\forall x \forall y(R[x, y] \vee R[y, x] \vee x \equiv y)$.*
- *Un ensemble ordonné est un couple (a, r) d'élément de U tels que $r \subseteq a^2$ et que la formule $R[x, y] = (x, y) \in r$ est un ordre strict*
- *Un bon ordre est un ensemble ordonné (a, r) dont tout sous-ensemble non vide possède un plus petit élément :*

$$\forall y \in \mathcal{P}(a)(y \not\equiv \emptyset \rightarrow \exists x_0 \in y \; \forall x \in y \; x_0 \leq_r x)$$

En particulier, un bon ordre est total.
- *Classe bien ordonnée (A, R) :*

$$\forall y((\forall z \in y \; A[z]) \wedge y \not\equiv \emptyset) \rightarrow \exists x_0 \in y \; \forall x \in y \; x_0 \leq_R x)$$

$$\forall x(A[x] \rightarrow \exists y \forall z(z \in y \leftrightarrow (A[z] \wedge z <_R x)))$$

On notera cet ensemble : $S_a(A, R) = \{z; A[z] \wedge z <_R a\}$, ou S_a s'il n'y a pas d'ambiguité.

Proposition 3.3 *Soit (A, R) une classe bien ordonnée. Alors toute sous-classe non vide de A admet un plus petit élément.*

Preuve Soit B une sous-classe non vide de A et a dans B. Par définition, S_a est un sous-ensemble de A ; si $\{x \in S_a; B[x]\}$ est vide, a est le plus petit élément de B (car l'ordre est total dans une classe bien ordonnée, puisque toute paire d'éléments a un plus petit élément), sinon, cet ensemble admet un plus petit élément, qui est aussi un plus petit élément pour B. □

Remarque : pour a un élément de U, toute sous-classe B de la classe $x \in a$ est un ensemble d'après le schéma de compréhension.

Corollaire 3.4 *Principe d'induction sur les classes bien ordonnées. Soit (A, R) une classe bien ordonnée, et B une classe. Alors*

$$\mathcal{U} \models \forall x(A[x] \to (\forall y <_R x\ B[y] \to B[x])) \to \forall x(A[x] \to B[x])$$

Preuve Appliquer la proposition précédente à la classe $A \wedge \neg B$. □

Exemples de bons ordres :
- les ordres totaux finis (au sens intuitif)
- $(\mathbb{N}, \leq)$ (noté ω) (au sens intuitif)
- la concaténation $a \cup^{\cdot} b$, où a et b sont des bons ordres
- le produit $a \times b$ avec l'ordre lexicographique, où a et b sont des bons ordres

Proposition et définition 3.5 *Soit (A, R) une classe bien ordonnée.*
- *Soit B une sous-classe de A, majorée dans A. Alors on peut définir sup B comme étant le plus petit majorant de B dans A.*
- *Soit x un élément de A. Alors soit $x = \sup S_x$, et on dit que x est un élément limite, soit $x > \sup S_x$, et nécessairment x est le plus petit majorant strict de sup S_x, on dit que x est le successeur de sup S_x.*

Définition 3.6 *Soit B une sous classe d'une classe totalement ordonnée (A, R). On dit que B est un segment initial de A s'il satisfait la formule $\forall x(B[x] \to \forall y <_R x\ B[y])$.*

Proposition 3.7 *Soit (A, R) une classe bien ordonnée. Tout segment initial propre B de A est de la forme $B = S_x(A, R)$ pour un certain élément x de A (c'est en particulier un ensemble). La fonctionnelle de (A, R) dans la classe des segments initiaux propres de A ordonnée par l'inclusion, définie par $x \mapsto S_x(A, R)$, est un isomorphisme de classes ordonnées.*

Preuve Si B est un segment initial propre de A, il existe un élément dans $A \wedge \neg B$, et cet élément est un majorant strict de B. On pose alors $x =$ le plus petit des majorants stricts de B. On a bien $B = S_x(A, R)$, car z n'est pas dans B si et seulement si c'est un majorant strict de B, si et seulement si $x \leq z$.
La deuxième partie vient immédiatement. □

Proposition 3.8 *Soit (A, R) une classe bien ordonnée et F une fonctionnelle totale strictement croissante de (A, R) dans (A, R). Alors pour tout x dans A, $x \leq F(x)$ ($F(x)$ dénote l'unique y tel que $\mathcal{U} \models F[x, y]$).*

Preuve On prouve la proposition par induction : soit x dans A, on suppose que $y \leq F(y)$ pour tout $y <_R x$ et on veut montrer que $x \leq F(x)$. Si x est le successeur de $y = \sup S_x$, alors par hypothèse d'induction $y \leq F(y) < F(x)$ car F est strictement croissante. Comme x est le successeur de y, $x \leq F(x)$. Si x est un élément limite, $x = \sup S_x \leq \sup \{F(y); y < x\} \leq F(x)$. □

Théorème 3.9 *Principe de définition par induction sur les classes bien ordonnées.*
Soit (A, R) une classe bien ordonnée. Soit H une fonctionnelle de domaine la classe des fonctions de domaine un segment initial propre de (A, R). Alors il existe une unique fonctionnelle F (unique au sens de la fonction intuitive définie, pas de la formule) de domaine A telle que pour tout x dans A, $F(x) = H(F_{|S_x})$.

Preuve On démontre l'unicité par induction : si F et G vérifient la condition requise, et si $F(y) = G(y)$ pour tout $y < x$, alors $F(x) = H(F_{|S_x}) = H(G_{|S_x}) = G(x)$.
Pour l'existence, on définit la formule $F[x, y]$:

$$A[x] \wedge \exists g(\mathrm{Dom}(g) = S_x \wedge \forall z \in S_x \; g(z) = H(g_{|S_z}) \wedge y \equiv H(g)).$$

On a alors que :

- pour x dans le domaine de F, la fonction g_x de domaine S_x exhibée dans la formule F est unique. C'est simplement une conséquence de l'unicité qu'on a vu précédemment appliquée à l'ensemble bien ordonné S_x.
- on en déduit directement que F est bien une fonctionnelle.
- le domaine de F est un segment initial de A. En effet, pour x dans le domaine de F et $z < x$, la fonction $g_z = g_{x|S_z}$ satisfait les bonnes conditions.
- F satisfait la condition voulue sur son domaine. On montre pour celà par induction sur x dans Dom F que $g_x = F_{|S_x}$ et que $F(x) = H(F_{|S_x})$: si c'est vrai pour tout $y < x$, on a par unicité, pour tout $y < x$, $g_{x|S_y} = g_y = F_{|S_y}$. Si x est un élément limite, $S_x = \bigcup_{y \in S_x} S_y$, donc l'égalité précédente donne $g_x = F_{|S_x}$. Si x est le successeur de $y = \sup S_x$, on a $S_x = S_y \cup \{y\}$, avec $g_{x|S_y} = F_{|S_y}$ et $g_x(y) = H(g_{x|S_y}) = H(g_y) = F(y)$. Dans les deux cas, on a $F(x) = H(g_x) = H(F_{|S_x})$.
- Le domaine de F est A. Dans le cas contraire, c'est un segment initial propre de A, donc de la forme S_x. Alors $g_x := F_{|S_x}$ satisfait l'hypothèse voulue dans la formule F, ce qui nous permet de prolonger F en x par $F(x) = H(g_x)$.

□

Remarque Dans l'énoncé du théorème, la condition demandée a un sens car $F_{|S_x}$ est une fonction d'après le schéma de remplacement.
Pour la même raison, si (A, R) est en fait un bon ordre, la fonctionnelle F obtenue est une fonction.

Théorème 3.10 *Trichotomie sur les bons ordres.*
Soient (a, r) et (b, s) deux ensembles bien ordonnés, alors un seul des cas suivants est réalisé :

1. (a, r) *et* (b, s) *sont isomorphes*
2. (a, r) *est isomorphe à un unique segment initial propre de* (b, s)
3. (b, s) *est isomorphe à un unique segment initial propre de* (a, r)

Preuve
Unicité : si on est dans les cas 1 et 3, on obtient par composition une application $f : (a, r) \to S_x(a, r)$ strictement croissante. D'après la proposition 3.8, on a $x \leq f(x)$, contradiction.
Même raisonnement dans le cas de deux segments intitiaux distincts isomorphes.
Existence : on définit une fonction f par induction sur (a, r). On fixe $e \notin b$ et on pose :

$$f(x) = \begin{cases} \text{le plus petit élément de } b \setminus \{f(y); y < x\} \text{ si cet ensemble est non-vide} \\ e \text{ sinon} \end{cases}$$

en utilisant le principe de définition par induction avec la fonctionnelle

$$H[g, y] = ((b \setminus \mathrm{Im}(g) \neq \emptyset \to y = \mathrm{ppe}(b \setminus \mathrm{Im}(g))) \wedge (b \setminus \mathrm{Im}(g) \neq \emptyset \to y = e)).$$

Soit $c = \{x \in a; f(x) \neq e\}$, c'est un segment initial de (a, r). De plus :
- f est un morphisme de (c, r_c) sur (b, s) : si $x < y$, $f(x) < f(y)$ par définition de f
- $\mathrm{Im}(f_{|c})$ est un segment initial de (b, s)
- si $c = a$, (a, r) est isomorphe à un segment initial de (b, s)
- si c est un segment initial propre de (a, r), alors $\mathrm{Im}(f_{|c}) = b$, et f^{-1} est un isomorphisme de (b, s) sur (c, r_c)

□

On montre de la même façon les résultats suivants (en utilisant le fait, conséquence du schéma de remplacement, qu'il n'y a pas de fonction surjective d'un ensemble sur une classe qui n'est pas un ensemble) :
- Soit (a, r) un ensemble bien ordonné et (B, S) une classe bien ordonnée qui n'est pas un ensemble. Alors (a, r) est isomorphe à un segment initial propre de (B, S).
- Deux classes bien ordonnées, qui ne sont pas des ensembles, sont isomorphes

Ordinaux

D'après le théorème de trichotomie, à isomorphisme près, deux bons ordres sont comparables en termes de segment initial l'un de l'autre. On cherche à définir un représentant canonique pour chaque classe d'isomorphisme de bon ordre.

Définition 3.11 *Un ensemble* α *est transitif si* $\forall x \forall y((x \in y \wedge y \in \alpha) \to x \in \alpha)$. *En d'autres termes, si* $y \in \alpha$, *alors* $y \subseteq \alpha$.

Exemples $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$, $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$

Définition 3.12 α *est un ordinal s'il vérifie la formule suivante*

$$On[\alpha] = (\alpha \textit{ est transitif } \wedge (\alpha, \in_{|\alpha}) \textit{ est un bon ordre}).$$

Proposition 3.13 *Soit α un ordinal.*
- $\alpha \notin \alpha$
- *si $\beta \in \alpha$, alors $\beta \subseteq \alpha$ et β est un ordinal*
- *si $\beta \in \alpha$, alors $\beta = S_\beta(\alpha, \in)$*
- *$\alpha \cup \{\alpha\}$ est un ordinal, c'est le plus petit ordinal γ (au sens de l'inclusion) tel que $\alpha \in \gamma$, on le note $\alpha + 1$*

Proposition 3.14 *Soient deux ordinaux α et β et $f : \alpha \to \beta$ un isomorphisme d'ensembles ordonnés. Alors $\alpha = \beta$ et $f = id$.*

Preuve On montre par induction sur $x \in \alpha$ que $f(x) = x$. Supposons $f(y) = y$ pour tout $y < x$. Pour tout $y \in x$, $y = f(y) \in f(x)$ puisque f est un isomorphisme, ce qui donne $x \subseteq f(x)$. Réciproquement, pour $y \in f(x) \in \beta$, il existe $z \in \alpha$ tel que $y = f(z)$ car f est un isomorphisme, comme $y = f(z) \in f(x)$, $z \in x$ et donc $z = f(z) = y \in x$. D'où l'égalité $x = f(x)$. □

Corollaire 3.15 *(ZF^-) Soient deux ordinaux α et β. Alors $\alpha = \beta$, $\alpha \in \beta$ ou $\beta \in \alpha$ (un seul cas est possible).*

Preuve On utilise le théorème de trichotomie sur les bons ordres. Soit il existe un isomorphisme entre α et β, ce qui implique $\alpha = \beta$ par la proposition précédente. Soit, quitte à échanger α et β, il existe un isomorphisme entre α et un segment initial propre de β. Un tel segment initial est de la forme $\gamma = S_\gamma(\beta, \in)$ pour un certain $\gamma \in \beta$, et la proposition précédente donne $\alpha = \gamma \in \beta$. □

Théorème 3.16 *(ZF^-) La classe des ordinaux, munie de la relation $\in$, est une classe bien ordonnée, qui n'est pas un ensemble.*

Preuve Comme tous les éléments d'un ordinal sont des ordinaux, on a bien que tous les segments initiaux propres sont de la forme $S_\alpha(On, \in) = \alpha$, donc ce sont des ensembles. Et si a est un ensemble non vide d'ordinaux, on choisit $\alpha \in a$ et α_0 le plus petit élément de $a \cap (\alpha + 1)$. En utilisant le fait que l'ordre sur On est total, on voit que α_0 est le plus petit élément de a. La classe $(On, \in)$ est donc bien ordonnée.
Enfin, si $On = a$ est un ensemble, a est transitif et bien ordonné par $\in$, c'est donc un ordinal. On doit donc avoir $a \in a$, ce qui est impossible. □

Remarque La relation d'ordre large sur la classe On est l'inclusion $\subseteq$.
Si a est un ensemble d'ordinaux, a est majoré dans On (car sinon la classe On est contenue dans l'ensemble $\bigcup a$, ce qui est impossible), on peut donc définir $sup\ a$, et on a $sup\ a = \bigcup a$.

Corollaire 3.17 *(ZF^-) Soit (a, r) un ensemble bien ordonné. Alors il existe un unique ordinal α tel que (a, r) soit isomorphe à $(\alpha, \in)$.*

Preuve Comme On est une classe bien ordonnée qui n'est pas un ensemble, on sait qu'il existe un isomorphisme de (a, r) vers un segement initial propre de On, et ce segment initial est un ordinal. L'unicité découle de la proposition

3.14. □

Comme pour tout élément d'une classe bien ordonné, on distingue les ordinaux limites (tels que $\alpha = \cup\alpha$) et les ordinaux successeurs (de la forme $\alpha = \beta \cup \{\beta\}$).

Définition 3.18 *Un ordinal α est dit fini si tous les ordinaux $\beta < \alpha$ sont l'ensemble vide ou des ordinaux successeurs, et infini sinon.*

On peut maintenant énoncer l'axiome de l'infini, que l'on doit ajouter aux théories Z^- et ZF^- pour obtenir les théories Z et ZF.

Axiome de l'infini : il existe un ordinal infini (ou encore : il existe un ordinal limite autre que $\emptyset$).

On note ω le plus petit ordinal infini. Par définition, c'est l'ensemble de tous les ordinaux finis.

3.3 Axiome du choix

Définition 3.19 *Soit a un ensemble. Une fonction de choix sur a est une fonction f de domaine a telle que*

$$\forall x \in a(x \neq \emptyset \rightarrow f(x) \in x)$$

Exemple 3.20

1. *Si a est un ensemble fini, c'est-à-dire en bijection avec un ordinal fini, il existe une fonction de choix sur a. On le démontre par induction sur l'ordinal ω :*
   - *si $a = \emptyset$, $f = \emptyset$ convient*
   - *si a est en bijection avec un ordinal successeur $n + 1 = n \cup \{n\}$, via la bijection $g : n + 1 \rightarrow a$. Soit $x = g(\{n\}) \in a$ et y un élément de x si $x \neq \emptyset$, $y = \emptyset$ si $x = \emptyset$. Clairement, $g_{|n} : n \rightarrow a \setminus \{x\}$ est une bijection, il existe donc par hypothèse d'induction une fonction de choix f sur $a \setminus \{x\}$. On la prolonge en une fonction de choix f' sur a en posant $f'(x) = y$.*
2. *si $\bigcup a$ est un ensemble bien ordonné, alors il existe une fonction de choix f sur a : il suffit de poser, pour $x \in a$, $f(x) =$ le plus petit élément de x si $x \neq \emptyset$, $f(x) = \emptyset$ sinon*

L'axiome du choix s'énonce :
(AC) pour tout ensemble a, il existe une fonction de choix sur a.

On notera : $ZC = Z + AC$ et $ZFC = ZF + AC$.

Remarque 3.21 *Par définition, le produit cartésien $\Pi_{x \in a} x$ est l'ensemble des fonctions totales f de domaine a telles que, pour tout $x \in a$, $f(x) \in x$.*
Donc, si tous les éléments de a sont non vides, une fonction de choix sur a est un élément de $\Pi_{x \in a} x$.
On montre facilement (exercice) que, dans ZF, l'axiome du choix est équivalent à :
(AC') le produit cartésien d'une famille d'ensembles non vides est non vide.

Théorème 3.22 (Théorème de Zermelo) *Dans ZFC, on peut munir tout ensemble d'un bon ordre.*

Preuve Soit a dans U. On se donne une fonction de choix f sur $\mathcal{P}(a) \setminus \{\emptyset\}$. On fixe un élément $e \notin a$, et on définit une fonctionnelle F sur la classe On, par induction :
$F(\alpha)$ = un élément de a qui n'a pas encore été atteint par F pour $\beta < \alpha$ s'il existe, e sinon.
On utilise pour cela le principe de construction par induction sur une classe bien ordonnée, avec la fonctionnelle :

$$H(g) = \begin{cases} f(a \setminus \text{Im } g) & \text{si } a \setminus \text{Im } g \neq \emptyset \\ e \text{ sinon} \end{cases}$$

définie pour g une fonction de domaine un ordinal.
Soit α_0 le plus petit ordinal tel que $F(\alpha_0) = e$ (un tel ordinal existe : sinon F réaliserait une injection de On vers l'ensemble a, et en considérant F^{-1}, on obtiendrait une fonctionnelle surjective d'une sous-classe de a, donc un ensemble, vers On, qui devrait donc être un ensemble par l'axiome de remplacement). Alors $h = F_{|\alpha_0}$ est une fonction (son domaine est un ensemble), injective par construction, d'image a (car sinon $F(\alpha_0) \neq e$). On peut donc définir un ordre r sur a par $x <_r y$ si et seulement si $h^{-1}(x) < h^{-1}(y)$: comme (a, r) est isomorphe à l'ordinal α_0, c'est un bon ordre. □

Proposition 3.23 (ZF) *L'axiome du choix est équivalent à : tout ensemble est bien ordonnable.*

Preuve Une direction est le théorème de Zermelo, l'autre est la remarque selon laquelle si $\bigcup a$ est bien ordonnable, il existe une fonction de choix sur a. □

Définition 3.24 *Un ensemble ordonné (a, r) est dit inductif si tout sous-ensemble totalement ordonné de a a un majorant dans a (en particulier $a \neq \emptyset$).*

Théorème 3.25 (ZFC, Lemme de Zorn) *Tout ensemble inductif a un élément maximal.*

Preuve Soit (a, r) ensemble ordonné inductif, $e \notin a$ et f une fonction de choix sur $\mathcal{P}(a) \setminus \{\emptyset\}$, prolongée en $f(\emptyset) = e$. On définit une fonctionnelle F par induction sur la classe des ordinaux, via la fonctionnelle :

$$H(g) = f(\{x \in a, \forall \beta \in \text{Dom}(g), x >_r g(\beta)\}).$$

Soit α_0 le plus petit ordinal tel que $F(\alpha_0) = e$ (il existe par le schéma de remplacement). La fonction $F_{|\alpha_0}$ est injective par construction, et $\{F(\beta); \beta \in \alpha_0\}$ est un ensemble (schéma de remplacement), totalement ordonné par construction (si $\beta < \alpha, F[\beta] < F[\alpha]$), donc il possède un majorant $m \in a$. Alors m est un élément maximal de (a, r), car sinon il existerait $y >_r m$, d'où $F(\alpha_0) \neq e$. □

Proposition 3.26 (ZF) *L'axiome du choix est équivalent à l'énoncé du lemme de Zorn.*

Preuve Il reste à montrer l'axiome du choix en supposant vrai le lemme de Zorn.
Soit a un ensemble (on peut supposer $\bigcup a \neq \emptyset$), on considère l'ensemble

$$b = \{f \in \mathcal{P}(a \times \bigcup a); f \text{ est une fonction et } \forall x \in \mathrm{Dom}(f)(x \neq \emptyset \to f(x) \in x)\},$$

ordonné par l'inclusion.
Pour $c \subseteq b$ un sous-ensemble totalement ordonné, c admet un majorant dans b : $\bigcup_{f \in c} f$ (c'est bien une fonction, et un élément de b).
Par le lemme de Zorn, soit h un élément maximal de b ; il suffit de vérifier que $\mathrm{Dom}(h) = a$ pour montrer que h est une fonction de choix sur a. Or, si $x \in a \setminus \mathrm{Dom}(h)$, on peut considérer $y \in x$ si $x \neq \emptyset$, ou $y \in \bigcup a$ sinon, et alors $h \cup \{(x, y)\}$ est un élément de b qui est un majorant strict de h : contradiction. □

3.4 Cardinaux

Définition 3.27 *Soient a et b deux éléments de $\mathcal{U}$.*
On dit que a et b sont équipotents, et on note $a \sim b$, s'il existe une bijection de a dans b.
On dit que a est subpotent à b, et on note $a \preccurlyeq b$, s'il existe une injection de a dans b.

Remarque 3.28 *La relation $\sim$ est une relation d'équivalence.*
La relation $\preccurlyeq$ est réflexive et transitive.

Théorème 3.29 (Théorème de Cantor-Bernstein) *Dans ZF, si $a \preccurlyeq b$ et $b \preccurlyeq a$, alors $a \sim b$.*

Preuve Soit deux injections $f : a \to b$ et $g : b \to a$. Pour $x_0 \in a$, on construit une suite par induction sur ω, avec $x_{2i} \in a$ et $x_{2i+1} \in b$:
si $x_{2i} \in \mathrm{Im}(g)$, on pose pour x_{2i+1} l'unique antécédent de x_{2i} par g, sinon la suite s'arrête ; si $x_{2i+1} \in \mathrm{Im}(f)$, on pose x_{2i+2} l'unique antécédent de x_{2i+1} par f, sinon la suite s'arrête.
On écrit une partition de a : $x_0 \in a^\infty$ si on peut construire une telle suite infinie à partir de x_0, $x_0 \in a^a$ si cette suite s'arrête du côté de a, et $x_0 \in a^b$ si elle s'arrête du côté de b.
De même, à partir de $x_0 \in b$, on construit une suite par la même méthode, et on pose $x_0 \in b^\infty, b^a$ ou b^b selon que la suite est infinie, s'arrête du côté de a, ou s'arrête du côté de b.
On définit alors $h : a \to b$ par :

$$h(x) = \begin{cases} f(x) \text{ si } x \in a^\infty \\ f(x) \text{ si } x \in a^a \\ g^{-1}(x) \text{ si } x \in a^b \end{cases}$$

On vérifie aisément que la restriction de h à a^∞, a^a et a^b est une bijection vers b^∞, b^a et b^b respectivement.
D'où $a \sim b$. □
On veut définir le cardinal d'un ensemble comme un représentant particulier de sa classe d'équivalence pour la relation d'équipotence. On utilise pour cela le théorème de Zermelo.

Définition 3.30 *On se place dans ZFC. Pour tout élément a de $\mathcal{U}$, il existe un ordinal équipotent à a, on définit le cardinal de a comme étant le plus petit ordinal équipotent à a, et on le note $card(a)$.*
On appelle cardinal un ordinal qui n'est équipotent à aucun ordinal qui lui est strictement inférieur, et on note Cn la classe des cardinaux.

Remarque Les définitions précédentes sont cohérentes entre elles, au sens où :
- si α est un cardinal, $\alpha = card(\alpha)$
- pour tout ensemble a, $card(a)$ est un cardinal

Exemple 3.31
1. *Les ordinaux finis sont des cardinaux (on montre par induction sur ω qu'un ordinal fini n'est pas en bijection avec une de ses parties propres).*
2. *ω est un cardinal : en effet, si $f : \omega \to n$ est une bijection pour un certain ordinal fini n, alors $f_{\restriction n} : n \to a$ est une bijection pour une partie propre a de n*
3. *$\omega + 1$ n'est pas un cardinal, à cause de la bijection $\omega + 1 \to \omega$, $\omega \mapsto 0$, $n \mapsto n + 1$ pour tout ordinal fini n.*

Proposition 3.32 *(Dans ZFC). Soient a et b deux éléments de $\mathcal{U}$, distincts de $\emptyset$. Les propriétés suivantes sont équivalentes :*
1. *il existe une injection de a dans b*
2. *il existe une surjection de b dans a*
3. *$card(a) \leq card(b)$*

Preuve $1 \Rightarrow 2$ Si $f : a \to b$ est une injection, et y un élément fixé de a, on pose pour $x \in b$, $g(x)$ l'unique antécédent de x par f si $x \in \mathrm{Im}(b)$, y sinon. Alors g est une surjection de b dans a
$2 \Rightarrow 1$ Soit f une surjection de b dans a, et h une fonction de choix sur $\mathcal{P}(b)\setminus\{\emptyset\}$. On pose pour $x \in a$, $g(x) = h(\{y \in b; f(y) = x\})$, c'est une injection de a dans b
$3 \Rightarrow 1$ Evident car l'inclusion est une injection de $\mathrm{card}(a)$ dans $\mathrm{card}(b)$
$1 \Rightarrow 3$ Après application de la bijection entre un ensemble et son cardinal, il suffit de vérifier que si $f : \lambda \to \mu$ est une injection, avec λ et μ des cardinaux, alors $\lambda \leq \mu$. L'image de f est bien ordonnée par $\in$, elle est donc isomorphe à un ordinal α. On a donc une fonction strictement croissante $g : \alpha \to \mathrm{Im}(f) \subseteq \mu$; on a déjà vu que cela implique que $g(x) \geq x$ pour tout $x \in \alpha$, et donc, en prenant les bornes supérieures, $\alpha \leq \mu$. Comme λ est en bijection avec $\mathrm{Im}(f)$, et donc avec α, et que λ est un cardinal, on obtient $\lambda \leq \alpha \leq \mu$. □

Remarque 3.33 *Le théorème de Cantor-Bernstein s'obtient directement comme conséquence de* $1 \Rightarrow 3$. *Mais on l'a démontré précédemment sans avoir recours à l'axiome du choix.*

Théorème 3.34 (argument diagonal de Cantor) *(ZF) Soit* a *un élément de* $\mathcal{U}$. *Il n'existe pas de surjection de* a *dans* $\mathcal{P}(a)$.

Preuve Supposons qu'une telle surjection $f : a \to \mathcal{P}(a)$ existe. On considère l'ensemble $b = \{x \in a; x \notin f(x)\} \in \mathcal{P}(a)$. Comme f est surjective, il existe $c \in a$ tel que $f(c) = b$. Mais alors $c \in b$ si et seulement si $c \notin f(c)$ si et seulement si $c \notin b$: contradiction. □

Corollaire 3.35 *(ZFC) Pour tout cardinal* λ, *il existe un cardinal* μ *tel que* $\lambda < \mu$.

Preuve D'après ce qui précède, $\text{card}(\mathcal{P}(\lambda)) > \text{card}(\lambda) = \lambda$. □

Définition 3.36 *On note* λ^+ *le plus petit cardinal supérieur à* λ.

On vient de voir que $\text{card}(\mathcal{P}(\lambda)) \geq \lambda^+$. L'hypothèse du continu (HC) est l'énoncé : $\text{card}(\mathcal{P}(\omega)) = \omega^+$. En utilisant le fait que $\mathbb{R}$ est en bijection avec $\mathcal{P}(\omega)$ (exercice), cela revient à dire que tout sous-ensemble infini de $\mathbb{R}$ est soit dénombrable, soit équipotent à $\mathbb{R}$. Il a été montré, par Gödel et Cohen, que cet énoncé est indépendant de ZFC : si ZFC est consistante, elle ne démontre ni HC, ni $\neg HC$.

Arithmétique cardinale

Proposition 3.37 *Soient des éléments* a, b, c, d *de* $\mathcal{U}$. *On suppose que* $a \sim b$ *et* $c \sim d$. *Alors* $a \dot{\cup} c \sim b \dot{\cup} d$, $a \times c \sim b \times d$, $a^c \sim b^d$.

Preuve Evident. □

Définition 3.38 *Pour deux cardianux* λ *et* μ, *on définit :*

- $\lambda + \mu = card(\lambda \dot{\cup} \mu)$
- $\lambda \cdot \mu = card(\lambda \times \mu)$
- $\lambda^\mu = card(\lambda^\mu)$

Fait 3.39
- *L'ensemble des cardianux finis est stable par addition, multiplication, puissance.*
- *Commutativité et associativité de* $+$ *et* $\cdot$
- *Distributivité de* $\cdot$ *sur* $+$
- $(\lambda \cdot \mu)^\nu = \lambda^\nu \cdot \mu^\nu$, $\lambda^{\mu+\nu} = \lambda^\mu \cdot \lambda^\nu$, $(\lambda^\mu)^\nu = \lambda^{\mu \cdot \nu}$

Théorème 3.40 (Trivialisation de l'arithmétique cardinale infinie) *Soit* λ *un cardinal infini, alors* $\lambda + \lambda = \lambda \cdot \lambda = \lambda$.

Preuve Comme $\lambda \sim \lambda \times \{0\} \subseteq \lambda \dot{\cup} \lambda = \lambda \times \{0,1\} \subseteq \lambda \times \lambda$, il suffit de montrer que $\lambda \cdot \lambda = \lambda$.
On définit l'ordre de Gödel pour $(\alpha, \beta) \in \lambda \times \lambda$ comme l'ordre lexicographique sur $(\max(\alpha,\beta), \beta, \alpha)$: $(\alpha,\beta) <_R (\alpha',\beta')$ si et seulement si $\max(\alpha,\beta) < \max(\alpha',\beta')$ ou $(\max(\alpha,\beta) = \max(\alpha',\beta')$ et $\beta < \beta')$ ou $(\max(\alpha,\beta) = \max(\alpha',\beta')$ et $\beta = \beta'$ et $\alpha < \alpha')$.
On montre facilement que $<_R$ est un bon ordre sur $\lambda \times \lambda$, et on note γ l'unique ordinal isomorphe à $(\lambda \times \lambda, <_R)$. On a $\lambda = card(\lambda) \leq card(\gamma) = \lambda \cdot \lambda \leq \gamma$.
Supposons par l'absurde que λ soit le plus petit cardinal infini tel que $\lambda \cdot \lambda > \lambda$. Soit f l'isomorphisme de γ dans $(\lambda \times \lambda, <_R)$, on a $\lambda < \gamma$ et $\mathrm{Im}(f_{|\lambda}) = \{(\alpha,\beta) \in \lambda \times \lambda; (\alpha,\beta) <_R f(\lambda)\}$. Posons $f(\lambda) = (\alpha_0, \beta_0)$ et $\delta_0 = \max(\alpha_0, \beta_0)$; on a $\delta_0 < \lambda$ donc $card(\delta_0) < \lambda$ (car λ est un cardinal). On a une injection $f_{|\lambda} : \lambda \to (\delta_0 + 1 \times \delta_0 + 1, <_R)$, et donc $card(\delta_0) < \lambda \leq card(\delta_0 \times \delta_0)$ (car δ_0 est infini, et donc $\delta_0 \sim \delta_0 + 1$), ce qui contredit la minimalité de λ. □

Corollaire 3.41 *Soient λ et μ deux cardinaux non nuls, dont l'un au moins est infini. Alors $\lambda + \mu = \lambda \cdot \mu = max(\lambda, \mu)$.*

Corollaire 3.42 *Soit S un ensemble de cardinal λ infini, et $S^{<\omega}$ l'ensemble des suites finies de S (c'est-à-dire l'ensemble des fonctions de domaine un élément $n \in \omega$, à valeurs dans S). Alors $card(S^{<\omega}) = \lambda$.*

3.5 Une utilisation des cardinaux : le théorème de Löwenheim-Skolem généralisé

Il s'agit d'un théorème portant sur le calcul des prédicats. Pour un langage $\mathcal{L} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, on appelle cardinal de $\mathcal{L}$ le cardinal $card(\mathcal{R}) + card(\mathcal{F}) + card(\mathcal{C})$.

Théorème 3.43 (Théorème de Löwenheim-Skolem (ZFC)) *Soit T une théorie consistante, admettant des modèles infinis, écrite dans un langage $\mathcal{L}$ de cardinal κ. Alors pour tout cardinal infini $\lambda \geq \kappa$, T admet au moins un modèle de cardinal λ.*

Lemme 3.44 (Théorème de Löwenheim-Skolem descendant) *Soit T une théorie consistante dans un langage $\mathcal{L}$ de cardinal κ. Alors T admet un modèle de cardinal $\leq max(\kappa, \omega)$.*

Preuve On reprend la preuve du théorème de complétude pour la théorie cohérente T. On étend T en une théorie T' dans un langage $\mathcal{L}'$, de manière à ce que T' admette des témoins de Henkin. Pour cela, on construit $\mathcal{L}' = \bigcup_{i \in \omega} \mathcal{L}_i$, avec $\mathcal{L}_0 = \mathcal{L}$ et $\mathcal{L}_{i+1} = \mathcal{L}_i \cup \{c_F; F[x] \text{ formule dans le langage } L_i\}$. Or l'ensemble des formules à une variable libre du langage $\mathcal{L}_i$ est un sous-ensemble de l'ensemble des suites finies de symboles parmi $\mathcal{S}$, constitué de l'ensemble $\mathcal{L}_i$ et des symboles communs à tous les langages. Et l'ensemble de ces suites finies s'écrit $\bigcup_{n \in \omega} \mathcal{S}^n$, avec $card(\mathcal{S}^n) \leq max(card(\mathcal{S}), \omega)$ par une récurrence facile sur $n \in \omega$, d'où $card(\bigcup_{n \in \omega} \mathcal{S}^n) \leq max(card(\mathcal{S}), \omega) \cdot \omega = max(card(\mathcal{S}), \omega)$. Cela nous permet de conclure que $\mathcal{L}'$ est de cardinal $\leq max(\kappa, \omega)$.
Ensuite, on construit un modèle $\mathcal{M}$ de T' d'ensemble de base l'ensemble des

termes sans variable de $\mathcal{L}'$, quotienté par une certaine relation d'équivalence. Par un argument similaire au précédent, l'ensemble de ces termes sans variable est de cardinal $\leq max(\kappa, \omega)$. Par la fonction surjective qui à un terme associe son représentant, on trouve bien $card(M) \leq max(\kappa, \omega)$. □

Preuve du théorème On considère le langage $\mathcal{L}' = \mathcal{L} \cup \mathcal{C}'$, où $\mathcal{C}'$ est un $\mathcal{C}'$ est un ensemble de symbole de constantes, de cardinal λ, disjoint de $\mathcal{L}$. Notons que $card(\mathcal{L}') = \kappa + \lambda = \lambda$. On regarde dans ce langage la théorie $T' = T \cup \{\neg c \equiv d; c, d \in \mathcal{C}', c \neq d\}$. On constate facilement, en utilsant le théorème de compacité et le fait que T admette des modèles infinis, que T est consistante. Par le lemme précédent, T' admet un modèle $\mathcal{M}$ de cardinal $\leq \lambda$. Mais comme $\mathcal{M}$ est un modèle de T', la fonction $\mathcal{C}' \to M$, $c \mapsto \overline{c}$ est injective. Donc $card(M) = \lambda$. □

Exemple d'application Soit K un corps et T la théorie des espaces vectoriels infinis sur K, dans le langage $(0, +, (f_q)_{q \in K})$ (les f_q sont des fonctions unaires interprétées comme la multiplication par q dans l'espace vectoriel). Montrer que la théorie T est complète.

Chapitre 4

Récursivité

Objectif : donner des définitions précises pour la notion de fonctions calculables « mécaniquement », ou encore par algorithme. Les fonctions considérées seront des fonctions de $\mathbb{N}^p$ dans $\mathbb{N}$, pour p quelconque.

4.1 Fonctions primitives récursives

Définition 4.1 *L'ensemble des fonctions primitives récursives est le plus petit sous-ensemble de* $\bigcup_{p\geq 1} \mathbb{N}^{\mathbb{N}^p}$ *tel que :*

1. *La fonction nulle* $\underline{0} = (x \mapsto 0)$ *est primitive récursive.*
   La fonction successeur $s = (x \mapsto x+1)$ *est primitive récursive.*
   Les projections $p_i^n = ((x_1, \ldots, x_n) \mapsto x_i)$*, pour* $1 \leq i \leq n$*, sont primitives récursives.*
2. *Composition :*
   Si $h : \mathbb{N}^k \to \mathbb{N}$ *et* $g_1, \ldots, g_k : \mathbb{N}^p \to \mathbb{N}$ *sont primitives récursives, la fonction* $h \circ (g_1, \ldots, g_k) = ((x_1, \ldots, x_p) \mapsto h(g_1(x_1, \ldots, x_p), \ldots, g_k(x_1, \ldots, x_p)))$ *est primitive récursive.*
3. *Récurrence primitive :*
   Si $g : \mathbb{N}^p \to \mathbb{N}$ *et* $h : \mathbb{N}^{p+2} \to \mathbb{N}$ *sont primitives récursives, la fonction* $f : \mathbb{N}^{p+1} \to \mathbb{N}$ *définie par*

$$\begin{cases} f(x_1, \ldots, x_p, 0) = g(x_1, \ldots, x_p) \\ f(x_1, \ldots, x_p, x+1) = h(x_1, \ldots, x_p, x, f(x_1, \ldots, x_p, x)) \end{cases}$$

   est primitive récursive (on note $f = R(g, h)$*).*

Exemples de base

1. Les fonctions constantes $(x_1, \ldots, x_p) \mapsto a$, pour $a \in \mathbb{N}$, sont primitives récursives.
   On le prouve par induction sur a :
   $\underline{0}$ est primitive récursive
   si $(x \mapsto a)$ est primitive récursive, alors $(x \mapsto a+1) = s \circ (x \mapsto a)$ aussi.
   Puis, pour les fonctions d'arité p quelconque, $((x_1, \ldots, x_p) \mapsto a) = (x \mapsto a) \circ p_1^n$ est primitive récursive.

2. La fonction identité $\mathbb{N} \to \mathbb{N}$ est primitive récursive : c'est la projection p_1^1.
3. L'addition $+ : \mathbb{N}^2 \to \mathbb{N}$ est primitive récursive, elle s'écrit $+ = R(g,h)$ avec $g = p_1^1$ et $h = s \circ p_3^3$.
4. La multiplication $\cdot : \mathbb{N}^2 \to \mathbb{N}$ est primitive récursive, elle s'écrit $\cdot = R(g,h)$ avec $g = \underline{0}$ et $h = + \circ (p_1^3, p_3^3)$ (c'est-à-dire $h(x,y,x \cdot y) = x + x \cdot y$).
5. L'exponentielle $exp = ((x,y) \mapsto x^y)$ est primitive récursive, elle s'écrit $exp = R(g,h)$ avec $g = (x \mapsto 1)$ et $h = \cdot \circ (p_1^3, p_3^3)$
6. Si $g, h : \mathbb{N}^p \to \mathbb{N}$ sont primitives récursives, alors la fonction f définie par cas
$$\begin{cases} (x_1, \ldots, x_p, 0) \mapsto g(x_1, \ldots, x_p) \\ (x_1, \ldots, x_p, x+1) \mapsto h(x_1, \ldots, x_p) \end{cases}$$
est primitive récursive, elle s'écrit $R(g, h \circ (p_1^{p+2}, \ldots, p_p^{p+2}))$
7. En particulier, les fonction $sg = \begin{cases} 0 \mapsto 0 \\ n+1 \mapsto 1 \end{cases}$ et $\overline{sg} = \begin{cases} 0 \mapsto 1 \\ n+1 \mapsto 0 \end{cases}$ sont primitives récursives.
8. La fonction $pred = \begin{cases} 0 \mapsto 0 \\ x+1 \mapsto x \end{cases}$ est primitive récursive, elle s'écrit $R(g,h)$ pour $g = 0$ et $h = p_1^2$.
9. La soustraction tronquée $\dot{-} : \mathbb{N}^2 \to \mathbb{N}$, telle que $x \dot{-} y = x - y$ si $x \geq y$, et 0 sinon, est primitive récursive. Elle s'écrit $\dot{-} = R(g,h)$ pour $g = p_1^1$ et $h = pred \circ p_1^3$.
10. La fonction $((x,y) \mapsto |\, x - y \,|) = + \circ (\dot{-}, \dot{-} \circ (p_2^2, p_1^2))$ est primitive récursive
11. Somme et produit bornés : si $f : \mathbb{N}^{p+1} \to \mathbb{N}$ est primitive récursive, alors $F = ((x_1, \ldots, x_p, n) \mapsto \sum_{i=0}^{n} f(x_1, \ldots, x_p, i))$ et $G = ((x_1, \ldots, x_p, n) \mapsto \Pi_{i=0}^{n} f(x_1, \ldots, x_p, i))$ sont primitives récursives : $F = R(f \circ (p_1^p, \ldots, p_p^p, 0), + \circ (p_p^{p+2}, f \circ (p_1^{p+2}, \ldots, p_p^{p+2}, s \circ p_{p+1}^{p+2})))$ et $G = R(f \circ (p_1^p, \ldots, p_p^p, 0), \cdot \circ (p_p^{p+2}, f \circ (p_1^{p+2}, \ldots, p_p^{p+2}, s \circ p_{p+1}^{p+2})))$.

Définition 4.2 *Un prédicat (ou un ensemble) primitif récursif est un prédicat (ou un ensemble) dont la fonction caractéristique est primitive récursive.*

Exemple 4.3
- *Un singleton $\{a\}$ est primitif récursif :* $\mathbf{1}_{\{a\}} = (x \mapsto \overline{sg}(|\, x - a \,|)$
- *L'ensemble $D = \{(x,y) \in \mathbb{N}^2; x \leq y\}$ est primitif récursif :* $\mathbf{1}_D = ((x,y) \mapsto \overline{sg}(x \dot{-} y))$.

Proposition 4.4 *La collection des ensembles primitifs récursifs est stable par union finie, intersection finie et complémentaire. En particulier, les ensembles finis et cofinis (c'est-à-dire de complémentaire fini) sont primitifs récursifs.*

Preuve Il suffit de le montrer pour les opérations de complémentaire et d'intersection de deux ensembles.
On a : $\mathbf{1}_{X^c} = \overline{sg} \circ \mathbf{1}_X$ et $\mathbf{1}_{X \cap Y} = \cdot \circ (\mathbf{1}_X, \mathbf{1}_Y)$. □

Proposition 4.5 *La collection des prédicats primitifs récursifs est stable par quantification bornée : si $P(x_1, \ldots, x_p, x)$ est un prédicat primitif récursif, alors $\exists t \leq x, P(x_1, \ldots, x_p, t)$ et $\forall t \leq x, P(x_1, \ldots, x_p, t)$ le sont aussi.*

Preuve Pour la quantification universelle, il suffit de considérer la fonction

$$\Pi_{t=0}^{x} \mathbf{1}_{P(x_1,\ldots,x_p,t)}.$$

Pour la quantification existentielle, $g(x_1, \ldots, x_p, x) = \mathbf{1}_{\exists t \leq x, P(x_1,\ldots,x_p,t)}$ est définie par un schéma de récurrence primitive :

$$\begin{cases} g(x_1, \ldots, x_p, 0) = \mathbf{1}_{P(x_1,\ldots,x_p,0)} \\ g(x_1, \ldots, x_p, x+1) = sg(g(x_1, \ldots, x_p, x) + \mathbf{1}_{P(x_1,\ldots,x_p,x+1)}) \end{cases}$$

□

Proposition et définition 4.6 *Soit $P(x_1, \ldots, x_p, t)$ un prédicat primitif récursif. On considère la fonction obtenue par minimisation bornée*

$$(x_1, \ldots, x_p, x) \mapsto \mu t \leq x, P(x_1, \ldots, x_p, t) = \begin{cases} \text{le plus petit entier } t \leq x \text{ tel que } P(x_1, \ldots, x_p, t) \text{ s'il existe} \\ 0 \text{ sinon} \end{cases}.$$

C'est une fonction primitive récursive.

Preuve Notons f cette fonction. On reprend la fonction g de la preuve précédente donnant la quantification existentielle bornée, et on a l'égalité

$$f(x_1, \ldots, x_p, x) = g(x_1, \ldots, x_p, x) \cdot \sum_{t=0}^{x} \overline{sg}(g(x_1, \ldots, x_p, t)),$$

ce qui nous montre que f est primitive récursive. □

Exemple 4.7

1. *Le prédicat « x divise y » est primitif récursif, donné par : $\exists t \leq y, x.t = y$.*
2. *Le prédicat x est premier est primitif récursif, donné par : $x \geq 2 \wedge \forall t \leq x, (\text{« } t \text{ divise } x \text{ »} \Rightarrow (t = 1 \vee t = x))$.*
3. *La fonction $(n \mapsto \pi(n))$, où $\pi(n)$ est le n-ième nombre premier est primitive récursive, donnée par récurrence primitive et minimisation bornée :*

$$\pi(n+1) = \mu x \leq \pi(n)! + 1, (x > \pi(n) \wedge \pi(n) \text{ est premier}).$$

4.2 Fonctions récursives partielles

Certaines fonctions sont calculables intuitivement, sans être primitives récursives. On introduit donc une classe plus large de fonctions. Pour une fonction partielle $f : \mathbb{N}^p \to \mathbb{N}$, on note $f(x_1, \ldots, x_p) \downarrow$ si f est définie en $(x_1, \ldots, x_p)$, $f(x_1, \ldots, x_p) \uparrow$ sinon.

Définition 4.8 *L'ensemble des fonctions récursives partielles est le plus petit sous-ensemble de fonctions partielles $\mathbb{N}^p \to \mathbb{N}$, pour $p \geq 1$, tel que :*

1. *La fonction nulle, la fonction successeur et les projections sont récursives partielles.*
2. *La composée de fonctions récursives partielles est récursive partielle (attention en un point quelconque tous les g_i doivent être définies pour que $h \circ (g_1, \ldots, g_k)$ le soit).*
3. *Si $f = R(g, h)$ est obtenue par réccurence primitive à partir de fonctions récursives partielles g et h, alors f est récursive partielle (la définition sous-entend en particulier qu'il faut que $f(x_1, \ldots, x_p, t) \downarrow$ pour tout $t \leq x$ pouyr que $f(x_1, \ldots, x_p, x+1) \downarrow$).*
4. *Clôture par minimisation :*
   si $g : \mathbb{N}^{p+1} \to \mathbb{N}$ est une fonction récursive partielle, alors la fonction $f = (x_1, \ldots, x_p) \mapsto \mu x, g(x_1, \ldots, x_p, x) = 0$, définie par $f(x_1, \ldots, x_p) = y$ si $g(x_1, \ldots, x_p, y) = 0$ et pour tout $z < y$, $g(x_1, \ldots, x_p, z)$ est défini et non nul, et $f(x_1, \ldots, x_p) \uparrow$ sinon, est récursive partielle.

Une fonction récursive est une fonction récursive partielle qui est partout définie. Un prédicat est récursif si sa fontion caractéristique est récursive.

Exemple 4.9
1. *Les fonctions primitives récursives sont récursives.*
2. *La fonction nulle part définie est récursive partielle : elle est donnée par $(x_1, \ldots, x_p) \mapsto \mu x, 1 = 0$.*
3. *Si $f : \mathbb{N} \to \mathbb{N}$ est une fonction bijective récursive, son inverse est récursive : elle est donnée par $f^{-1}(y) = \mu x, f(x) = y$.*

Remarque 4.10 *Il existe des fonctions récursives (totales) qui ne sont pas récursives primitives : on définit la fonction d'Ackerman A par*

$$\begin{cases} A(0, y) = 2^y \\ A(x, 0) = 1 \\ A(x+1, y+1) = A(x, A(x+1, y)) \end{cases}.$$

On constate qu'il existe un moyen effectif de calculer cette fonction, par induction sur les couples (x, y) rangés par le bon ordre lexicographique ; mais la récurrence qui intervient n'est pas primitive. On montre que cette fonction A est récursive mais pas primitive récursive (voir exercices).

4.3 Machines de Turing

La raison pour laquelle on s'intéresse aux fonctions récursives est donnée par l'affirmation suivante, connue sous le nom de « thèse de Church » :
les fonctions calculables sont exactement les fonctions récursives.
Cette thèse ne peut pas être démontrée, faute d'une définition précise pour le concept « calculable ». On la considère toutefois comme valide, du fait de l'observation empirique suivante : à chaque fois qu'on essaie de se donner une définition raisonnable de fonction calculable, on obtient toujours la classe des fonctions récursives.
En guise d'illustration, on introduit ici les fonctions calculables par machines de Turing, qui sont des abstractions de nos ordinateurs.

Définition 4.11 *Une machine de Turing est la donnée de :*
- *un nombre fini de bandes $B_1, \ldots, B_p$. Chaque bande est bornée à gauche et infinie à droite ; pour $x \in \mathbb{N}$, $B_i(x)$ sera le symbole écrit sur la case x de la bande i. Les symboles possibles sont d (pour début), la barre $|$, et $\square$ pour blanc (on note $S = \{d, |, \square\}$).*
- *un ensemble fini d'états E, parmi lesquels se trouveront toujours l'état initial e_i et l'état final e_f.*
- *une table de transition qui est une application $M : E \times S^p \to E \times S^p \times \{-1, 0, +1\}$.*

Le fonctionnement d'une machine de Turing est décrit comme suit :
- *à l'instant initial, la tête de lecture/écriture se trouve en $x = 0$, les bandes contiennent les données de départ (on aura toujours $B_i(0) = d$, et $B_i(y) = \square$ pour y suffisament grand), et la machine est dans l'état e_i*
- *à chaque instant, si la machine est dans l'état e et si la tête se trouve en $x \in \mathbb{N}$, on regarde $M(e, (B_1(x), \ldots, B_p(x))) = (e', (s_1, \ldots, s_p), \epsilon)$. Alors on assigne les nouvelles valeurs $B_i(x) = s_i$ pour $1 \leq i \leq p$, l'état de la machine devient e', et la tête est déplacée en $x + \epsilon$*
- *la machine s'arrête dès qu'elle est dans l'état e_f*

On supposera toujours dans la suite que la table de transition M vérifie les conditions pour un fonctionnement normal de la machine (on ne doit jamais écrire autre chose que le symbole d sur la case 0 des bandes, et la tête ne peut jamais être positionnée en $x < 0$).

Représentation des entiers et des fonctions
On dira que l'entier n est représenté sur la bande i si les symboles sur cette bande sont :

$$d \underbrace{| \, | \ldots |}_{n \text{ fois}} \square \ldots$$

Une machine de Turing à $p + 1$ bandes principales (plus éventuellement des bandes auxiliaires), donnée par un ensemble d'état E et une table de transition M, calcule une fonction partielle $f : \mathbb{N}^p \to \mathbb{N}$ de la manière suivante :
on a à l'instant initial les entiers $n_1, \ldots, n_p$ représentés sur les bandes $B_1, \ldots, B_p$, et 0 sur la bande B_{p+1} et les bandes auxiliaires, et on lance le fonctionnement de la machine. Si elle n'atteint jamais l'état e_f, alors $f(n_1, \ldots, n_p) \uparrow$. Si elle atteint l'état e_f, et si un entier m est représenté à cet instant sur la bande B_{p+1}, alors on pose $f(n_1, \ldots, n_p) = m$.

Exemple 4.12 *La fonction successeur $(n \mapsto n+1)$ est calculable par la machine de Turing suivante :*
c'est une machine à deux bandes B_1 et B_2, et deux états e_i, e_f, avec une table de transition donnée par :
- $(e_i, (d, d)) \mapsto (e_i, (d, d), +1)$
- $(e_i, (|, s)) \mapsto (e_i, (|, |), +1)$*, où s est un symbole quelconque*
- $(e_i, (\square, s)) \mapsto (e_f, (\square, |), 0)$*, où s est un symbole quelconque*

(les autres transitions n'ont pas d'importance).

Cet exemple n'est qu'un cas particulier du théorème suivant, illustration de la thèse de Church.

Théorème 4.13 *La classe des fonctions qui sont calculables par une machine de Turing est exactement la classe des fonctions récursives partielles.*

Preuve Les fonctions récursives partielles sont calculables par machine de Turing.
Il suffit de montrer qu'on peut calculer par machine de Turing les fonctions de base, et qu'on peut « programmer » les opérations de composition, de récurrence primitive et de minimisation. Quitte à ajouter quelques états après l'état final d'une machine, on supposera toujours que celle-ci revient en début de bande à la fin de son exécution, en effaçant les bandes auxiliaires.
La fonction nulle est calculable par machine de Turing, il suffit de passer directement dans l'état final, sachant que 0 est toujours représenté sur la bande de sortie au départ.
On vient de voir dans l'exemple précédent que la fonction successeur est calculable par machine de Turing.
La fonction projection p_i^p est calculable par machine de Turing : il suffit de recopier le contenu de la bande B_i sur la bande B_{p+1} en suivant ce qui a déjà été fait pour la fonction successeur.
On suppose que les fonctions $g_1, \ldots, g_k : \mathbb{N}^p \to \mathbb{N}$ et $h : \mathbb{N}^k \to \mathbb{N}$ sont calculables par des machines de Turing $\mathcal{M}_1, \ldots, \mathcal{M}_k$ et $\mathcal{M}$. Pour calculer la fonction composée $f = h \circ (g_1, \ldots, g_k)$, on lance la machine $\mathcal{M}_1$ avec les données des bandes $B_1, \ldots, B_p$ et on écrit le résultat sur une nouvelle bande auxiliaire B_{p+2}, puis $\mathcal{M}_2$ (techniquement, l'état initial de notre nouvelle machine est celui de $\mathcal{M}_1$, et on identifie l'état final de $\mathcal{M}_1$ et l'état initial de $\mathcal{M}_2$), jusqu'à $\mathcal{M}_k : (B_1, \ldots, B_p) \to B_{p+k+1}$. Enfin, on lance $\mathcal{M} : (B_{p+2}, \ldots, B_{p+k+1}) \to B_{p+1}$ (cette notation signifie que $B_{p+2}, \ldots, B_{p+k+1}$ sont les bandes de données et B_{p+1} la bande de sortie).
On suppose que $g : \mathbb{N}^p \to \mathbb{N}$ est calculable par une machine $\mathcal{M}$ et $h : \mathbb{N}^{p+2} \to \mathbb{N}$ est calculable par une machine $\mathcal{N}$, et on veut calculer la fonction $f = R(g, h) : \mathbb{N}^{p+1} \to \mathbb{N}$ obtenue par récurrence primitive. On introduit deux nouvelles bandes auxiliaires : B_{p+3} pour le compteur (qui devra varier de 0 à x_{p+1} pour calculer $f(x_1, \ldots, x_{p+1})$) et B_{p+4} pour les résultats intermédiaires $f(x_1, \ldots, x_p, x)$ pour $x < x_{p+1}$. On lance la machine $\mathcal{M} : (B_1, \ldots, B_p) \to B_{p+2}$, et on entre dans le nouvel état e. On compare le contenu des bandes B_{p+1} et B_{p+3} (il suffit de parcourir les bandes depuis le début et de voir si on obtient simultanément le premier $\square$). En cas d'égalité, on passe à l'état final et la bande B_{p+2} contient le résultat voulu. Sinon, on copie le contenu de B_{p+2} sur B_{p+4}, on lance la machine $\mathcal{N} : (B_1, \ldots, B_p, B_{p+3}, B_{p+4}) \to B_{p+2}$, on incrémente le contenu de la bande B_{p+3} et on retourne à l'état e.
On suppose que $g : \mathbb{N}^{p+1} \to \mathbb{N}$ est calculable par une machine de Turing $\mathcal{M}$, et on veut calculer la fonction $f : \mathbb{N}^p \to \mathbb{N}, (x_1, \ldots, x_p) \to \mu x, g(x_1, \ldots, x_p, x) = 0$. Les bandes de données seront les bandes $B_1, \ldots, B_p$, celle de sortie B_{p+1} (qui doit représenter 0 par convention au début du fonctionnement), et on aura aussi une nouvelle bande auxiliaire B_{p+2}. A l'état initial, on lance $\mathcal{M} : (B_1, \ldots, B_{p+1}) \to B_{p+2}$. Si la bande B_{p+2} représente 0, on passe à l'état final. Sinon, on incrémente le contenu de la bande B_{p+1} et on retourne à l'état initial.
On notera en particulier dans toutes ces constructions que les cas de fonctionnement infini de la machine sont exactement les cas où les fonctions à calculer ne sont pas définies. $\square$

Codages

Pour l'autre direction, on devra être capable de coder la situation d'une machine de Turing à un instant donné.

Codage des couples : pour coder le couple $(x, y) \in \mathbb{N}^2$, on utilise la bijection bien connue $\alpha_2 : \mathbb{N}^2 \to \mathbb{N}$,

$$\alpha_2(x, y) = (\sum_{i=0}^{x+y} i) + y = \frac{(x + y)(x + y + 1)}{2} + y.$$

C'est une fonction primitive récursive (la division par 2 des entiers pairs s'obtient par minimisation bornée). Notons que $x \leq \alpha_2(x, y)$ et $y \leq \alpha_2(x, y)$, ce qui nous permet de constater que ls fonctions de décodages sont elles aussi primitives récursives :

$\beta_1^2(z) = \mu x \leq z, (\exists y \leq z, \alpha_2(x, y) = z)$ et $\beta_1^2(z) = \mu y \leq z, (\exists x \leq z, \alpha_2(x, y) = z)$.

On peut ensuite définir par réccurence sur p les fonctions de codage et de décodage des p-uplets, qui sont encore primitives récursives :

$$\begin{cases} \alpha_{p+1}(x_1, \ldots, x_{p+1}) = \alpha_2(x_1, \alpha_p(x_2, \ldots, x_{p+1})) \\ \beta_1^{p+1} = \beta_1^2, \beta_2^{p+1} = \beta_1^p \circ \beta_2^2, \ldots, \beta_{p+1}^{p+1} = \beta_p^p \circ \beta_2^2. \end{cases}$$

On pose enfin $\alpha_1 = \beta_1^1 = \mathrm{id} : \mathbb{N} \to \mathbb{N}$.

Codage des suites nulles à partir d'un certain rang :
pour une telle suite $u = (u_0, u_1, \ldots)$, on pose

$$\Omega(u) = \Pi_{i=0}^{\infty} \pi(i)^{u_i}.$$

On rappelle que π est la fonction primitive récursive qui à i associe le $(i+1)$-ème nombre premier. La fonction Ω est bien définie car les suites sont nulles à partir d'un certain rang, elle est injective, d'image $\mathbb{N} \setminus \{0\}$. Notons que cela n'a pas de sens de se demander si Ω est récursive, puisque son domaine de définition n'est pas un $\mathbb{N}^p$. Remarquons que pour tout élément u_i de la suite u, $u_i \leq \Omega(u)$.
On définit une fonction de décodage $\delta : \mathbb{N}^2 \to \mathbb{N}$, qui est primitive récursive :

$$\delta(i, c) = \mu v \leq c, (\pi(i)^{v+1} \not| c).$$

On a bien la relation $\delta(i, \Omega(u)) = u_i$.

Considérons maintenant une machine de Turing $\mathcal{M}$, avec m états nommés $0, 1, \ldots, m - 1$. On supposera toujours que 0 est l'état initial et 1 l'état final. Chacune des bandes $B_1, \ldots, B_k$ de $\mathcal{M}$ est une suite de symboles d (codé par 2), | (codé par 1), ou $\square$ (codé par 0). En particulier, à tout instant, chaque bande B_i peut être vue comme une suite d'entiers nulle à partir d'un certain rang, on peut donc la coder par l'entier $\Omega(B_i)$. A tout instant t, la situation de la machine $\mathcal{M}$ est déterminé par son état e, le contenu de ses bandes $B_1, \ldots, B_k$, et la position x de la tête de lecture, on code cette situation par l'entier

$$\alpha_3(e, \alpha_k(\Omega(B_1), \ldots, \Omega(B_k)), x).$$

Lemme 4.14 *Il existe une fonction primitive récursive g telle que, si c est le code de la situation de $\mathcal{M}$ à l'instant t, alors $g(c)$ est le code de la situation de $\mathcal{M}$ à l'instant $t+1$.*

Preuve On décode c par des fonctions primitives récursives :
$e(c) = \beta_1^3(c)$ l'état de $\mathcal{M}$
$x(c) = \beta_3^3(c)$ la position de la tête de lecture
$s_i(c) = \delta(x(c), \beta_i^k(\beta_2^3(c)))$ le symbole de la bande B_i sous la tête de lecture.
En utilsant la table de transition M, et une définition de fonctions primitives récursives parmi les cas dans $\{0, \ldots, m-1\} \times \{0,1,2\}^k$, on a des fonctions primitives récursives $e'(c), t_1(c), \ldots, t_k(c), \epsilon(c)$ donnant respectivement le nouvel état de $\mathcal{M}$, les nouveaux symboles à écrire sur chaque bande, et le déplacement de la tête à faire (formellement, il faudrait coder $\epsilon(c)$ par un entier naturel).
On obtient alors g comme fonction primitive récursive :

$$g(c) = \alpha_3(e'(c), \alpha_k(\ldots, \beta_i^k(\beta_2^3(c))\pi(x(c))^{t_i(c)-s_i(c)}, \ldots), x(c) + \epsilon(c)).$$

□

Lemme 4.15 *Il existe une fonction primitive récursive $h : \mathbb{N}^{p+1} \to \mathbb{N}$ telle que, si le fonctionnement de la machine $\mathcal{M}$ est lancé avec les entiers $x_1, \ldots, x_p$ représentés sur les bandes $B_1, \ldots, B_p$, et 0 sur les autres bandes $B_{p+1}, \ldots, B_k$, alors $h(x_1, \ldots, x_p, t)$ est le code de la situation de $\mathcal{M}$ à l'instant t.*

Preuve On va définir la fonciton h par le schéma de récurrence primitive, en utilisant la fonction g du lemme précédent. Pour l'instant initial, notons que si l'entier x_i est représenté sur la bande B_i, alors $\Omega(B_i) = 4\Pi_{i=1}^{x_i}\pi(i)$; en particulier c'est une fonction primitive récursive par rapoort à x_i. On a donc la définition de h :

$$\begin{cases} h(x_1, \ldots, x_p, t) = \alpha_3(0, \alpha_k(4\Pi_{i=1}^{x_1}\pi(i), \ldots, \Pi_{i=1}^{x_i}\pi(i), 4, \ldots, 4), 0) \\ h(x_1, \ldots, x_p, t+1) = g(h(x_1, \ldots, x_p, t)). \end{cases}$$

□

Lemme 4.16 *Il existe une fonction récursive partielle $T : \mathbb{N}^p \to \mathbb{N}$ telle que, si le fonctionnement de $\mathcal{M}$ est lancée comme dans le lemme précédent, $T(x_1, \ldots, x_p)$ est le temps d'arrêt de la machine s'il existe, et $T(x_1, \ldots, x_p) \uparrow$ sinon.*

Preuve La fonction T est obtenue en utilisant la fonction primitive récursive h du lemme précédent, et le schéma de minimisation :

$$T(x_1, \ldots, x_p) = \mu t, \beta_1^3(h(x_1, \ldots, x_p, t)) = 1.$$

□

Corollaire 4.17 *La fonction f calculée par $\mathcal{M}$ est une fonction récursive partielle.*

Preuve En utilisant les fonctions h et T définies dans les lemmes précédents, on trouve le code de la bande de sortie B_{p+1} à l'instant d'arrêt s'il existe, et non défini sinon :

$$\tilde{f}(x_1, \ldots, x_p) = \beta^k_{p+1}(\beta^3_2(h(x_1, \ldots, x_p, T(x_1, \ldots, x_p)))).$$

Puis on décode l'entier représenté sur cette bande, en notant au passge que cet entier est nécessairement inférieur au code de la bande :

$$f(x_1, \ldots, x_p) = \mu i \leq \tilde{f}(x_1, \ldots, x_p), (\pi(i+1) \not| \tilde{f}(x_1, \ldots, x_p)).$$

La fonction f est donc bien récursive partielle. □

Remarque 4.18
- *L'introduction de la fonction T est le seul moment où on est sorti du schéma de construction des fonctions primitives récursives.*
- *Outre le fait que ce soit une instance particulière de la thèse de Church pour les fonctions calculables par machine de Turing, le théorème que l'on vient de démontrer est utile théoriquement pour obtenir certaines propriétés des ensembles récursivementénumérables introduits ci-dessous.*

4.4 Ensembles récursivement énumérables

Définition 4.19 *Un ensemble $X \subseteq \mathbb{N}^p$ est récursivement énumérable si c'est le domaine de définition d'une fonction récursive partielle.*

Proposition 4.20 *Les ensembles récursifs sont récursivement énumérables.*

Preuve On introduit la fonction $f : x \mapsto \mu y, y + 1 = x$. C'est une fonction récursive partielle, de domaine $\mathrm{Dom}(f) = \mathbb{N} \setminus \{0\}$. Alors, pour un ensemble récursif X, sa fonction caractéristique $\mathbf{1}_X$ est récursive et $X = \mathrm{Dom}(f \circ \mathbf{1}_X)$ est récursivement énumérable. □

Remarque 4.21 *La classe des ensembles récursivement énumérables est strictement plus large que celle des ensembles récursifs, on verra des exemples en exercice. Intuitivement, pour un ensemble récursivement énumérable X, on a un algorithme pour s'assurer que $(x_1, \ldots, x_p) \in X$, mais il ne termine pas pour les p-uplets qui ne sont pas dans X.*

Proposition 4.22 *Soient X_1 et X_2 deux sous-ensembles récursivement énumérables d'un même $\mathbb{N}^p$. Alors $X_1 \cap X_2$ et $X_1 \cup X_2$ sont récursivement énumérables.*

Preuve Soient f_1 et f_2 des fonctions récursives partielles telles que $X_i = \mathrm{Dom}(f_i)$. On a clairement $X_1 \cap X_2 = \mathrm{Dom}(f_1 + f_2)$, donc $X_1 \cap X_2$ est récursivement énumérable.
Pour l'union, on considère pour $i = 1, 2$ la fonction $h_i : \mathbb{N}^{p+1} \to \mathbb{N}$ comme donnée dans la section précédente, donnant la situation d'une machine de Turing $\mathcal{M}_i$ calculant f_i à tout instant. Alors $X_1 \cup X_2 = \mathrm{Dom}((x_1, \ldots, x_p) \mapsto \mu t, (\beta^3_1(h_1(x_1, \ldots, x_p, t)) = 1 \vee \beta^3_1(h_2(x_1, \ldots, x_p, t)) = 1))$ est récursivement énumérable. □

Remarque 4.23 – *La technique précédente permet de simuler l'exécution simultanée des machines $\mathcal{M}_1$ et $\mathcal{M}_2$, avec arrêt dès que l'une des deux machines s'arrête. On utilisera encore cette idée dans la suite.*
– *Attention, la classe des des ensembles récursivement énumérables n'est pas close par passage au complémentaire. La proposition suivante permet de clarifier la situation.*

Proposition 4.24 *Soit $X \subseteq \mathbb{N}^p$. Alors X est récursif si et seulement si X et X^c sont récursivement énumérables.*

Preuve $\Rightarrow$ On sait que si X est récursif, X^c l'est aussi ; or les ensembles récursifs sont récursivement énumérables.
$\Leftarrow$ On suppose que $X = \mathrm{Dom}(f_1)$ et $X^c = \mathrm{Dom}(f_2)$, avec pour $i = 1, 2$, $\mathcal{M}_i$ une machine de Turing calculant la fonction récursive partielle f_i et h_i la fonction primitive récurisve donnant la situation de la machine $\mathcal{M}_i$ à tout instant.
On a alors une fonction récursive (totale car $X \cup X^c = \mathbb{N}^p$)

$$T(x_1, \ldots, x_p) = \mu t, (\beta_1^3(h_1(x_1, \ldots, x_p, t)) = 1 \vee \beta_1^3(h_2(x_1, \ldots, x_p, t)) = 1)).$$

Et la fonction $\mathbf{1}_X$ est alors la fonction récursive

$$\mathbf{1}_X(x_1, \ldots, x_p) = \begin{cases} 1 \text{ si } \beta_1^3(h_1(x_1, \ldots, x_p, T(x_1, \ldots, x_p))) = 1 \\ 0 \text{ sinon} \end{cases} .$$

□

En particulier, pour un exemple d'ensemble X qui est récursivement énumérable mais pas récursif, on a nécessairement que X^c n'est pas récursivement énumérable.

Pour finir, on justifie le vocabulaire « récursivement énumérable » par la caractérisation suivante pour les sous-ensembles de $\mathbb{N}$.

Proposition 4.25 *Soit $X \subseteq \mathbb{N}$. Alors X est récursivement énumérable si et seulement si $X = \emptyset$ ou X est l'image d'une fonction récursive $f : \mathbb{N} \to \mathbb{N}$.*

Preuve $\Leftarrow$ $\emptyset$ est récursivement énumérable puisque la fonction nulle part définie est récursive partielle. Et si $X = \mathrm{Im}(f)$ pour une fonction récursive $f : \mathbb{N} \to \mathbb{N}$, alors $\mathbf{1}_X = \mathrm{Dom}(y \mapsto \mu x, f(x) = y)$ (il est nécessaire que f soit totale pour avoir cette égalité).
$\Rightarrow$ Soit X récursivement énumérable. Si X est vide il n'y a rien à montrer. Sinon, on choisit $a \in X$, et on considère une machine de Turing $\mathcal{M}$ calculant une fonction récursive partielle $f : \mathbb{N} \to \mathbb{N}$ de domaine X, et h une fonction primitive récursive donnant la situation de $\mathcal{M}$ à tout instant.
On pose la fonction primitive récursive $\tilde{g} : \mathbb{N}^2 \to \mathbb{N}$

$$\tilde{g}(x, t) = \begin{cases} a \text{ si } \beta_1^3(h(x, t)) \neq 1 \\ x \text{ sinon} \end{cases}$$

Autrement dit, pour $x \in \mathbb{N}$ fixé, $\tilde{g}(x, t) = a$ tant que t est strictement inférieur au temps de calcul de $f(x)$ par la machine $\mathcal{M}$, puis il prend la valeur x une fois ce temps de calcul atteint. On a donc bien $\mathrm{Im}(\tilde{g}) = \mathrm{Dom}(f) = X$. On se

ramène à une fonction $\mathbb{N} \to \mathbb{N}$ en codant le couple (x, t), c'est-à-dire en posant $g : z \mapsto \tilde{g}(\beta_1^2(z), \beta_2^2(z))$ (on peut même noter que cette fonction g telle que $X = \mathrm{Im}(g)$ est en fait primitive récursive). □

Chapitre 5

Arithmétique

Objectif : axiomatisation de l'arithmétique. Peut-on donner une liste d'axiomes "raisonnable" pour $\mathrm{Th}(\mathbb{N}, 0, +, \cdot, s)$?

5.1 Arithmétique de Péano

Langage $\mathcal{L} = (\emptyset, \{s, +, \cdot\}, \{0\})$ (s fonction unaire pour successeur).
Axiomes AP :

1. $\forall x\ \neg 0 \equiv s(x)$
2. $\forall x \forall y (s(x) \equiv s(y) \to x \equiv y)$
3. $\forall x \exists y (\neg x \equiv 0 \to x \equiv s(y))$
4. $\forall x\ x + 0 \equiv x$
5. $\forall x \forall y\ x + s(y) = s(x + y)$
6. $\forall x\ x \cdot 0 \equiv 0$
7. $\forall x \forall y\ x \cdot s(y) \equiv x \cdot y + x$
8. Schéma d'axiomes (prinicipe d'induction) : pour toute formule $F[x, x_1, \ldots, x_n]$, axiome $\forall x_1 \ldots \forall x_n ((F[0, x_1, \ldots, x_n] \wedge \forall y (F[y, x_1, \ldots, x_n] \to F[s(y), x_1, \ldots, x_n])) \to \forall x\ F[x, x_1, \ldots, x_n])$

Exemple 5.1 *$\mathbb{N}$, avec les interprétations naturelles du langage $\mathcal{L}$, est un modèle de AP, appelé modèle standard. Il existe aussi des modèles non standards de AP : considérer par exemple un modèle de cardinal 2^ω, qui existe d'après le théorème de Löwenheim-Skolem.*

Proposition 5.2 *Dans un modèle $\mathcal{M}$ de AP, $+$ et $\cdot$ sont associatives, commutatives, $\cdot$ est distributif sur $+$, $+$ est régulière, les éléments non nuls sont réguliers pour $\cdot$*

Preuve Montrons par exemple que $+$ est commutatif : on montre d'abord $\forall x\ 0 + x \equiv x$ (par induction, en utilisant 4 et 5), puis $\forall y \forall x\ s(x + y) \equiv s(x) + y$ (par induction sur y) ; cela nous donne l'étape initiale et le pas d'induction pour montrer $\forall x \forall y\ x + y \equiv y + x$ par induction sur x. □

Définition 5.3 *On écrit $x \leq y$ pour $\exists z\ x + z \equiv y$*

Proposition 5.4 *Dans tout modèle $\mathcal{M}$ de AP, $\leq$ est une relation d'ordre total, compatible avec $+$ et $\cdot$.*

Preuve Montrons par exemple que l'ordre est total : on montre successivement que $x \leq s(x)$ (car $s(x) = x + s(0)$), que $(x \leq y) \to (x \leq s(y))$ (par transitivité, qui découle immédiatement de la définition), que $(x < y) \to \exists z(x + s(z) = y)$ (en utilisant 3), que $(x < y) \to (s(x) \leq y)$ (car $x + s(y) = s(x) + y$), et enfin que $(x \leq y \vee y \leq x) \to (x \leq s(y) \vee s(y) \leq x)$ en combinant ce qui précède, ce qui nous permet de montrer que $\forall y \forall x(x \leq y \vee y \leq x)$ par induction sur y. □

Fait 5.5 *Les axiomes de AP permettent aussi de montrer l'existence d'une division euclidienne, et à partir de là, de nombreux résultats arithmétiques de base. On peut même énoncer et démontrer par AP une forme adaptée du théorème de répartition des nombres premiers (c'est un résultat beaucoup plus difficile).*

Proposition 5.6 *Dans tout modèle $\mathcal{M}$ de AP, il existe une unique sous-structure $\mathcal{N}$ de $\mathcal{M}$ isomorphe à $\mathbb{N}$. C'est un segment initial de $\mathcal{M}$.*

Preuve Pour tout $n \in \mathbb{N}$, on note $\underline{n}$ le terme $s(s \dots (0))$ (n fois). Alors $\overline{\underline{n}}^{\mathbb{N}} = n$, et un monomorphisme ϕ de $\mathbb{N}$ vers $\mathcal{M}$ doit nécessairement envoyer $\overline{\underline{n}}^{\mathbb{N}}$ vers $\overline{\underline{n}}^{\mathcal{M}}$. Notons $N = \{\overline{\underline{n}}^{\mathcal{M}}; n \in \mathbb{N}\}$; ce sous-ensemble de M contient 0, est stable par s, $+$ et $\cdot$, c'est donc l'ensemble de base d'une sous-structure $\mathcal{N}$ de $\mathcal{M}$, qui est isomorphe à $\mathbb{N}$.
On montre par récurrence sur les entiers (intuitifs) $n \in \mathbb{N}$ que $\forall x(x \leq \underline{n} \to (x \equiv 0 \vee \dots \vee x \equiv \underline{n}))$: si $n = 0$, $x \leq 0$ permet de trouver y tel que $x + y = 0$, et si $y \neq 0$, $x + y = x + s(z) = s(x + z) = 0$, ce qui est impossible, donc $y = 0$ et $x = x + 0 = 0$; et si c'est vrai à l'étape n, $x \leq \underline{n+1}$ permet de trouver y tel que $x + y = \underline{n+1}$, le résultat est vérifié si $y = 0$, et sinon, $y = s(z)$ et $x + y = s(x + z) = \underline{n+1}$, ce qui donne $x + z = \underline{n}$ par injectivité de s, et donc $x = 0 \vee \dots \vee x = \underline{n}$ par hypothèse de récurrence. On a donc bien montré que $\mathcal{N}$ est un segment initial de $\mathcal{M}$. □

5.2 Fonctions représentables

Définition 5.7 *On dit qu'une fonction $f : \mathbb{N}^p \to \mathbb{N}$ est représentée par une formule $F[y, x_1, \dots, x_p]$ si pour tout $(n_1, \dots, n_p) \in \mathbb{N}^p$, $AP \vdash \forall y(F[y, \underline{n_1}, \dots, \underline{n_p}] \leftrightarrow y \equiv \underline{f(n_1, \dots, n_p)})$.*
On dit qu'un ensemble $A \subseteq \mathbb{N}^p$ est représentable si sa fonction caractéristique $\mathbf{1}_A$ l'est.

Remarque 5.8 *L'ensemble A est représenté par F si $(n_1, \dots, n_p) \in A$ implique $AP \vdash F[\underline{n_1}, \dots, \underline{n_p}]$ et $(n_1, \dots, n_p) \notin A$ implique $AP \vdash \neg F[\underline{n_1}, \dots, \underline{n_p}]$.*

Exemple 5.9 *Les fonctions $s, +, \cdot$ sont représentables.*
L'ensemble $\{(x, y) \in \mathbb{N}^2; x \leq y\}$ est représentable.
La fonction puissance, définie par récurrence à partir de la fonction $\cdot$, n'est

pas dans le langage mais est représentable : c'est une conséquence du résultat suivant.

Théorème 5.10 *Les fonctions récursives sont représentables.*

Preuve On montre sans peine que la classe des fonctions représentables contient la fonction nulle, la fonction successeur et les projections.
Cette classe est close par composition : si $h : \mathbb{N}^k \to \mathbb{N}$ est représentée par $H[y, x_1, \ldots, x_k]$ et si $g_1, \ldots, g_k : \mathbb{N}^p \to \mathbb{N}$ sont représentée par $G_1[y, \overline{x}], \ldots, G_k[y, \overline{x}]$ (on écrit $\overline{x}$ pour $x_1, \ldots, x_p$), la fonction $h \circ (g_1, \ldots, g_k)$ est représentée par :

$$\exists y_1 \ldots \exists y_k (\bigwedge_{i=1}^{k} G_i[y_i, \overline{x}] \wedge H[y, y_1, \ldots, y_k]).$$

La classe est close par minimisation (totale) : si $f : \mathbb{N}^{p+1} \to \mathbb{N}$ est représentée par $F[z, x_1, \ldots, x_p, y]$, la fonction $\mu y(f(x_1, \ldots, x_p, y) = 0)$ est représentée par $G[y, x_1, \ldots, x_p] = (F[0, x_1, \ldots, x_p, y] \wedge \forall z < y \neg F[0, x_1, \ldots, x_p, z])$.
Il reste donc à voir que la classe des fonctions représentables est close par le schéma de récurrence primitive. Soient $g : \mathbb{N}^p \to \mathbb{N}$ représentée par une formule $G[y, x_1, \ldots, x_p]$ et $h : \mathbb{N}^{p+2} \to \mathbb{N}$ représentée par une formule $H[y, x_1, \ldots, x_{p+2}]$. Alors on veut écrire que la fonction $f = R(g, h)$ définie par récurrence primitive sera représentée par la formule

$$F[y, x_1, \ldots, x_p, n] = \text{il existe une suite finie } (y_0, \ldots, y_n) \text{ telle que}$$

$$(G[y_0, x_1, \ldots, x_p] \wedge (\forall i < n) H[y_{i+1}, x_1, \ldots, x_p, i, y_i] \wedge y_n \equiv y).$$

Reste à savoir comment exprimer "il existe une suite finie" par une formule : cela découle du résultat suivant.

Proposition et définition 5.11 *Il existe une fonction représentable $\beta : \mathbb{N}^3 \to \mathbb{N}$, dite fonction β de Gödel, telle que pour tout $n \in \mathbb{N}$, pour toute suite finie $(m_0, \ldots, m_n) \in \mathbb{N}^n$, il existe a et b dans $\mathbb{N}$, tels que pour tout i entre 0 et n, $\beta(i, a, b) = m_i$.*

Preuve On pose $\beta(i, a, b) =$ le reste de la division euclidienne de b par $a(i+1)+1$, ce qui est représentée par la formule $B[y, i, a, b] = (y < a \cdot (i+1) + 1 \wedge \exists z\ b = z \cdot (a \cdot (i+1) + 1) + y)$. Alors pour une suite $(m_0, \ldots, m_n)$, on choisit $m \geq n+1$ tel que $a := m! \geq m_i$ pour tout i. On vérifie que les $a(i+1)+1$ sont premiers entre eux deux à deux : si q premier divise à la fois $a(i+1)+1$ et $a(j+1)+1$ pour $j > i$, il divise aussi $a(j-i) = m!(j-i)$, avec $j - i \leq m$, ce qui implique que $q \leq m$, donc q divise a et ne peut pas diviser $a(i+1)+1$. On obtient donc b par le théorème des restes chinois. □

En particulier, pour finir la démonstration du théorème, la formule $F[y, x_1, \ldots, x_p, n]$ s'écrit :

$$\exists a \exists b (\exists z (G[z, \overline{x}] \wedge B[z, 0, a, b]) \wedge (\forall i < n\ \exists z \exists w (H[z, \overline{x}, i, w] \wedge B[z, i+1, a, b] \wedge B[w, i, a, b])) \wedge B[y, n, a, b]).$$

□

Remarque 5.12 *Dans la formule précédente, on a utilisé le couple (a, b) comme un code pour la suite finie des valeurs $f(\overline{x}, 0), \ldots, f(\overline{x}, n)$, et la fonction de Gödel β sert de fonction de décodage. Au chapitre précédent, on avait aussi introduit une fonction Ω de codage des suites finies, avec une fonction de décodage δ primitive récursive, donc représentable a posteriori d'après le théorème 5.10. L'intérêt de la fonction β est qu'on a été capable de montrer "à la main" qu'il s'agit d'une fonction représentable.*

5.3 Codage du calcul des prédicats

On associe à chaque symbole s un code $\sharp s \in \mathbb{N}$, en utilisant par exemple l'énumération : $\wedge \neg \forall \equiv s + \cdot 0\ x_0\ x_1\ x_2 \ldots$ (on a ainsi $\sharp x_i = i + 8$).
On utilisera les fonctions primitives récursives α_n et β_i^n vues au chapitre précédent.

Codage des termes
Si $t = 0$ ou $t = x_n$, $\sharp t = \alpha_3(\sharp 0, 0, 0)$ ou $\sharp t = \alpha_3(\sharp x_n, 0, 0)$
Si $t = s(t')$, $\sharp t = \alpha_3(\sharp s, \sharp t', 0)$
Si $t = t_1 + t_2$, $\sharp t = \alpha_3(\sharp +, \sharp t_1, \sharp t_2)$, et idem pour $\cdot$

Proposition 5.13 *Ce codage est injectif.*
L'ensemble $Term = \{\sharp t; t$ est un terme$\}$ est primitif récursif.

Preuve L'injectivité ne pose pas de problème (la fonction α_3 est bijective).
Pour montrer que $f : i \mapsto \mathbf{1}_{Term}(i)$ est primitive récursive, on utilise le fait que $f(i)$ est déterminé par les valeurs de $f(j)$ pour $j < i$. Pour montrer que cette construction entre dans le cadre d'une récurrence primitive, on introduit la fonction auxiliaire $g(n) := \pi(0)^{f(0)} \ldots \pi(n)^{f(n)}$, qui est définie par une récurrence primitive :
$g(0) = \pi(0)^{f(0)}$, une constante
$g(n+1) = g(n)\pi(n+1)^u$, avec $u = 1$ si et seulement si $n+1 = \alpha_3(\sharp 0, 0, 0)$ OU ...
OU $\exists n_1 \leq n\, \exists n_2 \leq n\, (n + 1 = \alpha_3(\sharp \cdot, n_1, n_2) \wedge \delta(n_1, g(n)) = 1 \wedge \delta(n_2, g(n)) = 1)$
(c'est bien une condition primitive récursive portant sur n et $g(n)$).
La fonction f voulue s'obtient alors par décodage : $f(n) = \delta(n, g(n))$, c'est une fonction primitive récursive. □

On ne fera pas les démonstrations suivantes, qui reprennent les mêmes arguments.

Codage des formules
Si $F = t_1 \equiv t_2$, $\sharp F = \alpha_3(\sharp \equiv, \sharp t_1, \sharp t_2)$
Si $F = \neg G$, $\sharp F = \alpha_3(\sharp \neg, \sharp G, 0)$
Si $F = (G \wedge H)$, $\sharp F = \alpha_3(\sharp \wedge, \sharp G, \sharp H)$
Si $F = \forall x_n G$, $\sharp F = \alpha_3(\sharp \forall, \sharp x_n, \sharp G)$

Proposition 5.14 *Ce codage est injectif.*
L'ensemble $Form = \{\sharp F; F$ est une formule$\}$ est primitif récursif.

Proposition 5.15 *Il existe des fonctions primitives récursives $Subst_f$ et $Subst_t$ telles que, pour tous termes t, u, toute formule F et toute variable x_n,*

$$Subst_t(\sharp t, \sharp u, \sharp x_n) = \sharp(t[u/x_n]) \quad Subst_f(\sharp F, \sharp u, \sharp x_n) = \sharp(F[u/x_n]).$$

L'ensemble $Enon = \{\sharp F; F$ est un énoncé$\}$ est primitif récursif.

Codage des démonstrations

Pour coder les suites de formules, on utilise une version modifiée de la fonction Ω définie au chapitre précédent :
$\Omega(\emptyset) = 1$
$\Omega((x_1, \ldots, x_n)) = \pi(0)^{x_1+1} \ldots \pi(n-1)^{x_n+1}$ (où $\pi(n)$ est le $(n+1)$-ième nombre premier).
Alors

- la fonction Ω est injective, de l'ensemble des suites finies d'entiers dans $\mathbb{N}$
- L'image de Ω est un ensemble primitif récursif
- La fonction longueur de $Im(\Omega)$ dans $\mathbb{N}$ est primitive récursive
- La fonction $\delta : (i, c) \mapsto \begin{cases} 0 \text{ si } c \notin Im(\Omega) \\ x_i \text{ si } c = \Omega((x_j)_{j \leq n}) \text{ et } i \leq n \\ 0 \text{ si } c = \Omega((x_j)_{j \leq n}) \text{ et } i > n \end{cases}$ est primitive récursive.

Preuve L'injectivité de Ω vient de l'unicité de la décomposition en facteurs premiers. On a la condition $c \in Im(\Omega)$ si et seulement si $\forall p \leq c((p \text{ premier} \wedge p \text{ divise } c) \rightarrow \forall q \leq p(q \text{ premier} \rightarrow q \text{ divise } c))$. La longueur de c est donnée par $l(c) = \mu i \leq c(\pi(i) \text{ ne divise pas } c)$, et la fonction de décodage par

$$\delta(i, c) = \begin{cases} \mu j \leq c(\pi(i)^{j+2} \text{ ne divise pas } c) \text{ si } c \in Im(\Omega) \wedge i < l(c) \\ 0 \text{ sinon} \end{cases}$$

□

On codera une suite finie de formules $D = (F_1, \ldots, F_n)$ par $\sharp D = \Omega((\sharp F_1, \ldots, \sharp F_n))$.

Définition 5.16 *Une théorie T est récursive si $\{\sharp F; F \in T\}$ est un ensemble récursif.*
Une théorie T est décidable si $Csq_T = \{\sharp F; F$ énoncé et $T \vdash F\}$ est un ensemble récursif.

Exemple 5.17 *AP est une théorie récursive : on montre que la fonction qui au code d'une formule $F[x, x_1, \ldots, x_n]$ associe le code du principe d'induction pour cette formule est primitive récursive, et alors le prédicat $x = \sharp A1 \vee \ldots \vee x = \sharp A7 \vee \exists y \leq x(y = \sharp F \wedge$ "x est le code du prinicpe d'induction pour F") est primitif récursif.*

Remarque 5.18 *Plus généralement, quand on recherche une axiomatisation "raisonnable" de $Th(\mathbb{N})$, on recherche une axiomatisation récursive : la moindre des choses qu'on demande à une liste d'axiomes est de pouvoir repérer automatiquement si un énoncé est un axiome.*
Remarquons aussi que si une théorie T est décidable, on peut l'axiomatiser par une théorie récursive, à savoir l'ensemble des conséquences de T. Les résultats suivants portent sur la réciproque de cette propriété.

Proposition 5.19 *Soit T une théorie récursive. Alors l'ensemble*

$Dem_T = \{(x,y) \in \mathbb{N}^2; x = \sharp F$ *pour un énoncé* $F, y = \sharp D$ *pour une démonstration* D *de* F *dans* $T\}$

est récursif.

Preuve On écrit les conditions (primitives récursives) $y \in Im(\Omega)$, $x \in$ Enon et $x = \delta(l(y)-1, y)$, puis, pour tout $i \leq l(y)-1$: $\delta(i,y)$ est le code d'un axiome logique (cette condition est primitive récursive : c'est une disjonction de prédicats du type

$$\exists a \leq \delta(i,y)\, \exists b \leq \delta(i,y)(a \in \text{Form} \wedge b \in \text{Form} \wedge \delta(i,y) = \alpha_3(\sharp \rightarrow, a, \alpha_3(\sharp \rightarrow, b, a))),$$

en écrivant la dernière expression de manière à ne faire intervenir que les connecteurs $\wedge$ et $\neg$), OU $\delta(i,y)$ est le code d'un axiome de T (récursif par hypothèse), OU il existe $j,k < i$ tels que $\delta(k,y) = \sharp(F_j \rightarrow F_i)$, où $\delta(i,y) = \sharp F_i$ et $\delta(j,y) = \sharp F_j$, OU il existe $j < i$ et $m \leq \delta(i,y)$ tels que $\delta(i,y) = \sharp(\forall x_m F_j)$, où $\delta(j,y) = \sharp F_j$. □

Corollaire 5.20 *Soit T une théorie récursive. Alors l'ensemble des conséquences de T Csq_T est récursivement énumérable.*

Preuve C'est le domaine de définition de la fonction récursive $x \mapsto \mu y((x,y) \in \text{Dem}_T)$, qui permet bien de décrire toutes les conséquences de T d'après le théorème de complétude. □

Corollaire 5.21 *Soit T une théorie complète et récursive. Alors T est décidable.*

Preuve On vient de voir que Csq_T est récursivement énumérable, et son complémentaire $\text{Csq}_T^c = \text{Enon}^c \cup \{x \in \text{Enon}; \alpha_3(\sharp\neg, x, 0) \in \text{Csq}_T\}$ est aussi récursivement énumérable (on utilise la complétude de T pour dire que si $T \nvdash F$, alors $T \vdash \neg F$). □

Exemple 5.22 *La théorie des corps algébriquement clos de caractéristique* 0 *est récursive et complète, donc décidable.*

5.4 Théorèmes d'incomplétude de Gödel

Lemme 5.23 *Soit T une théorie cohérente contenant AP, alors T est indécidable.*

Théorème 5.24 (Premier théorème d'incomplétude) *Soit T une théorie récursive cohérente contenant AP. Alors T n'est pas complète.*

Théorème 5.25 (Second théorème d'incomplétude) *Soit T une théorie récursive cohérente contenant AP. Alors $T \nvdash Coh_T$, où Coh_T est l'énoncé $\neg\exists x(C_T(x) \wedge C_T(\alpha_3(\sharp\neg, x, 0))$ et C_T une "bonne" formule représentant l'ensemble Csq_T.*

Le premier théorème d'incomplétude découle directement du lemme et du corollaire 5.21. Montrons le lemme.

Preuve On se donne une théorie $T \supseteq AP$ décidable, et on pose

$$\Theta = \{(x, y) \in \mathbb{N}^2; x = \sharp(F[x_0]) \wedge \sharp(F[\underline{y}/x_0]) \in \mathrm{Csq}_T\}.$$

Comme T est décidable, Θ est un ensemble récursif.
Soit l'ensemble récursif $B = \{n \in \mathbb{N}; (n, n) \notin \Theta\}$ et $G[x_0]$ une formule qui représente B.
Alors $n \in B$ implique $AP \vdash G[\underline{n}/x_0]$ et donc $T \vdash G[\underline{n}/x_0]$ et de même $n \notin B$ implique $T \vdash \neg G[\underline{n}/x_0]$.
Soit $a = \sharp G[x_0]$. Si $a \in B$, alors $(a, a) \notin \Theta$, c'est-à-dire $T \nvdash G[\underline{a}/x_0]$, ce qui donne $a \notin B$ par la remarque précédente. Ainsi $a \notin B$, ou encore $(a, a) \in \Theta$, donc $T \vdash G[\underline{a}/x_0]$, et $T \vdash \neg G[\underline{a}/x_0]$ puisque $a \notin B$: la théorie T est donc incohérente. □

Remarque 5.26
1. *La formule $G[\underline{a}]$ de la démonstration signifie : "je ne suis pas démontrable".*
2. *Il est essentiel de supposer que T est récursive dans le premier théorème d'incomplétude : en effet $T = Th(\mathbb{N})$ est complète (cohérente) et contient AP.*

Montrons maintenant le second théorème d'incomplétude.

Preuve On pose la fonction $g : \mathbb{N} \to \mathbb{N}$

$$g(n) = \begin{cases} \sharp(F[\underline{n}/x_0]) \text{ si } n = \sharp(F[x_0]) \\ 0 \text{ sinon} \end{cases}.$$

En utilisant la fonction $Subst_f$, on voit que g est une fonction primitive récursive, donc représentable par une fonction $G[y, x]$.
On a ainsi, pour tout $n \in \mathbb{N}$,

$$AP \vdash \forall y(G[y, \underline{n}] \leftrightarrow y \equiv \underline{g(n)}). \quad (1)$$

Soit $D_T[x, y]$ une formule représentant le prédicat Dem_T, on considère la formule

$$\epsilon[x_0] = \exists x_1 \exists x_2 (D_T[x_2, x_1] \wedge G[x_2, x_0]).$$

En particulier, si $n = \sharp(F[x_0])$,

$$\mathbb{N} \models \epsilon[\underline{n}] \text{ si et seulement si } T \vdash F[\underline{n}/x_0].$$

Soit $a = \sharp(\neg\epsilon[x_0])$, et $b = g(a) = \sharp(\neg\epsilon[\underline{a}/x_0])$. Par (1), on a

$$AP \vdash (\epsilon[\underline{a}/x_0] \leftrightarrow \exists x_1 D_T[\underline{b}, x_1]). \quad (2)$$

Si $T \vdash \neg\epsilon[\underline{a}/x_0]$, il existe $c \in \mathbb{N}$ le code d'une démonstration de $\neg\epsilon[\underline{a}/x_0]$, et donc, par définition de la notion de représentation, $AP \vdash D_T[\underline{b}, \underline{c}]$. Alors par (2), $AP \vdash \epsilon[\underline{a}/x_0]$, et a fortiori $T \vdash \epsilon[\underline{a}/x_0]$. Ainsi T est incohérente, ce qui est

faux par hypotèse.
On a donc montré $T \nvdash \neg\epsilon[\underline{a}/x_0]$. Montrons maintenant que $T \vdash (\mathrm{Coh}_T \to \neg\epsilon[\underline{a}/x_0])$ (où Coh_T est l'énoncé donné dans l'énoncé du théorème) :
par (2), $AP \cup \{\epsilon[\underline{a}/x_0]\} \vdash \exists x_1 D_T[\underline{b}, x_1]$. Soit $d = \sharp(\epsilon[\underline{a}/x_0])$, pour un "bon" choix de formule D_T représentant Dem_T, $AP \vdash (\epsilon[\underline{a}/x_0] \to \exists x_2 D_T[\underline{d}, x_2])$. On a $b = \alpha_3(\sharp\neg, d, 0)$, et

$$AP \cup \{\epsilon[\underline{a}/x_0]\} \vdash (\exists x_1 D_T[\underline{b}, x_1] \wedge \exists x_2 D_T[\underline{d}, x_2]),$$

et donc $AP \cup \{\epsilon[\underline{a}/x_0]\} \vdash \neg\mathrm{Coh}_T$. Par contraposition, on a bien $AP \vdash (\mathrm{Coh}_T \to \neg\epsilon[\underline{a}/x_0])$, et a fortiori, $T \vdash (\mathrm{Coh}_T \to \neg\epsilon[\underline{a}/x_0])$.
Comme on a vu que $T \nvdash \neg\epsilon[\underline{a}/x_0]$, $T \nvdash \mathrm{Coh}_T$. □

Remarque 5.27 *1. La formule $\epsilon[\underline{a}]$ de la démonstration signifie : "ma négation est démontrable".*

2. La preuve du deuxième théorème d'incomplétude repose sur un bon choix de formule représentant le prédicat Dem_T. Voir par exemple Logique mathématique de Cori et Lascar, tome II, chapitre 6, pour les détails.

3. Par construction, $\mathbb{N} \models Coh_T$ si et seulement si T est cohérente. Pour T cohérente, le deuxième théorème indique qu'on peut trouver un modèle $\mathcal{M}$ de $T \cup \{\neg Coh_T\}$; cela ne signifie pas pour autant que T est incohérente, il existe seulement dans $\mathcal{M}$ une "démonstration non-standard" de $0 \neq 0$.