Logique et Calculabilité
(L3 - deuxième semestre)

Serge Haddad
Professeur de l'ENS Cachan
61, Avenue du Président Wilson
94235 Cachan cedex, France
adresse électronique : haddad@lsv.ens-cachan.fr
page personnelle : www.lsv.ens-cachan.fr/~haddad/

23 mai 2008

Table des matières

Chapitre 1

Systèmes formels

1.1 Décidabilité et récursivité

L'un des objectifs de la logique est de formaliser la spécification d'un problème et sa résolution au moyen d'un système formel. Cette formalisation repose sur la notion de langage.

Définition 1 (Alphabet, Mot, Langage)
- *Un alphabet Σ est un ensemble fini ou dénombrable dont les élements sont appelés des lettres. Dans le cas où il est dénombrable, on dispose d'une procédure d'énumération des lettres.*
- *Un mot fini est une suite finie éventuellement vide de lettres. Le mot vide est noté ϵ. L'ensemble des mots est noté Σ^*.*
- *Un langage d'alphabet Σ est un sous-ensemble de Σ^*.*
- *Un mot infini est une suite infinie éventuellement vide de lettres. L'ensemble des mots infinis est noté Σ^∞.*

Soit w un mot alors on note $w = w_0 w_1 \ldots$. Pour $i \leq j$, on définit le sous-mot $w_{i:j} = w_i \ldots w_j$. Dans cette notation j peut-être égal à ∞ et dans ce cas $w_{i:\infty}$ désigne le suffixe infini démarrant à laposition i.

La notion de problème est quant à elle une notion sémantique.

Définition 2 (Notion de problème) *Un problème est une fonction f de A dans B. Dans le cas où B est le domaine booléen, on parle de problème de décision.*

Informellement un élément a de A est une donnée possible du problème et $f(a)$ est le résultat du problème pour cette donnée. En voici deux exemples simples :
- L'accessibilité dans un graphe orienté est un problème où A est l'ensemble des graphes avec deux noeuds distingués, la source et la destination, et B est l'ensemble $\{\texttt{V}, \texttt{F}\}$. f est la fonction qui renvoie `V` ssi il existe un chemin de la source vers la destination.
- Une variante de la couverture minimale d'un graphe non orienté est un problème où A est l'ensemble des graphes et B est l'ensemble des entiers naturels. f est la fonction qui associe à un graphe la cardinalité minimale

d'un sous-ensemble de sommets tels que toute arête soit adjacente à un des sommets de l'ensemble.

Les éléments de A et B disposent d'(au moins) une représentation finie susceptible d'être lue ou produite par une machine ou un programme. Cette représentation n'est rien d'autre qu'un mot d'un certain alphabet. Par exemple, les noeuds d'un graphe peuvent être numérotés à partir de 1. Dans ce cas en prenant comme alphabet $\mathbb{N} \cup \{;, -\}$, le mot $3; 1-2; 1-3$ représente le graphe des trois sommets $1, 2, 3$ et de deux arêtes, l'une joignant 1 à 2 et l'autre joignant 1 à 3.

Notre objectif est bien entendu de résoudre des problèmes à l'aide de programmes. Suivant la thèse de Church, nous ne préciserons pas le langage de programmation utilisé à partir du moment où il possède les constructions minimales : *si alors sinon, tant que, pour*, etc.

Définition 3 (Problème décidable et langage récursif) *Un problème f de A dans $\{\mathtt{V}, \mathtt{F}\}$ est décidable s'il existe un programme qui prend en entrée un élément quelconque a de A et se termine en produisant (affichant, imprimant) $f(a)$.*

Un langage d'alphabet Σ est récursif s'il existe un programme qui prend en entrée un mot de Σ^* et renvoie `V` *ssi le mot appartient au langage considéré.*

La décidabilité d'un problème est souhaitable mais on se contente d'une notion plus faible lorsque ceci est impossible.

Définition 4 (Semi-décidabilité et langage récursivement énumérable) *Un problème f de A dans $\{\mathtt{V}, \mathtt{F}\}$ est semi-décidable s'il existe un programme qui énumère les éléments a de A tels que $f(a) = \mathtt{V}$.*

Un langage d'alphabet Σ est récursivement énumérable s'il existe un programme qui énumère les mots de ce langage.

Lemme 1 *Un problème décidable est semi-décidable. Ce qui est équivalent à dire qu'un langage récursif est récursivement énumérable.*

Preuve
Le programme consiste à énumérer les mots de Σ^* ce qui est toujours possible d'après nos hypothèses. Par exemple, on exécute une boucle infinie telle qu'au $n^{ième}$ tour, on produit les mots de longueur $\leq n$ utilisant les n premières lettres de l'alphabet qui n'ont pas encore été produits. Pour chaque mot de Σ^* ainsi produit on l'affiche s'il appartient au langage (à l'aide de la procédure associée à la récursivité du langage).

c.q.f.d. $\Diamond\Diamond\Diamond$

Lemme 2 *Soit L un langage et $\overline{L}$ son complémentaire. Si L et $\overline{L}$ sont récursivement énumérables alors L et $\overline{L}$ sont récursifs.*

Autrement dit, si un problème (de décision) et son problème complémentaire sont semi-décidables alors ils sont décidables.

Preuve
Pour décider si $w \in L$, le programme consiste à alterner les pas d'exécution des deux programmes qui énumérent les mots de L et de $\overline{L}$. Lorsque w apparaît sur

une des listes (ce qui arrive nécessairement), alors l'identité du programme qui a produit la liste nous donne la réponse.

c.q.f.d. ◊◊◊

La technique usuelle pour démontrer qu'un problème est indécidable consiste à réduire un autre problème indécidable au problème initial. Cette technique suppose que l'on a préalablement déterminé d'une autre manière l'indécidabilité d'un problème. L'un de ces problèmes est celui de l'arrêt d'un programme.

Proposition 1 (Arrêt d'un programme à paramètres) *L'arrêt d'un programme prog, prenant en entrée un paramètre entier x est un problème indécidable.*

Preuve
Nous allons démontrer ce résultat par l'absurde. Supposons qu'il existe un programme *testarret* prenant en entrée deux paramètres entiers : une représentation (par un entier) d'un programme *prog* et une valeur d'entrée de ce programme. Le choix de la représentation du programme est ici sans importance ; par exemple, on pourrait choisir comme représentation le nombre entier correspondant à la suite de bits du programme (en prenant soin d'ajouter un bit de poids fort à 1 pour éviter une ambiguïté au décodage). On notera $\overline{prog}$ cette représentation. *testarret* renvoie vrai si *prog* s'arrête avec la valeur fournie et sinon renvoie faux. Le comportement de *testarret* est indéterminé si le premier paramètre n'est pas la représentation d'un programme.
Nous construisons alors un programme *fou* à un paramètre entier qui fonctionne ainsi.

- *fou* vérifie que son paramètre x est bien la représentation d'un programme *prog* (comme le fait un compilateur). Si ce n'est pas le cas, il s'arrête.
- *fou* appelle $testarret(x, x)$. Autrement dit, il teste si le programme *prog* s'arrête en prenant comme entrée sa représentation.
- Si $testarret(x, x)$ renvoie vrai, alors *fou* boucle sans fin sinon il s'arrête.

Examinons le comportement de $fou(\overline{fou})$.

- Si $fou(\overline{fou})$ s'arrête alors $testarret(\overline{fou}, \overline{fou})$ renvoie vrai et par conséquent $fou(\overline{fou})$ ne s'arrête pas ce qui est absurde.
- Dans le cas contraire, $testarret(\overline{fou}, \overline{fou})$ renvoie faux et par conséquent $fou(\overline{fou})$ s'arrête ce qui est absurde. Il n'existe donc pas de programme *testarret*.

c.q.f.d. ◊◊◊

Le fait que le programme ait un paramètre en entrée est tout à fait accessoire comme l'indique le corollaire suivant. Par ailleurs, celui-ci illustre le principe de réduction.

Corollaire 1 (Arrêt d'un programme sans paramètre) *L'arrêt d'un programme prog sans paramètre est un problème indécidable.*

Preuve
Montrons que le problème de l'arrêt d'un programme à un paramètre est réductible au problème de l'arrêt d'un programme sans paramètre. Nous supposons donc qu'il existe un programme *testarretbis* pour le problème du corollaire

et nous décrivons comment construire un programme *testarret*. Soit *prog* un programme à un paramètre et x une valeur entière. Alors *testarret* fonctionne comme suit :

- *testarret* construit la représentation du programme $prog'$ sans paramètre qui consiste à appeler $prog(x)$.
- Puis *testarret* appelle $testarretbis(\overline{prog'})$ et renvoie le résultat correspondant.

Donc *testarretbis* ne peut exister.

c.q.f.d. ◊◊◊

On a cependant le résultat plus faible suivant.

Proposition 2 *L'ensemble des programmes sans paramètre qui se terminent (vu comme un langage) est récursivement énumérable.*

Preuve
Le programme consiste en une boucle infinie. Au $n^{ième}$ tour de boucle on exécute n pas d'exécution des n premiers programmes du langage considéré et on affiche ceux qui se terminent (en éliminant ceux qui sont déjà apparus). Pour énumérer les programmes, on énumére les mots de Σ^* où Σ est l'alphabet du langage de programmation et pour chaque mot produit on appelle le compilateur du langage afin de vérifier qu'il s'agit bien d'un programme.

c.q.f.d. ◊◊◊

Corollaire 2 *L'ensemble des programmes sans paramètre qui ne se terminent pas (vu comme un langage) n'est pas récursivement énumérable.*

Preuve
Sinon d'après la proposition 2 et le lemme 2 le problème de l'arrêt des programmes serait décidable.

c.q.f.d. ◊◊◊

1.2 Récursivité dans $\mathbb{N}$

Nous formalisons dans cette section la notion de fonction récursive, de relation (et d'ensemble) récursive et récursivement énumérable dans $\mathbb{N}$.

Définition 5 *Une fonction partielle de $\mathbb{N}^p$ dans $\mathbb{N}$ est un couple (A, f) où $A \subseteq \mathbb{N}^p$ et f est une application de A dans $\mathbb{N}$. A est appelé le domaine de définition de $\mathbb{N}$.*

On dira que (A, f) est totale si son domaine de définition est égal $\mathbb{N}^p$.

On notera $\mathcal{F}_p$ l'ensemble des fonctions partielles de $\mathbb{N}^p$ dans $\mathbb{N}$. Afin d'alléger les définitions, on notera une fonction partielle (A, f) simplement par f et on dira que f est définie en $(k_1, \ldots, k_p)$ si $(k_1, \ldots, k_p) \in A$.

Les définitions qui suivent introduisent des schémas de construction de fonctions.

Définition 6 *Soient $g_1, \ldots, g_n \in \mathcal{F}_p$ et $h \in \mathcal{F}_n$, la fonction composée $f \equiv h(g_1, \ldots, g_n)$ est définie ainsi :*

- *f est définie en $(k_1, \ldots, k_p)$ ssi $\forall i \leq n, g_i(k_1, \ldots, k_p)$ est définie et $h(g_1(k_1, \ldots, k_p), \ldots, g_n(k_1, \ldots, k_p))$ est définie.*
- *Dans ce cas, $f(k_1, \ldots, k_p) = h(g_1(k_1, \ldots, k_p), \ldots, g_n(k_1, \ldots, k_p))$.*

Définition 7 *Soient $g \in \mathcal{F}_p$ et $h \in \mathcal{F}_{p+2}$, la fonction $f \equiv \mathbf{rec}(g, h) \in \mathcal{F}_{p+1}$ est définie ainsi :*

- *f est définie en $(k_1, \ldots, k_p, 0)$ ssi $g(k_1, \ldots, k_p)$ est définie et dans ce cas $f(k_1, \ldots, k_p, 0) = g(k_1, \ldots, k_p)$.*
- *f est définie en $(k_1, \ldots, k_p, k_{p+1}+1)$ ssi f est définie en $(k_1, \ldots, k_p, k_{p+1})$ et $h(k_1, \ldots, k_p, k_{p+1}, f(k_1, \ldots, k_p, k_{p+1}))$ est définie. Dans ce cas, $f(k_1, \ldots, k_p, k_{p+1} + 1) = h(k_1, \ldots, k_p, k_{p+1}, f(k_1, \ldots, k_p, k_{p+1}))$.*

Définition 8 *Soient $g \in \mathcal{F}_{p+1}$, la fonction $f \equiv \mu(g) \in \mathcal{F}_p$ est définie ainsi :*

- *f est définie en $(k_1, \ldots, k_p)$ ssi il existe un k (unique) tel que : $g(k_1, \ldots, k_p, k')$ est définie pour tout $k' \leq k$, nulle pour $k' = k$ et différente de 0 pour $k' < k$.*
- *Dans ce cas, $f(k_1, \ldots, k_p) = k$.*

Définition 9 *L'ensemble des fonctions partielles récursives est défini inductivement ainsi :*

- *Les fonction contantes totales sont récursives.*
- *Les projections (totales) pr_i^p avec $pr_i^p(k_1, \ldots, k_p) = k_i$ sont récursives.*
- *La fonction successeur (totale) s avec $s(k) = k + 1$ est récursive.*
- *Si $g_1, \ldots, g_n \in \mathcal{F}_p$ et $h \in \mathcal{F}_n$ sont récursives alors $h(g_1, \ldots, g_n)$ est récursive.*
- *Si $g \in \mathcal{F}_p$ et $h \in \mathcal{F}_{p+2}$ sont récursives alors $\mathbf{rec}(g, h)$ est récursive.*
- *Si $g \in \mathcal{F}_{p+1}$ est récursive alors $\mu(g)$ est récursive.*

Définition 10
Une relation n-aire (autrement dit un sous-ensemble de $\mathbb{N}^p$) est récursive si sa fonction caractéristique est récursive totale.
Une relation n-aire (autrement dit un sous-ensemble de $\mathbb{N}^p$) est récursivement énumérable si elle est le domaine de définition d'une fonction récursive.

1.3 Présentation des systèmes formels

Les systèmes formels ont pour objectif de transformer la résolution de problèmes de décision en une suite d'opérations « syntaxiques ». Autrement dit, dans la définition qui suit, une formule est une instance du problème considéré, un axiome est une instance dont on sait *a priori* que la réponse est vraie. Enfin une règle d'inférence permet de déduire pour une instance que la réponse est vraie sachant qu'il en est de même pour d'autres instances qui sont reliées à l'instance originelle par un prédicat ayant une signification sémantique implicite (voir la règle de modus ponens).

Définition 11 *Un système formel $\mathcal{S}$ est défini par :*

1. *$\Sigma_{\mathcal{S}}$ un alphabet fini ou dénombrable ;*

2. *$F_{\mathcal{S}}$, un sous-ensemble récursif de $\Sigma_{\mathcal{S}}^*$, appelé ensemble des formules de $\mathcal{S}$;*
3. *$A_{\mathcal{S}}$, un sous-ensemble récursif de $F_{\mathcal{S}}$, appelé ensemble des axiomes de $\mathcal{S}$;*
4. *un ensemble fini $R_{\mathcal{S}}$ de prédicats décidables définis sur $F_{\mathcal{S}}$, appelés règles d'inférence. Soit r un prédicat n+1-aire avec $n > 0$, on notera $r(f_1, \ldots, f_n, g)$ par : $[r]$: $f_1, \ldots, f_n \vdash g$*

Exemple. L'exemple qui suit est fourni uniquement à titre d'illustration car il ne présente aucun intérêt en termes de résolution de problèmes. Les nombres entiers non nuls sont représentés par des suites de 1 (*i.e.* une représentation unaire). Une formule correspond à une affirmation sur le résultat d'une addition. L'unique axiome indique que $1 + 1 = 2$. Enfin les deux règles d'inférence déduisent d'une formule, une autre formule où l'un des opérandes et le résultat sont simultanément incrémentés. Ci-dessous 1^+ représente l'ensemble des suites non nulles de 1 et 1^m dénote une suite de m « 1 ».

1. $\Sigma_{\mathcal{S}} = \{1, +, =\}$;
2. $F_{\mathcal{S}} = 1^+ + 1^+ = 1^+$
3. $A_{\mathcal{S}} = \{1 + 1 = 11\}$
4. deux règles d'inférence,
   (a) $[g]$: $1^m + 1^n = 1^p \vdash 1^{m+1} + 1^n = 1^{p+1}$
   (b) $[d]$: $1^m + 1^n = 1^p \vdash 1^m + 1^{n+1} = 1^{p+1}$

Une déduction d'un système formel s'appelle une démonstration et suit d'assez près la notion intuitive de démonstration mathématique.

Définition 12 *Une démonstration d'un système formel $\mathcal{S}$ à partir d'hypothèses $h_1, \ldots, h_m$ est une suite $f_1, \ldots, f_n$ de formules de $\mathcal{S}$ telles pour tout $1 \leq i \leq n$:*
- *Soit f_i est un axiome ;*
- *Soit $f_i = h_j$ pour un certain j ;*
- *Soit $\exists r \in R, r(f_{i_1}, \ldots, f_{i_p}, f_i)$ et $\forall 1 \leq k \leq q, i_k < i$;*

Notations. Une démonstration de f à partir d'hypothèses $h_1, \ldots, h_m$ se note $h_1, \ldots, h_m \vdash f$. On appelle *théorème* toute formule f pour laquelle il existe une démonstration sans hypothèses, *i.e.* $\vdash f$. L'ensemble des théorèmes est noté $T_{\mathcal{S}}$.

Du point de vue de la résolution de problèmes, notre objectif est de (semi-)décider si une formule est un théorème. Cependant l'intérêt des démonstrations avec hypothèses apparaîtra clairement au chapitre suivant (voir la proposition 3).

Exemple. Nous démontrons $11 + 111 = 1111$ à partir de l'hypothèse $1 + 11 = 11$.

$f_1 : 1 + 11 = 11$ $\quad [Hyp.]$
$f_2 : 1 + 111 = 111$ $\quad [g : f_1]$
$f_3 : 11 + 111 = 1111$ $\quad [d : f_2]$

Le lemme suivant explique l'intérêt des systèmes formels.

Lemme 3 *L'ensemble des démonstrations (vu comme un langage) est récursif. L'ensemble des théorèmes est récursivement énumérable même dans le cas où l'ensemble des axiomes est uniquement récursivement énumérable.*

Preuve
Pour tester si une suite de lignes est une démonstration, on vérifie la justification de toutes les lignes. Si la formule est dite être un axiome alors on peut le vérifier puisque cet ensemble est récursif. Si la formule est dite être une hypothèse alors on peut le vérifier puisque cet ensemble est fini (donc récursif). Enfin si la justification est une règle d'inférence, on peut le vérifier aussi puisque le prédicat associé est décidable.

Pour énumérer les théorèmes, il suffit d'énumérer les démonstrations. Par exemple, on exécute une boucle infinie telle qu'au $n^{ième}$ tour de boucle, on énumère les démonstrations de longueur au plus n, utilisant uniquement les n premiers axiomes (puisque ceux-ci sont récursifs, ils sont récursivement énumérables).

c.q.f.d. $\Diamond\Diamond\Diamond$

On peut généraliser la notion de démonstration d'une formule f à partir d'un ensemble H d'hypothèses quelconque ce qu'on note $H \vdash f$. Bien sûr, sans hypothèse supplémentaire sur H, le lemme précédent n'est plus valable.

1.4 Interprétation

Les systèmes formels peuvent être étudiés pour eux-mêmes. Ici nous sommes principalement intéressés au lien avec la résolution de problèmes. Ce lien est réalisé *via* la notion *d'interprétation*.

Une interprétation d'un système formel associe une valeur de vérité à chaque formule du système. Il peut y avoir plusieurs interprétations pour un même système formel. On dira qu'une formule est une *tautologie* si elle est vraie dans toutes les interprétations considérées.

Dans notre exemple, on ne considère qu'une unique interprétation la formule $1^m + 1^n = 1^p$ est vraie ssi $m + n = p$.

Une fois fixées les interprétations possibles, il y a deux propriétés qu'on souhaite voir satisfaites par un système formel. *L'adéquation* signifie qu'un théorème est une tautologie tandis que la *complétude* signifie qu'une tautologie est un théorème. En général, il est facile de vérifier l'adéquation (car elle est visée par les concepteurs du système formel) tandis que la complétude est plus difficile à obtenir.

Exemple. L'adéquation du système formel est facile à vérifier puisque l'axiome vérifie que le nombre de « 1 » à gauche et à droite du signe = est identique tandis que les deux règles d'inférence incrémentent simultanément les termes gauches et droits de l'égalité.

Ici la complétude du système formel est obtenue par induction sur le nombre de « 1 » à gauche du signe = dans une tautologie. Si ce nombre est égal à 2 (le minimum possible) alors la seule tautologie possible est l'axiome.

Pour appliquer la récurrence sur une tautologie, on remarque que dans la formule soit l'opérande gauche du + comporte plus d'un unique 1 et on applique la règle d'inférence g à partir d'une tautologie plus petite (donc un théorème par hypothèse de récurrence), soit l'opérande droite du + comporte plus d'un unique 1 et on applique la règle d'inférence d à partir d'une tautologie plus petite (donc un théorème par hypothèse de récurrence).

Chapitre 2

Calcul propositionnel

2.1 Syntaxe, démonstrations et théorèmes

Il existe de nombreuses formalisations du calcul propositionnel. Nous en choisissons une relativement économique en termes de symboles et d'axiomes. Nous en verrons d'autres ultérieurement dans le chapitre.

Définition 13 *Le calcul propositionnel est le système formel défini par :*
- *son alphabet* $\Sigma_0 = \{p_1, p_2, \ldots\} \cup \{\neg, \Rightarrow, (,)\}$
- *l'ensemble de ses formules bien formées* F_0 *défini inductivement par :*
  1. $\forall i, p_i \in F_0$
  2. $\forall A, B \in F_0, \neg A, (A \Rightarrow B) \in F_0$
- *l'ensemble de ses axiomes qui sont de trois types (avec* $A, B, C \in F_0$*)*
  (A_1) *:* $(A \Rightarrow (B \Rightarrow A))$
  (A_2) *:* $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)))$
  (A_3) *:* $((\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A))$
- *son unique règle d'inférence « modus ponens » notée m.p. :*
  $m.p. : A, (A \Rightarrow B) \vdash B$ *avec* $A, B \in F_0$

Remarques. Nous laissons le soin au lecteur de vérifier que l'ensemble des formules est récursif (par exemple en écrivant un analyseur syntaxique qui vérifie si une formule est bien formée). Cette remarque s'applique aussi à l'ensemble (infini) des axiomes et au prédicat associé à la règle d'inférence « modus ponens ».

Notation. Soit φ une formule, on note $prop(\varphi)$, l'ensemble des propositions apparaissant dans φ.

Exemple de démonstration en calcul propositionnel.

$f_1 : ((p_1 \Rightarrow ((p_1 \Rightarrow p_1) \Rightarrow p_1)) \Rightarrow ((p_1 \Rightarrow (p_1 \Rightarrow p_1)) \Rightarrow (p_1 \Rightarrow p_1)))$	$[A_2]$
$f_2 : (p_1 \Rightarrow ((p_1 \Rightarrow p_1) \Rightarrow p_1))$	$[A_1]$
$f_3 : ((p_1 \Rightarrow (p_1 \Rightarrow p_1)) \Rightarrow (p_1 \Rightarrow p_1))$	$[m.p.\ f_2, f_1]$
$f_4 : (p_1 \Rightarrow (p_1 \Rightarrow p_1))$	$[A_1]$
$f_5 : (p_1 \Rightarrow p_1)$	$[m.p.\ f_4, f_3]$

D'où le lemme suivant.

Lemme 4 *Pour tout* $A \in F_0$, $(A \Rightarrow A) \in T_0$.

Nous établissons maintenant une proposition importante reliant l'opérateur $\Rightarrow$ avec sa signification intuitive.

Proposition 3 $A_1, \ldots, A_n \vdash B$ *ssi* $A_1, \ldots, A_{n-1} \vdash (A_n \Rightarrow B)$

Cette proposition simplifie grandement l'établissement de théorèmes de T_0 et nous en donnons ci-dessous quelques uns utiles pour la suite.

Lemme 5 *Les formules suivantes sont des théorèmes de* T_0 *:*

1. $((A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)))$
2. $(B \Rightarrow ((B \Rightarrow C) \Rightarrow C))$
3. $(\neg B \Rightarrow (B \Rightarrow C))$
4. $(\neg\neg B \Rightarrow B)$
5. $(B \Rightarrow \neg\neg B)$
6. $((A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A))$
7. $(B \Rightarrow (\neg C \Rightarrow \neg(B \Rightarrow C)))$
8. $((B \Rightarrow A) \Rightarrow ((\neg B \Rightarrow A) \Rightarrow A))$

2.2 Interprétation

Une interprétation consiste d'abord à donner une valeur de vérité aux propositions puis de l'étendre aux formules en suivant la sémantique des opérateurs logiques.

Définition 14 *Une interprétation* ι *est une fonction de* $\{p_1, p_2 \ldots, \}$ *vers* $\{\mathtt{V}, \mathtt{F}\}$. *L'interprétation* ι *s'étend de manière inductive aux formules du calcul propositionnel de la façon suivante :*

- $\iota(\neg\varphi) = \mathtt{V}$ *ssi* $\iota(\varphi) = \mathtt{F}$
- $\iota(\varphi \Rightarrow \psi) = \mathtt{V}$ *ssi* $\iota(\varphi) = \mathtt{F}$ *ou* $\iota(\psi) = \mathtt{V}$

Définition 15 *Soit* Φ *un ensemble (non nécessairement fini) de formules, on dit qu'une interprétation* ι *est un* modèle *de* Φ *si* $\forall\varphi \in \Phi, \iota(\varphi) = \mathtt{V}$. Φ *est dit* satisfaisable *(ou* satisfiable *selon la littérature) s'il existe un modèle de* Φ.

On dira aussi que ι satisfait Φ si ι est un modèle de Φ.

Définition 16 *Soit* Φ *un ensemble de formules, et* φ *une formule, on dit que* φ *est une* conséquence *de* Φ *si pour tout modèle* ι *de* Φ, *on a* $\iota(\varphi) = \mathtt{V}$. *On note alors* $\Phi \models \varphi$.

Comme vue précédemment une tautologie est une formule φ telle que $\models \varphi$. Autrement dit φ est vraie pour toute interprétation. De manière évidente, $\{\varphi\}$ est satisfaisable ssi $\neg\varphi$ n'est pas une tautologie.

2.3 Compacité, adéquation et complétude

Dans cette partie, nous démontrons les principaux résultats liés au calcul propositionnel.

Proposition 4 (Compacité) *Soit Φ un ensemble de formules tel que pour tout sous-ensemble fini Φ' de Φ, Φ' soit satisfaisable. Alors Φ est satisfaisable.*

Preuve
On va construire par récurrence sur n une interprétation ι^* vérifiant l'hypothèse suivante.

> Pour tout $\Phi' \subset \Phi$ fini, il existe un modèle ι de Φ' qui coïncide avec ι^* sur $\{p_1, \ldots, p_n\}$.

Cas de base : $n = 0$. Il n'y a rien à prouver puisque c'est l'hypothèse de la proposition.

Induction : détermination de $\iota^*(p_{n+1})$
- Si pour tout $\Phi' \subset \Phi$ fini, il existe un modèle ι de Φ' qui coïncide avec ι^* sur $\{p_1, \ldots, p_n\}$ tel que $\iota(p_{n+1}) = \mathtt{F}$, on choisit $\iota^*(p_{n+1}) = \mathtt{F}$.
- Sinon on choisit $\iota^*(p_{n+1}) = \mathtt{V}$.

Prouvons que l'hypothèse de récurrence est à nouveau vérifiée. Si $\iota^*(p_{n+1}) = \mathtt{F}$ alors c'est exactement l'hypothèse qui a conduit à cette valeur. Dans le cas $\iota^*(p_{n+1}) = \mathtt{V}$. Il existe un modèle $\Phi_0 \subset \Phi$ fini, tel que pour tout modèle ι de Φ_0 qui coïncide avec ι^* sur $\{p_1, \ldots, p_n\}$, on a $\iota(p_{n+1}) = \mathtt{V}$. Soit maintenant un sous-ensemble quelconque $\Phi' \subset \Phi$ fini, on a $\Phi' \cup \Phi_0 \subset \Phi$ et $\Phi' \cup \Phi_0$ fini. *Par hypothèse de récurrence*, il existe donc un modèle ι de $\Phi' \cup \Phi_0$ qui coïncide avec ι^* sur $\{p_1, \ldots, p_n\}$; c'est aussi bien sûr un modèle de Φ_0. Par conséquent, $\iota(p_{n+1}) = \mathtt{V}$. ι est aussi un modèle de Φ', ce qui prouve que l'hypothèse de récurrence reste vérifiée.

Soit maintenant $\varphi \in \Phi$, il existe un $n \in \mathbb{N}$ tel que $prop(\varphi) \subset \{p_1, \ldots, p_n\}$. $\{\varphi\} \subset \Phi$, par conséquent en utilisant l'hypothèse de récurrence, il existe une modèle ι de $\{\varphi\}$ qui coïncide avec ι^* sur $\{p_1, \ldots, p_n\}$. Comme la valeur de vérité de φ ne dépend que de ces propositions, ι^* est aussi un modèle de $\{\varphi\}$.

c.q.f.d. ◊◊◊

Proposition 5 (Finitude) *Soit Φ un ensemble de formules et φ une formule tels que $\Phi \models \varphi$, alors il existe $\Phi' \subset \Phi$ fini tel que $\Phi' \models \varphi$.*

Preuve
L'hypothèse peut se réécrire ainsi : il n'existe pas de modèle de $\Phi \cup \{\neg\varphi\}$. En appliquant la proposition 4, on obtient l'existence d'un sous-ensemble fini $\Phi'' \subset \Phi \cup \{\neg\varphi\}$ qui n'admet pas de modèle. Soit $\Phi' = \Phi'' \setminus \{\neg\varphi\}$, on a $\Phi' \subset \Phi$. Alors $\Phi' \cup \{\neg\varphi\}$ n'admet pas de modèle. Autrement dit, φ est une conséquence de Φ'.

c.q.f.d. ◊◊◊

Lemme 6 *Soit φ un axiome du calcul propositionnel alors φ est une tautologie.*

Preuve
La preuve se conduit en décomposant selon les valeurs de vérité des sous-formules apparaissant dans l'axiome. Nous ne traitons que l'axiome $(A \Rightarrow (B \Rightarrow A))$. Soit ι une interprétation.

- Si $\iota(A) = \mathtt{V}$, alors *(par définition de l'interprétation de $\Rightarrow$)*, $\iota((B \Rightarrow A)) = \mathtt{V}$ et $\iota((A \Rightarrow (B \Rightarrow A))) = \mathtt{V}$.
- Si $\iota(A) = \mathtt{F}$, alors *(par définition de l'interprétation de $\Rightarrow$)*, $\iota((A \Rightarrow (B \Rightarrow A))) = \mathtt{V}$.

Proposition 6 (Adéquation) *Soit Φ un ensemble de formules et φ une formule, alors $\Phi \vdash \varphi$ implique $\Phi \models \varphi$.*

Preuve
Soit ι un modèle de Φ. On va prouver la proposition par récurrence sur l, la longueur d'une plus courte démonstration associée à $\Phi \vdash \varphi$.

Cas de base : $l = 1$. Soit φ un axiome, soit $\varphi \in \Phi$. Dans le premier cas, le lemme 6 permet de conclure. Dans le deuxième cas, puisque ι est un modèle de Φ, $\iota(\varphi) = \mathtt{V}$.

Induction. Nécessairement, φ est obtenue par modus ponens à partir de deux formules ψ et $\psi \Rightarrow \varphi$ qui apparaissent plus tôt dans la démonstration. *Par hypothèse de récurrence*, $\iota(\psi) = \mathtt{V}$ et $\iota(\psi \Rightarrow \varphi) = \mathtt{V}$. L'interprétation de $\Rightarrow$ implique alors que $\iota(\varphi) = \mathtt{V}$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Lemme 7 *Soit φ une formule telle que $prop(\varphi) \subset \{p_1, \ldots, p_n\}$. Soit ι une interprétation, on définit ψ_k pour $1 \leq k \leq n$ ainsi : si $\iota(p_k) = \mathtt{V}$ alors $\psi_k = p_k$ sinon $\psi_k = \neg p_k$.*
Si $\iota(\varphi) = V$ alors $\{\psi_1, \ldots, \psi_n\} \vdash \varphi$ sinon $\{\psi_1, \ldots, \psi_n\} \vdash \neg\varphi$.

Preuve
On effectue une preuve par récurrence sur la longueur l de φ.

Cas de base : $l = 1$. $\varphi = p_k$ pour un certain k. Si $\iota(\varphi) = \mathtt{V}$ (resp. $\iota(\varphi) = \mathtt{F}$), la démonstration est effectuée par l'introduction de l'hypothèse p_k (resp. $\neg p_k$).

Premier cas d'induction : $\varphi = \neg\psi$
Si $\iota(\varphi) = \mathtt{V}$ alors *par hypothèse de récurrence*, on a une démonstration de $\neg\psi$ qui est justement φ.
Si $\iota(\varphi) = \mathtt{F}$ alors *par hypothèse de récurrence*, on a une démonstration de ψ. On complète la démonstration par la démonstration de $\psi \Rightarrow \neg\neg\psi$ (établie au lemme 5) et on applique modus ponens pour obtenir $\neg\neg\psi$ qui est justement $\neg\varphi$.

Deuxième cas d'induction : $\varphi = (\psi \Rightarrow \chi)$
Si $\iota(\chi) = \mathtt{V}$ alors *par hypothèse de récurrence*, on a une démonstration de χ. On complète la démonstration par l'introduction de l'axiome $\chi \Rightarrow (\psi \Rightarrow \chi)$ et on applique modus ponens pour obtenir une démonstration de φ.

Si $\iota(\psi) = \mathtt{F}$ alors *par hypothèse de récurrence*, on a une démonstration de $\neg\psi$. On complète la démonstration par la démonstration de $\neg\psi \Rightarrow (\psi \Rightarrow \chi)$ (établie au lemme 5) et on applique modus ponens pour obtenir une démonstration de φ.

Si $\iota(\chi) = \mathtt{F}$ et $\iota(\psi) = \mathtt{V}$ alors *par hypothèse de récurrence*, on a une démonstration de $\neg\chi$ et une démonstration de ψ qu'on concatène. On complète la démonstration par la démonstration de $(\psi \Rightarrow (\neg\chi \Rightarrow \neg(\psi \Rightarrow \chi)))$ (établie au lemme 5) et on applique deux fois modus ponens pour obtenir une démonstration de $\neg\varphi$.

c.q.f.d. ◊◊◊

Proposition 7 (Complétude) *Soit Φ un ensemble de formules et φ une formule, alors $\Phi \models \varphi$ implique $\Phi \vdash \varphi$.*

Preuve
D'après la proposition 5, il existe $\{\psi_1, \ldots, \psi_m\} \subset \Phi$ tel que $\{\psi_1, \ldots, \psi_m\} \models \varphi$. En notant $\psi = (\psi_1 \Rightarrow (\psi_2 \Rightarrow \ldots (\psi_m \Rightarrow \varphi)) \ldots)$, on a donc $\models \psi$.

Soit n tel que $prop(\psi) \subset \{p_1, \ldots, p_n\}$, alors
$\forall 1 \leq i \leq n, \forall \chi_i \in \{p_i, \neg p_i\}$, $\{\chi_1, \ldots, \chi_n\} \models \psi$
(de manière évidente puisque ψ est une tautologie).

En appliquant le lemme 7, on obtient :
$\forall 1 \leq i \leq n, \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots, \chi_n\} \vdash \psi$

Nous allons démontrer par une récurrence (inversée) que :
$\forall 0 \leq n' \leq n, \forall 1 \leq i \leq n', \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots, \chi_{n'}\} \vdash \psi$

Le cas de base $n' = n$ vient d'être démontré. Supposons l'hypothèse vraie pour n' et démontrons-là pour $n' - 1$. On a :

1. $\forall 1 \leq i \leq n' - 1, \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots \chi_{n'-1}, p_{n'}\} \vdash \psi$
2. $\forall 1 \leq i \leq n' - 1, \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots \chi_{n'-1}, \neg p_{n'}\} \vdash \psi$

En utilisant la proposition 3, on obtient :

1. $\forall 1 \leq i \leq n' - 1, \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots \chi_{n'-1}\} \vdash (p_{n'} \Rightarrow \psi)$
2. $\forall 1 \leq i \leq n' - 1, \forall \chi_i \in \{p_i, \neg p_i\}, \{\chi_1, \ldots \chi_{n'-1}\} \vdash (\neg p_{n'} \Rightarrow \psi)$

On concatène les deux démonstrations, puis on insère la démonstration de $((p_{n'} \Rightarrow \psi) \Rightarrow ((\neg p_{n'} \Rightarrow \psi) \Rightarrow \psi))$ (établie au lemme 5) et on applique deux fois modus ponens pour obtenir une démonstration de $\{\chi_1, \ldots \chi_{n'-1}\} \vdash \psi$. Pour $n' = 0$, l'hypothèse de récurrence s'écrit $\vdash \psi$.

On applique alors itérativement m fois la proposition 3 pour établir :
$\{\chi_1, \ldots, \chi_n\} \vdash \varphi$
Puisque $\{\chi_1, \ldots, \chi_n\} \subset \Phi$, on en déduit que $\Phi \vdash \varphi$.

c.q.f.d. ◊◊◊

2.4 Décidabilité

Nous avons vu que la notion sémantique de conséquence et la notion syntaxique de démonstration coïncident (adéquation et complétude du calcul propositionnel). Aussi pour développer des algorithmes, on peut s'appuyer sur une

approche syntaxique ou sémantique. Nous allons illustrer notre propos en présentant le problème de la satisfaisabilité d'une formule.

Nous étudions ici une variante syntaxique de la logique des propositions avec les connecteurs $\{\neg, \wedge, \vee\}$. Tous les résultats de la section précédente restent valables (moyennant l'introduction d'autres schémas d'axiomes et une variante du modus ponens).

On rappelle l'interprétation du « et » et du « ou ». Soit ι une interprétation, alors $\iota(\varphi \wedge \psi)$ est vraie ssi $\iota(\varphi)$ et $\iota(\psi)$ sont vraies. $\iota(\varphi \vee \psi)$ est vraie ssi $\iota(\varphi)$ ou $\iota(\psi)$ sont vraies.

2.4.1 Tables de vérité

Soit φ une formule, étant donnée une interprétation, la valeur de vérité de φ ne dépend que de l'interprétation des propositions de $prop(\varphi)$.

Pour tester la satisfaisabilité, on génère toutes les interprétations possibles restreintes à ces propositions (soit $2^{Card(prop(\varphi))}$ interprétations possibles) et on s'arrête dès que l'une d'elles rend vraie φ. Dans le cas contraire, la formule n'est pas satisfaisable.

Nous illustrons notre propos sur la formule $\varphi = \neg((p_1 \wedge p_2) \vee \neg p_1)$

p_1, p_2	$p_1 \wedge p_2$	$\neg p_1$	$(p_1 \wedge p_2) \vee \neg p_1$	φ
F,F	F	V	V	F
F,V	F	V	V	F
V,F	F	F	F	V
V,V	V	F	V	F

Dans cet exemple, la quatrième ligne de la table n'aurait pas été générée.

2.4.2 Tableaux

On remarque qu'hormis lors de la phase d'évaluation, la méthode précédente ne tient pas compte de la structure de la formule mais uniquement des propositions qui y apparaissent.

La méthode des tableaux se déroule en deux étapes. Tout d'abord on *normalise* la formule en poussant les négations devant les propositions, à l'aide des équivalences suivantes :

- $\neg\neg\varphi$ se transforme en φ.
- $\neg(\varphi \wedge \psi)$ se transforme en $\neg\varphi \vee \neg\psi$.
- $\neg(\varphi \vee \psi)$ se transforme en $\neg\varphi \wedge \neg\psi$.

Par exemple, $\varphi = \neg(\neg p_1 \wedge p_2) \wedge \neg(p_3 \wedge \neg p_2)$ devient $(p_1 \vee \neg p_2) \wedge (\neg p_3 \vee p_2)$. Remarquons qu'au pire la normalisation augmente la taille de la formule d'un facteur multiplicatif de $\frac{3}{2}$.

Un tableau T est un ensemble de formules normalisées. Nous construisons (tout ou partie d')un arbre de tableaux. La racine appelée tableau initial est réduite à la formule elle-même. On construit des successeurs pour un tableau T à l'aide des deux règles présentées ci-dessous. Les feuilles de cet arbre sont :

- soit des tableaux contradictoires : ils contiennent une paire de formules p_i et $\neg p_i$.
- soit des tableaux complets : ils ne sont pas contradictoires et aucune règle ne leur est applicable.

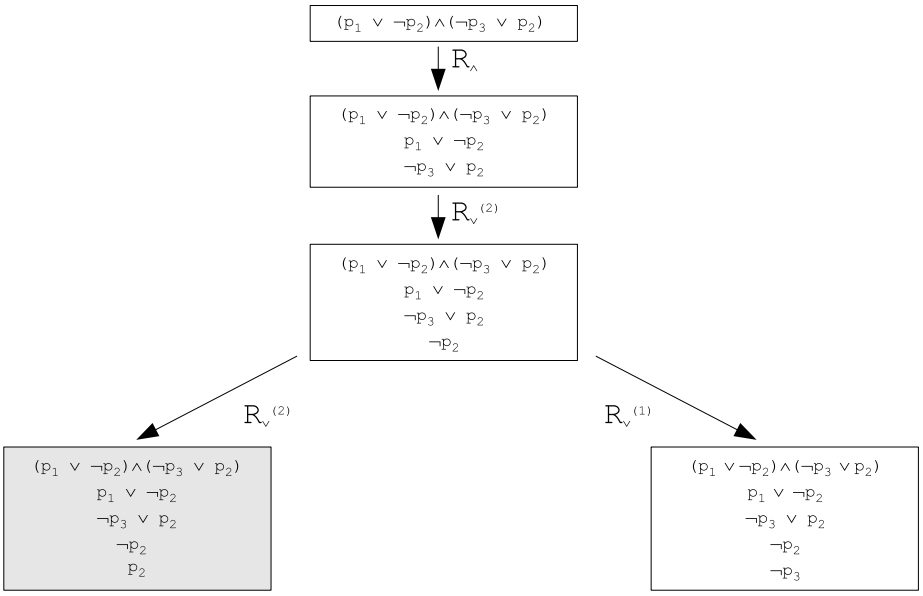

FIG. 2.1: Une partie d'un arbre de tableaux

La construction s'arrête soit lorsqu'on rencontre un tableau complet auquel cas la formule est satisfaisable, soit lorsque toutes les feuilles sont des tableaux contradictoires auquel cas la formule n'est pas satisfaisable. Il y a donc ici tout intérêt à construire l'arbre en profondeur à la recherche d'un hypothétique tableau complet. De nombreuses améliorations sont possibles mais de toute façon le problème de la satisfaisabilité en logique propositionnelle est NP-complet.

$$[R_\wedge] : \frac{T}{T \cup \{\varphi, \psi\}} \ \texttt{si}\ \varphi \wedge \psi \in T\ \texttt{et}\ \{\varphi, \psi\} \not\subset T$$

La règle associée au $\wedge$ ajoute au tableau les deux conjonctions d'une formule du tableau si elles ne sont pas déjà présentes dans le tableau.

$$[R_\vee] : \frac{T}{T \cup \{\varphi\} \mid T \cup \{\psi\}} \ \texttt{si}\ \varphi \vee \psi \in T\ \texttt{et}\ \{\varphi, \psi\} \cap T = \emptyset$$

La règle associée au $\vee$ ajoute au tableau l'une des deux disjonctions d'une formule du tableau si aucune n'est présente dans le tableau. Notons que cette règle est non déterministe ce qui explique qu'on produit un arbre.

La figure 2.1 représente une partie de l'arbre des tableaux associés à la formule qu'on a normalisée. On peut imaginer que l'arbre a été construit en profondeur d'abord et « de droite à gauche ». Lors de la construction, on a rencontré un tableau contradictoire puis une tableau complet ce qui a permis de déterminer que la formule était satisfaisable.

L'arbre des tableaux est fini car tout tableau est composé de sous-formules de φ donc il contient au plus $Card(\varphi)$ formules. Comme tout fils d'un noeud

contient au moins une formule supplémentaire, la profondeur maximale d'une branche est au plus $Card(\varphi)$. Le nombre de fils d'un noeud est d'au plus 2 en choisissant de ne justifier qu'une formule par noeud (ce qui ne modifie pas la complétude de la méthode).

Démontrons la correction de cette méthode.

Proposition 8 *Une formule (normalisée) φ est satisfaisable ssi son arbre de tableaux contient un tableau complet.*

Preuve
Condition suffisante. Soit T un tableau complet, une interprétation $\iota[T]$ qui satisfait φ est construite ainsi : si p_i, une proposition, appartient à T alors $\iota[T](p_i) = \mathtt{V}$ sinon $\iota[T](p_i) = \mathtt{F}$. Nous allons montrer par induction sur la taille de la formule que pour tout $\psi \in T$, on a $\iota[T](\psi) = \mathtt{V}$.

- Si $\psi = p_i$ alors $\iota[T](\psi) = \mathtt{V}$ par définition de $\iota[T]$.
- Si $\psi = \neg p_i$ alors puisque T n'est pas contradictoire $p_i \notin T$ et $\iota[T](\psi) = \mathtt{F}$ par définition de $\iota[T]$.
- Si $\psi = \psi_1 \wedge \psi_2$ alors puisque T est complet, $\{\psi_1, \psi_2\} \subset T$. *Par hypothèse d'induction*, $\forall i \in \{1,2\}, \iota[T](\psi_i) = \mathtt{V}$. Donc $\iota[T](\psi) = \mathtt{V}$.
- Si $\psi = \psi_1 \vee \psi_2$ alors puisque T est complet, $\{\psi_1, \psi_2\} \cap T \neq \emptyset$. Sans perte de généralité, supposons que $\psi_1 \in T$. *Par hypothèse d'induction*, $\iota[T](\psi_1) = \mathtt{V}$. Donc $\iota[T](\psi) = \mathtt{V}$.

Condition nécessaire. Soit ι une interprétation de φ telle que $\iota(\varphi) = \mathtt{V}$. On remarque d'abord qu'un tableau T tel que $\iota(\psi) = \mathtt{V}$ pour toute formule $\psi \in T$ ne peut être contradictoire. Partant du tableau initial, nous démontrons que tant que le tableau courant est incomplet, nous pouvons l'enrichir et préserver la propriété « $\iota(\psi) = \mathtt{V}$ pour toute formule $\psi \in T$ ». Puisque l'arbre des tableaux est fini, on obtient nécessairement le tableau recherché. Supposons que la règle $R_\wedge$ soit applicable à $\psi = \psi_1 \wedge \psi_2$ alors puisque $\iota(\psi) = \mathtt{V}$, $\iota(\psi_1) = \mathtt{V}$ et $\iota(\psi_2) = \mathtt{V}$. Supposons que la règle $R_\vee$ soit applicable à $\psi = \psi_1 \wedge \psi_2$ alors puisque $\iota(\psi) = \mathtt{V}$, $\iota(\psi_1) = \mathtt{V}$ ou $\iota(\psi_2) = \mathtt{V}$. Si $\iota(\psi_1) = \mathtt{V}$, on applique la règle « à gauche » sinon on applique la règle « à droite ».

c.q.f.d. ◊◊◊

2.4.3 Coupures

Nous développons maintenant une méthode que nous généraliserons au calcul des prédicats. On procède d'abord à une seconde normalisation de la formule pour la transformer en une conjonction de *clauses*. Une clause est de la forme $\bigvee_{i=1}^{m} \neg A_i \vee \bigvee_{j=1}^{n} B_j$ ce qui est sémantiquement équivalent à $\bigwedge_{i=1}^{m} A_i \Rightarrow \bigvee_{j=1}^{n} B_j$. On dit que les propositions A_i apparaissent *négativement* dans la clause tandis que les propositions B_j apparaissent *positivement*.

Cette nouvelle normalisation consiste à repousser les $\vee$ devant les propositions ou leur négation à l'aide des règles :

- $(A \wedge B) \vee C$ devient $(A \vee C) \wedge (B \vee C)$
- $A \vee (B \wedge C)$ devient $(A \vee B) \wedge (A \vee C)$

A la différence de la précédente normalisation, celle-ci peut entrainer un accroissement exponentiel de la taille de la formule.

Le principe de la méthode consiste à ajouter de nouvelles clauses à partir de clauses existantes ou à les transformer à l'aide des règles suivantes.

- Soit une clause telle que $\exists i, j, A_i = B_j$ alors on supprime la clause.
- Soit une clause telle que $A_i = A_{i'}$ pour $i < i'$ alors on supprime $A_{i'}$ dans la clause.
- Soit une clause telle que $B_j = B_{j'}$ pour $j < j'$ alors on supprime $B_{j'}$ dans la clause. Ces trois règles sont dites règles de simplification. A l'issue de l'application de ces règles, toute proposition apparaît au plus une fois dans une clause.
- Soient deux clauses $\bigvee_{i=1}^{m} \neg A_i \vee \bigvee_{j=1}^{n} B_j$ et $\bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$ telle que $\exists i^*, j^*, B_{j^*} = C_{i^*}$ alors on ajoute la clause :
  $\bigvee_{i=1}^{m} \neg A_i \vee \bigvee_{j=1, j \neq j^*}^{n} B_j \vee \bigvee_{i=1, i \neg i^*}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$
  Cette règle est dite règle de coupure.
- Soit une proposition p_k qui n'apparait que positivement dans les clauses alors on supprime les clauses où elle apparaît.
- Soit une proposition p_k qui n'apparait que négativement dans les clauses alors on supprime les clauses où elle apparaît. Ces deux règles sont dites règles d'élimination.

Proposition 9 *Soit φ une formule (normalisée) et φ' une formule obtenue à partir de φ par une règle de simplification, de coupure ou d'élimination alors φ est satisfaisable ssi φ' est satisfaisable.*

Preuve
La validité des règles de simplification repose sur les tautologies $A \vee \neg A$, $((A \vee A) \Rightarrow A)$, $(A \Rightarrow (A \vee A))$. La validité de la règle de coupure repose sur la tautologie $((A \vee B) \wedge (\neg A \vee C)) \Rightarrow (B \vee C)$. La validité des règles d'élimination repose sur le fait qu'une interprétation ι' qui rend vraie φ' peut être transformée en une interprétation ι qui rend vrai φ, identique sur toutes les propositions à ι', exceptée peut-être pour $\iota(p_k)$ rendu vrai (resp. faux) si p_k n'apparaît que positivement (resp. négativement).

c.q.f.d. $\Diamond\Diamond\Diamond$

La complétude de la méthode est prouvée par la proposition suivante.

Proposition 10 *Soit φ une formule (normalisée) non satisfaisable alors il existe une suite d'applications de règles qui produit une clause vide (*i.e. `F`*).*

Preuve
La preuve se conduit par récurrence sur le nombre nb de propositions qui apparaissent dans φ.

Cas de base : $nb = 1$. Dans ce cas, après simplification, il ne peut y avoir que deux clauses p^* et $\neg p^*$. Les deux sont forcément présentes car sinon φ serait satisfaisable. En appliquant la règle de coupure on obtient la clause vide.

Cas inductif : $nb > 1$. Choisissons une proposition p^* qui apparait dans φ. On applique les régles de simplification à p^* . Si p^* disparait il n'y a rien à prouver. Si p^* apparait toujours positivement (resp. négativement) dans les clauses alors on applique une des règles d'élimination qui fait disparaître p^*. La formule obtenue est non satisfaisable car si elle admettait un modèle, ce modèle

pourrait être transformé en modèle de φ en choisissant l'interprétation de p^* appropriée (voir le lemme précédent).

Sinon on partitionne les clauses en :

– $\mathcal{CL}^0$, l'ensemble des clauses où p^* n'apparaît pas.

– $\mathcal{CL}^+ \neq \emptyset$, l'ensemble des clauses où p^* apparaît positivement.

– $\mathcal{CL}^- \neq \emptyset$, l'ensemble des clauses où p^* apparaît négativement.

Puis on construit l'ensemble de clauses $\mathcal{CL}^{+-}$, en appliquant la règle de coupure par p^* à toutes les paires de clauses de $\mathcal{CL}^+ \times \mathcal{CL}^-$.

Remarquons que $\mathcal{CL}^{+-} \cup \mathcal{CL}^0$ est un ensemble de clauses où p^* n'apparaît plus. Pour conclure, il reste à prouver que cet ensemble de clauses n'est pas simultanément satisfaisable.

Raisonnons par l'absurde. Supposons qu'il y ait une interprétation ι' qui soit un modèle de $\mathcal{CL}^{+-} \cup \mathcal{CL}^0$. Puisque φ n'est pas satisfaisable, une clause de $\mathcal{CL}^+ \cup \mathcal{CL}^-$ n'est pas satisfaite par ι'. Nous traitons le cas où cette clause $p^* \vee \bigvee_{i=1}^m \neg A_i \vee \bigvee_{j=1}^n B_j$ appartient à $\mathcal{CL}^+$, l'autre cas se traite de manière similaire. Nécessairement $\iota'(p^*) = \mathtt{F}$. Soit maintenant ι identique à ι' excepté que $\iota(p^*) = \mathtt{V}$. ι satisfait $\mathcal{CL}^0$ puisque p^* n'apparaît pas dans ces clauses et $\mathcal{CL}^+$ puisque $\iota(p^*) = \mathtt{V}$. Soit maintenant $\neg p^* \vee \bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$ une clause quelconque de $\mathcal{CL}^-$. ι' satisfait $\bigvee_{i=1}^m \neg A_i \vee \bigvee_{j=1}^n B_j \vee \bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$. Puisque ι' ne satisfait pas $\bigvee_{i=1}^m \neg A_i \vee \bigvee_{j=1}^n B_j$, ι' satisfait $\bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$. Comme p^* n'apparait dans ce terme, ι le satisfait aussi et a fortiori la clause $\neg p^* \vee \bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$. Autrement dit ι satisfait φ ce qui établit la contradiction.

c.q.f.d. ◊◊◊

La proposition précédente nous indique la méthode à appliquer pour tester la satisfaisabilité. On élimine les propositions une par une et on conclut positivement s'il ne reste plus de clauses ou négativement si on rencontre la clause vide.

2.5 TD n°1

Question n°1. Démontrer la proposition 3.

Question n°2. Démontrer le lemme 5.

Question n°3. Normaliser la formule $\varphi = \neg(\neg p_1 \wedge p_2) \wedge \neg(p_3 \wedge \neg p_2)$.

Question n°4. Construire l'arbre de tableaux associé à la formule :
$(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge (\neg p_3 \vee p_1) \wedge (p_1 \vee p_2 \vee p_3) \wedge (\neg p_1 \vee \neg p_2 \vee \neg p_3)$
Nous avons tenu compte dans l'écriture de cette formule de l'associativité des opérateurs $\vee$ et $\wedge$.

Question n°5. Appliquer la méthode des coupures à la formule :
$(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge (\neg p_3 \vee p_1) \wedge (p_1 \vee p_2 \vee p_3) \wedge (\neg p_1 \vee \neg p_2 \vee \neg p_3)$

Chapitre 3

Calcul des prédicats du premier ordre

3.1 Syntaxe, démonstrations et théorèmes

La définition d'un système formel associé au calcul des prédicats nécessite l'introduction d'éléments préalables.

Définition 17 *Un support* $Supp = \langle Var, Cst, \{Fct_i\}_{i>0}, \{Pred_i\}_{i\geq 0}\rangle$ *d'un calcul de prédicats est défini par :*

- Var*, un ensemble dénombrable de variables.*
- Cst*, un ensemble fini ou dénombrable de constantes.*
- $\{Fct_i\}_{i>0}$*, une famille d'ensembles (disjoints) finis ou dénombrables de fonctions.* Fct_i *désigne l'ensemble des fonctions d'arité* i*. On note* $Fct = \biguplus_{i>0} Fct_i$*.*
- $\{Pred_i\}_{i\geq 0}$*, une famille d'ensembles (disjoints) finis ou dénombrables de prédicats.* $Pred_i$ *désigne l'ensemble des prédicats d'arité* i*. On note* $Pred = \biguplus_{i\geq 0} Pred_i$*.*

Tous les ensembles de la définition précédente sont supposés disjoints.

A l'aide de ces éléments, on définit d'abord des termes.

Définition 18 *Soit un support Supp, un terme est défini inductivement comme suit :*

- *Une variable ou une constante est un terme.*
- *Soit* $f \in F_i$*,* $t_1, \ldots, t_i$ *des termes alors* $f(t_1, \ldots, t_i)$ *est un terme.*

Un terme *clos* est un terme sans occurrence de variables. A partir des termes, on définit les atomes.

Définition 19 *Un atome est défini comme suit :*
Soit $p \in P_i$*,* $t_1, \ldots, t_i$ *des termes alors* $p(t_1, \ldots, t_i)$ *est un atome.*

On dit aussi formule atomique pour atome. Une formule atomique close est une formule sans occurrence de variables.

Nous sommes maintenant prêts à définir les formules du calcul des prédicats.

Définition 20 *L'ensemble des formules F_1 du calcul des prédicats associé à un support Supp est défini inductivement comme suit :*
- *Un atome de Supp est une formule de F_1.*
- *Si $A, B \in F_1$ et $x \in Var$ alors $\neg A \in F_1$, $A \Rightarrow B \in F_1$, $\forall x A \in F_1$ et $\exists x A \in F_1$.*

Deux notions fondamentales dans le calcul des prédicats sont celles d'occurrence de variable libre ou liée.

Définition 21 *Soit $f \in F_1$ les ensembles des occurrences de variables libres et des variables liées de f sont définis inductivement comme suit :*
- *Si f est une formule atomique alors l'ensemble des occurrences de variables libres est l'ensemble de ses occurrences de variables et celui des occurrences de variables liées est vide.*
- *Si $f = \neg g$ alors l'ensemble des occurrences de variables libres est celui de g et l'ensemble des occurrences des variables liées est celui de g.*
- *Si $f = g \Rightarrow h$ alors l'ensemble des occurrences de variables libres est l'union (disjointe) de ceux de g et de h et l'ensemble des occurrences des variables liées est l'union (disjointe) ceux de g et h.*
- *Si $f = \forall x g$ (resp. $f = \exists x g$) alors l'ensemble des occurrences de variables libres est celui de g excepté les occurrences de x et l'ensemble des occurrences des variables liées est l'union (disjointe) de celui de g et des occurrences libres de x dans g. Ces dernières sont dites liées par le quantificateur externe $\forall$ (resp. $\exists$).*

Une formule est dite *close* si son ensemble des occurrences de variables libres est vide. Une *théorie* est un ensemble de formules closes.

Etant donnée un formule, on va pratiquer une forme spécifique de substitution.

Définition 22 (Substitution) *On définit la substitution à la fois pour une formule propositionnelle, un terme et une formule prédicative.*
- *Soit φ une formule de F_0, $\psi_1, \ldots, \psi_n$ des formules de F_1 et $prop(\varphi) \subset \{p_1, \ldots p_n\}$. Alors $\varphi[\{p_i \leftarrow \psi_i\}_{i \in 1..n}]$ est la formule de F_1 obtenue en substituant ψ_i à chaque occurrence de p_i dans φ.*
- *Soit $t, t_1, \ldots, t_n$ des termes du calcul des prédicats et $x_1, \ldots x_n$ des variables. Alors $t[\{x_i \leftarrow t_i\}_{i \in 1..n}]$ est le terme obtenu en substituant t_i à chaque occurrence de x_i dans t.*
- *Soit φ une formule de F_1, $t_1, \ldots, t_n$ des termes du calcul des prédicats et $x_1, \ldots x_n$ des variables. Alors $\varphi[\{x_i \leftarrow t_i\}_{i \in 1..n}]$ est la formule obtenue en substituant t_i à chaque occurrence* libre *de x_i dans φ.*

Lorsqu'on procède à une seule substitution, on la note plus simplement $\varphi[p \leftarrow \psi]$ (resp. $t[x \leftarrow t'], \varphi[x \leftarrow t']$).

Exemples.
Soit $\varphi \equiv (p_1 \Rightarrow p_2)$, alors $\varphi[\{p_1 \leftarrow p(y,z), p_2 \leftarrow \exists r(z)\}] = p(y,z) \Rightarrow \exists r(z)$.
Soit $\varphi \equiv (\forall x \forall y\ p(x,y,z) \Rightarrow \exists z\ q(x,z))$, alors
$\varphi[\{x \leftarrow f(y,z), z \leftarrow y\}] = (\forall x \forall y\ p(x,y,y) \Rightarrow \exists z\ q(f(y,z),z))$.

Définition 23 *Le calcul des prédicats associé à un support Supp est le système formel défini par :*

- *son alphabet* $\Sigma_1 = Var \cup Cst \cup Fct \cup Pred \cup \{\neg, \Rightarrow, (,), \forall, \exists\}$
- *l'ensemble de ses formules bien formées* F_1 *défini ci-dessus.*
- *l'ensemble de ses axiomes qui sont de quatre types :*
  1. *Les « tautologies pseudo-propositionnelles » (dites tautologies p.p.) qui sont construites ainsi. On prend une tautologie* φ *du calcul propositionnel tel que* $prop(\varphi) \subset \{p_1, \ldots, p_n\}$ *et n formules de* F_1, $\psi_1, \ldots, \psi_n$ *et on construit la tautologie* $\varphi[\{p_i \leftarrow \psi_i\}_{i \in 1..n}]$.
  2. *Les équivalences des quantificateurs,* $(\exists x F \Rightarrow \neg\forall x \neg F)$ *et* $(\neg\forall x \neg F \Rightarrow \exists x F)$
  3. *L'inversion (limitée) du* $\forall$ *et du* $\Rightarrow$, $(\forall x(F \Rightarrow G) \Rightarrow (F \Rightarrow \forall x G))$ *à condition que* x *n'ait pas d'occurrence libre dans* F.
  4. *L'instanciation,* $(\forall x F \Rightarrow F[x \leftarrow t])$ *à condition qu'aucune occurrence de variable de* t *ne se trouve liée dans* $F[x \leftarrow t]$.
- *ses deux règles d'inférence*
  1. *La règle de modus ponens notée m.p. :*
     $m.p. : A, (A \Rightarrow B) \vdash B$ *avec* $A, B \in F_0$
  2. *La règle de généralisation notée gen :*
     $gen : A \vdash \forall x A$ *avec* $A \in F_1$ *et* $x \in Var$

Exemple. Démontrons le théorème $(\varphi[x \leftarrow t] \Rightarrow \exists x \varphi)$ avec l'hypothèse qu'aucune occurrence de variable dans t ne se trouve liée dans $\varphi[x \leftarrow t]$.
On insère d'abord l'axiome d'instanciation $(\forall x \neg\varphi \Rightarrow \neg\varphi[x \leftarrow t])$.
Puis on ajoute la tautologie p.p. :
$((\forall x \neg\varphi \Rightarrow \neg\varphi[x \leftarrow t]) \Rightarrow (\varphi[x \leftarrow t] \Rightarrow \neg\forall x \neg\varphi))$.
Par m.p., on obtient : $(\varphi[x \leftarrow t] \Rightarrow \neg\forall x \neg\varphi)$
On insère la tautologie p.p. :
$((\varphi[x \leftarrow t] \Rightarrow \neg\forall x \neg\varphi) \Rightarrow ((\neg\forall x \neg\varphi \Rightarrow \exists x \varphi) \Rightarrow (\varphi[x \leftarrow t] \Rightarrow \exists x \varphi)))$
Par m.p., on obtient :
$((\neg\forall x \neg\varphi \Rightarrow \exists x \varphi) \Rightarrow (\varphi[x \leftarrow t] \Rightarrow \exists x \varphi)))$
On insère alors l'axiome d'équivalence :
$(\neg\forall x \neg\varphi \Rightarrow \exists x \varphi)$
et on conclut par m.p.

Notation. Dans la suite, lorsqu'on complètera une démonstration, en introduisant des tautologies p.p. et des formules obtenues par M.P., on parlera de *raisonnement propositionnel*. Assez souvent, on ne détaillera pas les raisonnements propositionnels.

La proposition suivante est équivalente à la proposition 3 du calcul propositionnel.

Proposition 11 *Soit* A_n *une formule close, alors :*
$A_1, \ldots, A_n \vdash B$ *ssi* $A_1, \ldots, A_{n-1} \vdash (A_n \Rightarrow B)$

Preuve
La preuve est identique à celle de la proposition 3 excepté que dans le sens de

gauche à droite, il faut traiter le cas d'une formule dont la dernière ligne de la démonstration est obtenue par généralisation.

Soit $A_1, \ldots, A_n \vdash \forall x\ \varphi$ avec une démonstration dont la dernière ligne est une généralisation. Alors dans la démonstration, on trouve la formule φ et par hypothèse de récurrence, on a : $A_1, \ldots, A_{n-1} \vdash (A_n \Rightarrow \varphi)$. Par généralisation, on insère $\forall x(A_n \Rightarrow \varphi)$. Puis on ajoute l'axiome d'inversion $(\forall x(A_n \Rightarrow \varphi) \Rightarrow (A_n \Rightarrow \forall x\varphi))$, justifié puisque A_n est une formule close et on conclut par m.p.

c.q.f.d. $\Diamond\Diamond\Diamond$

Soit φ une formule et $\{x_{i_1}, \ldots, x_{i_k}\}$, l'ensemble des variables qui ont une occurrence libre dans φ, alors $\varphi' = \forall x_{i_1} \ldots \forall x_{i_k}\varphi$ est une *cloture universelle* de φ. Il y a autant de clotures universelles que d'énumérations des variables libres de φ. Le lemme suivant nous indique que l'on peut se restreindre aux formules closes pour la déduction.

Lemme 8 *Soit T une théorie, φ une formule et φ' une cloture universelle de φ. Alors : $T \vdash \varphi$ ssi $T \vdash \varphi'$.*

Preuve
Il suffit de prouver que $T \vdash \forall x\varphi$ ssi $T \vdash \varphi$.

Supposons que $T \vdash \forall x\varphi$. Insérons l'axiome d'instanciation $(\forall x\varphi \Rightarrow \varphi[x \leftarrow x])$ ce qui est justifié par le fait que x « se substitue » aux occurrences libres de x dans φ. On applique m.p. et on obtient une démonstration de φ.

Supposons que $T \vdash \varphi$. On obtient $T \vdash \forall x\varphi$ par la règle de généralisation.

c.q.f.d. $\Diamond\Diamond\Diamond$

Définition 24 *Une théorie T est dite cohérente ssi pour toute formule φ, soit $T \nvdash \varphi$ soit $T \nvdash \neg\varphi$.*

En réalité, une seule formule suffit pour tester la cohérence.

Lemme 9 *Une théorie T est cohérente ssi il existe une formule φ, t.q. $T \nvdash \varphi$.*

Preuve
Raisonnons par l'absurde. Supposons qu'il existe une formule ψ t.q. $T \vdash \psi$ et $T \vdash \neg\psi$. Introduisons la tautologie p.p. $(\psi \Rightarrow (\neg\psi \Rightarrow \varphi))$, Par une double application de m.p., on en conclut que contrairement à l'hypothèse $T \vdash \varphi$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Lemme 10 *Soit T une théorie et φ une formule close. Alors $T \vdash \varphi$ ssi $T \cup \{\neg\varphi\}$ est incohérente. De même, $T \vdash \neg\varphi$ ssi $T \cup \{\varphi\}$ est incohérente.*

Preuve
Supposons que $T \vdash \varphi$. Puisque $T \cup \{\neg\varphi\} \vdash \neg\varphi$, $T \cup \{\neg\varphi\}$ est incohérente.

Supposons que $T \cup \{\neg\varphi\}$ soit incohérente. En vertu du lemme 9, $T \cup \{\neg\varphi\} \vdash \varphi$. Par la proposition 11, on en déduit que : $T \vdash (\neg\varphi \Rightarrow \varphi)$. On ajoute la tautologie p.p. $((\neg\varphi \Rightarrow \varphi) \Rightarrow \varphi)$ et on conclut par m.p.

La preuve de la deuxième équivalence est similaire et laissée en exercice.

c.q.f.d. ◊◊◊

Le lemme suivant qui indique sous quelles conditions une constante peut jouer le rôle d'une variable libre nous sera fort utile pour démontrer la complétude.

Lemme 11 *Soit T une théorie, $\varphi = \forall x_i \psi$ une formule close et c une constante qui n'apparaît ni dans T ni dans ψ. Si $T \vdash \psi[x_i \leftarrow c]$ alors $T \vdash \varphi$.*

Preuve
Soit $\theta_1, \ldots, \theta_n$ la démonstration de $\psi[x_i \leftarrow c]$. Choisissons une variable x_j n'apparaisssant pas dans la démonstration. Appelons θ'_k la formule obtenue en substituant dans θ_k les occurrences de c par x_j. Alors

- Nous laissons en exercice la preuve que pour tout k, si θ_k est un axiome alors θ'_k est un axiome (l'hypothèse sur x_j n'est importante que pour l'instanciation).
- Si θ_k est une hypothèse alors $\theta'_k = \theta_k$ (en raison de l'hypothèse sur c)
- Si θ_k se déduit par m.p. à partir de θ_j et $(\theta_j \Rightarrow \theta_k)$ alors θ'_k se déduit aussi par m.p. à partir de θ'_j et $(\theta'_j \Rightarrow \theta'_k)$.
- Si $\theta_k = \forall x_i \ \theta_j$ se déduit par généralisation alors θ'_k se déduit aussi par généralisation de θ'_j.

On applique alors la règle de généralisation pour obtenir que $T \vdash \forall x_j \psi[x_i \leftarrow x_j]$, puis la règle d'instanciation avec $t = x_i$ (justifiée car x_i n'est inséré qu'à la place de ses occurrences libres dans ψ) pour obtenir que $T \vdash \psi$ et de nouveau la règle de généralisation pour conclure que $T \vdash \forall x_i \psi$.

c.q.f.d. ◊◊◊

3.2 Interprétation

Notations. Dans la suite on notera $Var = \{x_1, \ldots, x_n, \ldots\}$. $\overline{e}$ désignera une suite infinie $e_1, \ldots, e_n, \ldots$ et $\overline{e}[e', n]$ la suite obtenue à partir de $\overline{e}$ en substituant à e' à e_n.

Définition 25 *Une interprétation ι associée à un support Supp est définie par :*

- *Un ensemble non vide E_ι.*
- *Pour chaque constante $c \in Cst$, un élement $c^\iota \in E_\iota$;*
- *Pour chaque $f \in Fct_i$, une fonction f^ι de E_ι^i dans E_ι.*
- *Pour chaque $p \in Pred_i$, une fonction p^ι de E_ι^i dans $\{\mathtt{V}, \mathtt{F}\}$.*

L'interprétation d'un terme t est une fonction t^ι de E_ι^{Var} dans E_ι où $t^\iota(\overline{e})$ est défini inductivement par :

- *Si $t = c \in Cst$ alors $t^\iota(\overline{e}) = c^\iota$*
- *Si $t = x_n$ alors $t^\iota(\overline{e}) = e_n$*
- *Si $t = f(t_1, \ldots, t_n)$ alors $t^\iota(\overline{e}) = f^\iota(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$*

L'interprétation d'un atome $p(t_1, \ldots, t_i)$ est une fonction $p(t_1, \ldots, t_i)^\iota$ de E_ι^{Var} dans $\{\mathtt{V}, \mathtt{F}\}$ définie par : $p(t_1, \ldots, p_n)^\iota(\overline{e}) = p^\iota(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$.

L'interprétation d'une formule φ est alors définie inductivement comme une fonction φ^ι de E_ι^{Var} dans $\{\mathtt{V}, \mathtt{F}\}$ ainsi :

cas $\varphi = \neg\psi$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\psi^\iota(\overline{e}) = \mathtt{F}$.
cas $\varphi = \psi_1 \Rightarrow \psi_2$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\psi_1^\iota(\overline{e}) = \mathtt{F}$ *ou* $\psi_2^\iota(\overline{e}) = \mathtt{V}$.

cas $\varphi = \forall x_n \psi$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\forall e' \in E_\iota\ \psi^\iota(\overline{e}[e', n]) = \mathtt{V}$.
cas $\varphi = \exists x_n \psi$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\exists e' \in E_\iota\ \psi^\iota(\overline{e}[e', n]) = \mathtt{V}$.

Remarque. L'interprétation d'une formule close φ est une fonction constante. Par conséquent, pour toute interprétation ι, soit $\varphi^\iota = \mathtt{V}$ soit $(\neg\varphi)^\iota = \mathtt{V}$.

On dit qu'une interprétation ι est un modèle d'une théorie T si $\forall \varphi \in T, \varphi^\iota = \mathtt{V}$. Soit φ une formule, on note $T \models \varphi$ si pour tout modèle ι, φ^ι est la fonction constante $\mathtt{V}$.

Nous donnons maintenant le pendant sémantique des lemmes 8 et 10.

Lemme 12 *Soit T une théorie, φ une formule et φ' une cloture universelle de φ. Alors : $T \models \varphi$ ssi $T \models \varphi'$.*

Preuve
Il suffit de démontrer l'équivalence : $T \models \varphi$ ssi $T \models \forall x_n\ \varphi$. Soit ι un modèle de T, alors $\forall \overline{e}, \varphi^\iota(\overline{e})$ ssi $\forall \overline{e}, \forall e', \varphi^\iota(\overline{e}[e', n])$.

c.q.f.d. ◊◊◊

Lemme 13 *Soit T une théorie et φ une formule close. Alors $T \models \varphi$ ssi $T \cup \{\neg\varphi\}$ n'a pas de modèle.*

Preuve
$T \models \varphi$ ssi
pour tout modèle ι de T, $\varphi^\iota = \mathtt{V}$ ssi
pour tout modèle ι de T, $(\neg\varphi)^\iota = \mathtt{F}$ ssi
$T \cup \{\neg\varphi\}$ n'a pas de modèle.

c.q.f.d. ◊◊◊

Le lemme suivant justifie l'introduction de l'axiome d'instanciation.

Lemme 14 *Soit ϕ une formule et t un terme. Si aucune occurrence de variable de t ne se trouve liée dans $\varphi[x_n \leftarrow t]$ alors pour toute interprétation ι, pour tout $\overline{e}$, $(\varphi[x_n \leftarrow t])^\iota(\overline{e}) = \varphi^\iota(\overline{e}[t^\iota(\overline{e}), n])$.*

Preuve
On démontre d'abord par induction sur la taille d'un terme que :
$t'[x_n \leftarrow t]^\iota(\overline{e}) = t'^\iota(\overline{e}[t^\iota(\overline{e}), n])$.
Si $t' = c$ avec c une constante alors les deux termes sont égaux à c^ι.
Si $t' = x_n$ alors les deux termes sont égaux à $t^\iota(\overline{e})$.
Si $t' = x_m$ avec $m \neq n$ alors les deux termes sont égaux à e_m.
Si $t' = f(t_1, \ldots, t_m)$ alors
$t'[x_n \leftarrow t]^\iota(\overline{e})$
$= f^\iota(t_1[x_n \leftarrow t]^\iota(\overline{e}), \ldots, t_m[x_n \leftarrow t]^\iota(\overline{e}))$
$= f^\iota(t_1^\iota(\overline{e}[t^\iota(\overline{e}), n]), \ldots, t_m^\iota(\overline{e}[t^\iota(\overline{e}), n]))$ (par hypothèse de récurrence)
$= t'^\iota(e_1, \ldots, t^\iota(\overline{e}), \ldots)$

Puis la preuve se fait par induction sur le nombre d'opérateurs de la formule φ.
Si $\varphi = p(t_1, \ldots, t_m)$ est une formule atomique alors
$\varphi[x_n \leftarrow t]^\iota(\overline{e})$
$= p^\iota(t_1[x_n \leftarrow t]^\iota(\overline{e}), \ldots, t_m[x_n \leftarrow t]^\iota(\overline{e}))$

$= p^\iota(t_1^\iota(\overline{e}[t^\iota(\overline{e}), n]), \ldots, t_m^\iota(\overline{e}[t^\iota(\overline{e}), n]))$
$= \varphi^\iota(\overline{e}[t^\iota(\overline{e}), n])$

Le cas des opérateurs $\neg$ et $\Rightarrow$ est immédiat et il est laissé en exercice.

Si $\varphi = \forall x_n \psi$ alors $\varphi[x_n \leftarrow t] = \varphi$ et puisque, x_n n'ayant pas d'occurrence libre dans φ, $\varphi^\iota(\overline{e})$ est indépendant de e_n et on a $\varphi^\iota(\overline{e}) = \varphi^\iota(\overline{e}[t(\overline{e}), n])$.

Soit $\varphi = \forall x_m \psi$ $(m \neq n)$

$$(\varphi[x_n \leftarrow t])^\iota(\overline{e}) = \mathtt{V}$$
$$\text{ssi pour tout } e',\ (\psi[x_n \leftarrow t])^\iota(\overline{e}[e', m]) = \mathtt{V}$$
$$\text{ssi pour tout } e',\ \psi^\iota(\overline{e}[e', m][t(\overline{e}[e', m]), n]) = \mathtt{V}$$

(par l'hypothèse de récurrence),

$$\text{ssi pour tout } e',\ \psi^\iota(\overline{e}[e', m][t(\overline{e}), n]) = \mathtt{V}$$

(puisque par hypothèse, t n'a pas d'occurrences de x_m)

$$\text{ssi } \varphi^\iota(\overline{e}[t(\overline{e}), n]) = \mathtt{V}$$

Le cas de l'opérateur $\exists$ est similaire à l'opérateur $\forall$ et il est laissé en exercice.

c.q.f.d. $\Diamond\Diamond\Diamond$

Notons l'importance de la condition associée à l'axiome d'instanciation. Ainsi soit $\varphi = \forall x \exists y \neg p(x, y)$ et $\psi = \exists y \neg p(x, y)[x \leftarrow y] = \exists y \neg p(y, y)$. Soit ι une interprétation dans un ensemble d'au moins deux éléments où p est interprétée par l'identité. Alors $\varphi^\iota = \mathtt{V}$ et $\psi^\iota = \mathtt{F}$.

3.3 Adéquation et complétude

3.3.1 La méthode de Henkin

Lemme 15 *Soit φ un axiome du calcul des précidats alors φ est une tautologie.*

Preuve
Le cas d'une tautologie pseudo-propositionnelle φ est une conséquence de l'adéquation du calcul propositionnel.

Soit ι une interprétation.
$(\exists x_n \varphi)^\iota(\overline{e}) = \mathtt{V}$
ssi il existe un e' tel que $\varphi^\iota(\overline{e}[e', n]) = \mathtt{V}$
ssi il n'est pas vrai que pour tout e' $\neg\varphi^\iota(\overline{e}[e', n]) = \mathtt{V}$
ssi $\forall x_n \neg\varphi^\iota(\overline{e}) = \mathtt{F}$
ssi $\neg\forall x_n \neg\varphi^\iota(\overline{e}) = \mathtt{V}$
Ce qui démontre les axiomes d'équivalence.

Nous traitons maintenant l'axiome d'inversion. Soit ι une interprétation.
Supposons que $(\forall x_n(\varphi \Rightarrow \psi))^\iota(\overline{e}) = \mathtt{V}$
alors pour tout e', $(\varphi \Rightarrow \psi)^\iota(\overline{e}[e', n]) = \mathtt{V}$
Nous étudions les deux cas suivants.

- soit pour tout e', $\psi^\iota(\overline{e}[e', n]) = \mathtt{V}$
  autrement dit, $(\forall x_n \psi)^\iota(\overline{e}) = \mathtt{V}$
  et par conséquent $(\varphi \Rightarrow \forall x_n \psi)^\iota(\overline{e}) = \mathtt{V}$

- soit il existe un e' tel que $\psi^\iota(\overline{e}[e', n]) = \mathtt{F}$
  Par conséquent, $\varphi^\iota(\overline{e}[e', n]) = \mathtt{F}$
  mais puisque x_n n'a pas d'occurrence libre dans φ, $\varphi^\iota(\overline{e}) = \mathtt{F}$ et finalement

$(\varphi \Rightarrow \forall x_n \psi)^\iota(\overline{e}) = \mathtt{V}$

Nous traitons maintenant le cas de l'instanciation.
Supposons que $(\forall x_n \varphi)^\iota(\overline{e}) = \mathtt{V}$
Ce qui signifie que pour tout e', $(\varphi)^\iota(\overline{e}[e', n]) = \mathtt{V}$
Choisissons $e' = t^\iota(\overline{e})$. Alors $\varphi^\iota(\overline{e}[t^\iota(\overline{e}), n]) = \mathtt{V}$
Mais le terme gauche n'est rien d'autre que : $(\varphi[x_n \leftarrow t])^\iota(\overline{e})$ (d'après le lemme 14).
Ce qui permet de conclure.

c.q.f.d. ◊◊◊

Proposition 12 (Adéquation) *Soit T une théorie et φ une formule, alors $T \vdash \varphi$ implique $T \models \varphi$.*

Preuve
Soit ι un modèle de T. On va prouver la proposition par récurrence sur l, la longueur d'une plus courte démonstration associée à $T \vdash \varphi$.

Cas de base : $l = 1$. Soit φ un axiome, soit $\varphi \in T$. Dans le premier cas, le lemme 15 permet de conclure. Dans le deuxième cas, puisque ι est un modèle de T, $\iota(\varphi) = \mathtt{V}$.

Induction. Supposons que φ est obtenue par modus ponens à partir de deux formules ψ et $\psi \Rightarrow \varphi$ qui apparaissent plus tôt dans la démonstration. *Par hypothèse de récurrence*, $\iota(\psi) = \mathtt{V}$ et $\iota(\psi \Rightarrow \varphi) = \mathtt{V}$. L'interprétation de $\Rightarrow$ implique alors que $\iota(\varphi) = \mathtt{V}$.

Supposons maintenant que $\varphi = \forall x_n \psi$ est obtenue par généralisation à partir de ψ qui apparait plus tôt dans la démonstration. *Par hypothèse de récurrence*, $\iota(\psi) = \mathtt{V}$. Pour tout $\overline{e}$, $\iota(\psi)(\overline{e}) = \mathtt{V}$ si pour tout e', $\iota(\psi)(\overline{e}[e', n]) = \mathtt{V}$ ce qui est bien le cas.

c.q.f.d. ◊◊◊

Définition 26 *T est dite* syntaxiquement complète *ssi T est cohérente et pour tout φ formule close, $T \vdash \varphi$ ou $T \vdash \neg\varphi$.*

Définition 27 *T est dite* admettre des témoins de Henkin *ssi pour tout φ t.q. x est la seule variable qui ait des occurrences libres dans φ, il existe une constante c telle que la formule $(\exists x \varphi \Rightarrow \varphi[x \leftarrow c])$ appartienne à T.*

Proposition 13 *Soit T une théorie syntaxiquement complète et admettant des témoins de Henkin, alors T a un modèle.*

Preuve
On construit le modèle ι de T ainsi. E_ι est l'ensemble des termes clos de *Supp*. Puisqu'il y a des temoins de Henkin, $E_\iota \neq \emptyset$.
Pour c une constante, $c^\iota = c$. Pour f une fonction n-aire, $f^\iota(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$. Pour p un prédicat n-aire, $p^\iota(t_1, \ldots, t_n) = \mathtt{V}$ ssi $T \vdash p(t_1, \ldots, t_n)$.
On démontre maintenant par induction sur le nombre de connecteurs que pour toute formule close φ, $T \vdash \varphi$ ssi $\varphi^\iota = \mathtt{V}$.

Puisque T démontre toutes ses formules cela nous permettra de conclure que ι est un modèle de T.
Le cas de base, *i.e.* φ, formule atomique close, est obtenu par construction de ι. Aussi nous étudions les différents connecteurs.

- $\neg$: $T \vdash \neg\varphi$ ssi $T \nvdash \varphi$ (T syntaxiquement complète) ssi $\varphi^\iota = \mathtt{F}$ ssi $(\neg\varphi)^\iota = \mathtt{V}$
- $\Rightarrow$: Montrons que $T \vdash (\varphi \Rightarrow \psi)$ ssi $T \nvdash \varphi$ ou $T \vdash \psi$. Supposons que $T \nvdash \varphi$, alors puisque T est syntaxiquement complète $T \vdash \neg\varphi$ on insère la tautologie p.p. $(\neg\varphi \Rightarrow (\varphi \Rightarrow \psi))$ et on applique m.p. Supposons que $T \vdash \psi$, alors on insère la tautologie p.p. $(\psi \Rightarrow (\varphi \Rightarrow \psi))$ et on applique m.p.
  Supposons que $T \vdash (\varphi \Rightarrow \psi)$ et qu'on ait pas $T \nvdash \varphi$, i.e. $T \vdash \varphi$ alors on applique m.p. pour conclure que $T \vdash \psi$.

  En appliquant l'hypothèse de récurrence, $T \vdash (\varphi \Rightarrow \psi)$ ssi $\varphi^\iota = \mathtt{F}$ ou $\psi^\iota = \mathtt{V}$ ssi $(\varphi \Rightarrow \psi)^\iota = \mathtt{V}$
- $\forall$: Supposons que $T \vdash \forall x_n\ \varphi$ (avec x_n unique variable libre de φ), alors en appliquant m.p. à partir de l'axiome d'instanciation, on obtient que $T \vdash \varphi[x_n \leftarrow t]$ pour tout terme clos t. Par hypothèse de récurrence $\varphi(x_n \leftarrow t)^\iota = \mathtt{V}$ pour tout terme clos t. Or $\varphi(x_n \leftarrow t)^\iota(\overline{e}) = \varphi^\iota(\overline{e}[t, n])$. Autrement dit, $(\forall x_n\ \varphi)^\iota = \mathtt{V}$.

  Supposons que $T \nvdash \forall x_n\ \varphi$. Alors on démontre par l'absurde que $T \nvdash \forall x_n\ \neg\neg\varphi$. En effet par l'axiome d'instanciation on obtiendrait alors que $T \vdash \neg\neg\varphi$ puisque $\varphi = \varphi(x_n \leftarrow x_n)$. Puis en utilisant la tautologie p.p $(\neg\neg\varphi \Rightarrow \varphi)$ et m.p., on obtiendrait $T \vdash \varphi$ et la généralisation fournirait $T \vdash \forall x_n\ \varphi$.
  Puisque T est syntaxiquement complète, on obtient $T \vdash \neg\forall x_n\ \neg\neg\varphi$. En utilisant un axiome d'équivalence et m.p., on obtient $T \vdash \exists x_n\ \neg\varphi$. Puisque T admet des témoins de Henkin il existe une constante c t.q. $(\exists x_n\ \neg\varphi \Rightarrow \neg\varphi(x_n \leftarrow c) \in T$. Par m.p., $T \vdash \neg\varphi[x_n \leftarrow c]$. Par hypothèse de récurrence $(\neg\varphi(x_n \leftarrow c))^\iota = \mathtt{V}$. Or $\neg\varphi(x_n \leftarrow c)^\iota(\overline{e}) = \neg\varphi^\iota(\overline{e}[c, n])$. Par conséquent $(\forall x_n\ \varphi)^\iota = \mathtt{F}$.
- $\exists$: Ce cas est similaire au $\forall$ et il est laissé en exercice.

c.q.f.d. $\Diamond\Diamond\Diamond$

On dit qu'un support *Supp′ étend* un support *Supp* si les constantes (resp. variables, fonctions n-aires, prédicats n-aires) de *Supp* sont des constantes (resp. variables, fonctions n-aires, prédicats n-aires) de *Supp′*.

Proposition 14 *Soit T une théorie cohérente sur Supp alors il existe un support Supp′ qui étend Supp et T' une théorie syntaxiquement complète et admettant des témoins de Henkin telle que $T \subset T'$.*

Preuve
Le support *Supp′* est obtenu en ajoutant à l'ensemble des constantes de *Supp*, un nouvel ensemble dénombrable $\{c_0, c_1, \ldots\}$. L'ensemble des formules closes de support *Supp′* est dénombrable. Soit $\{\varphi_1, \ldots\}$ une énumération de cet ensemble, on définit par récurrence sur n une théorie T_n (avec $T_0 = T$) qui vérifie les conditions suivantes :

- T_n est cohérente.
- $T_n \subset T_{n+1}$.

- $T_n \setminus T$ est fini.
- $\varphi_m \in T_n$ ou $\neg\varphi_m \in T_n$ pour tout $0 < m \leq n$.
- Pour tout $0 < m \leq n$, si $\varphi_m \equiv \exists x_i \psi$ et $\varphi_m \in T_n$ alors il existe une constante c t.q. $\psi[x_i \leftarrow c] \in T_n$.

Le cas de base est vérifié par hypothèse puisque $T_0 = T$. Supposons que T_n est défini. Si $T_n \cup \{\neg\varphi_n\}$ est cohérente, on pose $\psi_n = \neg\varphi_n$, sinon d'après le lemme 10, on a $T \vdash \varphi_n$ et on pose $\psi_n = \varphi_n$. Dans les deux cas, $T_n \cup \{\psi_n\}$ est cohérente. Si l'opérateur externe de ψ_n n'est pas $\exists$, alors on définit $T_{n+1} = T_n \cup \{\psi_n\}$. La validité des conditions est immédiate.

Si $\psi_n = \exists x_i \psi$, on choisit un symbole c_k qui n'apparaît dans aucune formule de $T_n \cup \{\psi_n\}$ (possible puisque $T_n \setminus T$ est fini). On pose alors $T_{n+1} = T_n \cup \{\psi_n, \psi[x_i \leftarrow c_k]\}$. A l'exception de la cohérence, toutes les conditions de l'hypothèse de récurrence sont trivialement vérifiées.

Montrons par l'absurde que T_{n+1} est cohérente. Si ce n'est pas le cas, alors le lemme 10 implique que $T_n \cup \{\exists x_i \psi\} \vdash \neg\psi[x_i \leftarrow c_k]$. En raison du choix de la constante c_k, le lemme 11 entraine alors que $T_n \cup \{\exists x_i \psi\} \vdash \forall x_i \neg\psi$. En insérant la tautologie p.p. $(\exists x_i \psi \Rightarrow \neg\forall x_i \neg\psi) \Rightarrow (\forall x_i \neg\psi \Rightarrow \neg\exists x_i \psi)$, l'axiome d'équivalence sous-formule gauche de cette formule et en appliquant m.p., on conclut que $T_n \cup \{\exists x_i \psi\} \vdash \neg\exists x_i \psi$ et par conséquent que $T_n \cup \{\exists x_i \psi\}$ est incohérente ce qui est faux en raison du choix de ψ_n.

Posons maintenant $T' = \bigcup_{n \in \mathbb{N}} T_n$. Supposons que T' est incohérente. On a alors deux démonstrations de φ et de $\neg\varphi$ pour un certain φ. L'ensemble des hypothèses apparaissant dans ces démonstrations étant fini, celles-ci sont contenues dans un certain T_n. T_n serait alors incohérent ce qui est contraire à la construction. La construction implique alors que T est syntaxiquement complet.

Soit maintenant $\varphi = \exists x_i \psi$ une formule close. Si $\varphi \in T$ alors, par construction, $\psi[x_i \leftarrow c] \in T$ pour un certain T. Insérons la tautologie p.p. $(\psi[x_i \leftarrow c] \Rightarrow (\varphi \Rightarrow \psi[x_i \leftarrow c]))$, en appliquant m.p., on déduit que $T \vdash (\varphi \Rightarrow \psi[x_i \leftarrow c])$ et puisque toute formule ou sa négation appartient à T et que T est cohérente, $(\varphi \Rightarrow \psi[x_i \leftarrow c]) \in T$. Si $\varphi \notin T$ alors $\neg\varphi \in T$. Choisissons n'importe quelle constante c. Insérons la tautologie p.p. $(\neg\varphi \Rightarrow (\varphi \Rightarrow \psi[x_i \leftarrow c]))$, en appliquant m.p., on déduit que $T \vdash (\varphi \Rightarrow \psi[x_i \leftarrow c])$ et puisque toute formule ou sa négation appartient à T et que T est cohérente, $(\varphi \Rightarrow \psi[x_i \leftarrow c]) \in T$.

c.q.f.d. ◊◊◊

Proposition 15 *Soit T une théorie cohérente, alors T a un modèle.*

Preuve
D'après la proposition 14, T peut être étendue en T' une théorie syntaxiquement complète et admettant des témoins de Henkin. D'après la proposition 13, T' admet un modèle. En oubliant l'interprétation des symboles de support ajoutés, ce modèle est un modèle de T.

c.q.f.d. ◊◊◊

Corollaire 3 (Complétude) *Soit T une théorie, φ une formule close. Si $T \models \varphi$ alors $T \vdash \varphi$.*

Preuve
Si $T \models \varphi$ alors $T \cup \{\neg\varphi\}$ n'a pas de modèle (d'après le lemme 13). Donc $T \cup \{\neg\varphi\}$ est incohérente (d'après la proposition 15). Par conséquent, $T \vdash \varphi$ (d'après le lemme 10).

c.q.f.d. $\Diamond\Diamond\Diamond$

Proposition 16 (Compacité) *Soit T une théorie telle que $\forall T' \subset T$, avec T' fini, T' a un modèle. Alors T a un modèle.*

Preuve
Par l'absurde. Supposons que T n'ait pas de modèle alors T est incohérente (d'après la proposition 15). Donc $\exists\varphi, T \vdash \varphi$ et $T \vdash \neg\varphi$. Soit maintenant T' l'ensemble fini des formules de T apparaissant dans ces deux démonstrations. On a $T' \vdash \varphi$ et $T' \vdash \neg\varphi$. Donc $T' \models \varphi$ et $T' \models \neg\varphi$ (d'après la proposition 12) ce qui entraine que T' n'a pas de modèle.

c.q.f.d. $\Diamond\Diamond\Diamond$

3.3.2 Formes prénexes

On dira qu'une formule φ est *sous forme prénexe* ssi les quantificateurs se trouvent en tête de la formule (autrement dit ce sont les constructeurs les plus externes). On dira qu'elle est sous forme prénexe *polie* si de plus les quantificateurs lient des variables toutes différentes.

Lemme 16 *Soit $\forall x_i \varphi$ une formule, x_j une variable n'apparaissant pas dans φ. Alors : $\vdash \forall x_i \varphi \Leftrightarrow \forall x_j \varphi(x_i \leftarrow x_j)$.*

Lemme 17 *Soient φ, ψ des formules et x une variable. Alors :*
$\vdash (\forall x\ (\varphi \Rightarrow \psi)) \Rightarrow (\forall x\ \varphi \Rightarrow \forall x\ \psi)$ *et* $\vdash (\forall x\ (\varphi \Leftrightarrow \psi)) \Rightarrow (\forall x\ \varphi \Leftrightarrow \forall x\ \psi)$.
Si $\vdash \varphi \Leftrightarrow \psi$ et si φ est une sous-formule de θ, alors en notant θ' la formule obtenue en substituant ψ à φ dans θ on a $\vdash \theta \Leftrightarrow \theta'$.

Lemme 18 *Soient φ et ψ deux formules et x une variable alors :*

1. *Si x n'apparaît pas dans φ alors $\vdash (\varphi \Rightarrow \forall x\psi) \Leftrightarrow (\forall x(\varphi \Rightarrow \psi))$*
2. *Si x n'apparaît pas dans φ alors $\vdash (\varphi \Rightarrow \exists x\psi) \Leftrightarrow (\exists x(\varphi \Rightarrow \psi))$*
3. *Si x n'apparaît pas dans ψ alors $\vdash ((\forall x\varphi) \Rightarrow \psi) \Leftrightarrow (\exists x(\varphi \Rightarrow \psi))$*
4. *Si x n'apparaît pas dans ψ alors $\vdash ((\exists x\varphi) \Rightarrow \psi) \Leftrightarrow (\forall x(\varphi \Rightarrow \psi))$*

Proposition 17 *Soit φ une formule quelconque alors il existe une formule φ' sous forme prénexe polie qui comporte le même nombre d'opérateurs telle que $\vdash \varphi \Leftrightarrow \varphi'$*

On dira que φ est sous forme prénexe *alternée* si elle est sous forme prénexe polie et si elle alterne des quantificateurs universels et existentiels en commençant par un quantificateur universel et en terminant par un quantificateur existentiel (excepté si la suite de quantificateurs est vide).

Lemme 19 *Soit φ une formule et x une variable n'apparaissant pas dans φ. Alors $\vdash \varphi \Leftrightarrow \forall x\varphi$ et $\vdash \varphi \Leftrightarrow \exists x\varphi$*

Proposition 18 *Soit φ une formule quelconque alors il existe une formule φ' sous forme prénexe alternée telle que $\vdash \varphi \Leftrightarrow \varphi'$*

3.3.3 La méthode de Herbrand

La méthode de Herbrand (présentée ici) a pour objectif de traiter un cas particulier de la complétude. Soit φ une formule close, alors :

$$\varphi \text{ n'est pas satisfaisable ssi } \vdash \neg\varphi$$

L'intérêt de la méthode de Herbrand est de produire dans le cas d'une formule non satisfaisable une démonstration de $\neg\varphi$. Sans perte de généralité, nous supposons que φ est sous forme prénexe alternée.

Afin de faciliter la présentation cette méthode, nous supposons que l'ensemble des variables est $\{x_0, x_1, \ldots\}$ et nous nous donnons une énumération quelconque (partant de 1) des termes de notre logique et nous notons $\sharp(t)$ le nombre associé au terme t. Puis nous introduisons un fonction auxiliaire α qui associe à une suite finie de termes $(t_1, \ldots, t_i)$ un entier défini par $\alpha(t_1, \ldots, t_i) \equiv 2^m 3^{\sharp(t_1)} \ldots p_i^{\sharp(t_i)}$ où m est le plus grand indice de variable apparaissant dans $t_1, \ldots, t_i$ et p_i est le $i+1^{eme}$ nombre premier. Les propriétés intéressantes de cette fonction sont décrites dans le lemme suivant.

Lemme 20 *α vérifie les propriétés suivantes.*
- *α est injective.*
- *Si x_k apparaît dans $t_1, \ldots, t_i$ alors $k < \alpha(t_1, \ldots, t_i)$.*
- *Soit $t_1, \ldots, t_i$ et $j < i$ alors $\alpha(t_1, \ldots, t_j) < \alpha(t_1, \ldots, t_i)$.*

Preuve
(laissée en exercice)

c.q.f.d. $\Diamond\Diamond\Diamond$

Notre objectif est de transformer notre problème de satisfaisabilité de logique du premier ordre en un problème de satisfaisabilité de logique propositionnelle. La définition suivante précise quelles propositions nous avons en vue.

Définition 28 *Soit $\varphi \equiv \forall x_1 \exists x_2 \ldots \forall x_{2k-1} \exists x_{2k} \psi$ une formule sous forme prénexe alternée et $t_1, \ldots, t_k$ des termes. Alors* l'avatar *de φ associé à $t_1, \ldots, t_k$ est la formule :*

$$\psi(\{x_{2i-1} \leftarrow t_i\}_{i \leq k} \cup \{x_{2i} \leftarrow x_{\alpha(t_1,\ldots,t_i)}\}_{i \leq k})$$

At_φ est l'ensemble des atomes qui apparaissent dans un avatar de φ.

L_{At_φ} est la logique propositionnelle dont l'ensemble des propositions atomiques est At_φ.

On remarque qu'un avatar peut être vu à la fois comme une formule de logique du premier ordre et de la logique L_{At_φ}.

Le théorème suivant constitue la première partie de la preuve de complétude *via* la méthode de Herbrand.

Théorème 1 *Soit φ une formule sous forme prénexe alternée. Si l'ensemble des avatars est satisfaisable dans L_{At_φ} alors φ admet un modèle.*

Preuve
Notons ϵ l'interprétation dans L_{At_φ} qui satisfait l'ensemble des avatars de φ.

On associe à chaque variable x_i un élément x_i^*. On étend l'opération $\{\}^*$ à tout terme t, en définissant t^* comme le terme t dans lequel chaque variable est remplacée par son élément.

L'ensemble E_ι associé à notre domaine est l'ensemble des termes t^*.
Soit f une fonction n-aire, alors $f^\iota(t_1^*, \ldots, t_n^*) \equiv f(t_1^*, \ldots, t_n^*)$.

Soit p un prédicat n-aire, si $p(t_1, \ldots, t_n) \in At_\varphi$
alors $p^\iota(t_1^*, \ldots, t_n^*) = \epsilon(p(t_1, \ldots, t_n))$
sinon $p^\iota(t_1^*, \ldots, t_n^*)$ prend une valeur arbitraire.

Par construction pour un avatar quelconque et un $\overline{e}$ quelconque,
$\psi^\iota(\overline{e}[t_1^*, 1] \ldots \overline{e}[t_k^*, 2k-1]\overline{e}[x^*_{\alpha(t_1)}, 2] \ldots \overline{e}[x^*_{\alpha(t_1,\ldots,t_k)}, 2k]) = \mathtt{V}$
En effet l'interprétation s'étend de la même façon en logique propositionnelle et du premier ordre en l'absence de quantificateur et ϵ satisfait les avatars.

Montrons que $\varphi^\iota = \mathtt{V}$. Pour cela, nous démontrons par une récurrence inversée que pour tout $0 \leq i \leq k$ pour tout $\overline{e}, t_1, \ldots, t_i$,
$(\forall x_{2i+1} \exists x_{2i+2} \ldots \forall x_{2k-1} \exists x_{2k} \psi)^\iota(\overline{e}[t_1^*, 1] \ldots \overline{e}[t_i^*, 2i-1]\overline{e}[x^*_{\alpha(t_1)}, 2] \ldots \overline{e}[x^*_{\alpha(t_1,\ldots,t_i)}, 2i]) = \mathtt{V}$
(L'affectation des autres variables n'influe pas sur le résultat)
Pour $i = k$, c'est vrai puisqu'on a affaire à un avatar.
Supposons l'égalité vraie pour i et démontrons la pour $i-1$. L'égalité de la récurrence est vraie pour tout terme t_i^*. Autrement dit, pour tout terme t_i^*,
$(\exists x_{2i} \forall x_{2i+1} \exists x_{2i+2} \ldots \forall x_{2k-1} \exists x_{2k} \psi)^\iota$
$(\overline{e}[t_1^*, 1] \ldots \overline{e}[t_i^*, 2i-1]\overline{e}[x^*_{\alpha(t_1)}, 2] \ldots \overline{e}[x^*_{\alpha(t_1,\ldots,t_{i-1})}, 2i-2]) = \mathtt{V}$
Par conséquent,
$(\forall x_{2i-1} \exists x_{2i} \ldots \forall x_{2k-1} \exists x_{2k} \psi)^\iota$
$(\overline{e}[t_1^*, 1] \ldots \overline{e}[t_{i-1}^*, 2i-3]\overline{e}[x^*_{\alpha(t_1)}, 2] \ldots \overline{e}[x^*_{\alpha(t_1,\ldots,t_{i-1})}, 2i-2]) = \mathtt{V}$
Pour $i = 0$, on obtient donc que $\varphi^\iota = \mathtt{V}$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Dans cette preuve nous n'avons pas utilisé les propriétés de la fonction α que nous allons maintenant utiliser pour la deuxième partie de la preuve de la méthode de Herbrand.

Théorème 2 *Soit $\varphi \equiv \forall x_1 \exists x_2 \ldots \forall x_{2k-1} \exists x_{2k} \psi$ une formule sous forme prénexe alternée. Si l'ensemble des avatars de φ n'est pas satisfaisable dans L_{At_φ} alors $\vdash \neg\varphi$.*

Preuve
D'après le théorème de compacité de la logique propositionnelle, on en déduit qu'il existe un nombre fini d'avatars non simultanément satisfaisable dans L_{At_φ}.

Soit Φ, l'ensemble des formules

$$\forall x_{n_{2i+1}} \exists x_{n_{2i+2}} \ldots \forall x_{n_{2k-1}} \exists x_{n_{2k}}$$
$$\psi(\{x_j \leftarrow x_{n_j}\}_{2i+1 \leq j \leq 2k} \cup \{x_{2j+1} \leftarrow t_j\}_{1 \leq j < i} \cup \{x_{2j+2} \leftarrow x_{\alpha(t_1,\ldots,t_j)}\}_{1 \leq j < i})$$

pour i compris entre 0 et k, $t_1, \ldots, t_i$ des termes quelconques et $\{x_{n_j}\}_{2i+1 \leq j \leq 2k}$ des variables distinctes n'apparaissant pas dans $t_1, \ldots, t_i$ et différentes des variables de $\{x_{\alpha(t_1,\ldots,t_j)}\}_{1 \leq j < i}$.

Les avatars de φ appartiennent à Φ. Par conséquent il existe un sous-ensemble fini de formules de Φ, appelons-le Θ tel que $\vdash \bigvee_{\theta \in \Theta} \neg\theta$ (tautologie p.p.). Nous

allons quantifier peu à peu les variables libres des formules de Θ jusqu'à ce que l'on obtienne un théorème équivalent à $\neg\varphi$.

A une formule $\theta \in \Theta$, on associe $n(\theta) = \alpha(t_1, \ldots, t_i)$ (avec les notations vues ci-dessus). Remarquons que $n(\theta)$ est le plus grand indice d'une variable à occurrences libres dans θ. Remarquons aussi que si deux formules θ et θ' sont t.q. $n(\theta) = n(\theta')$ alors en vertu des propriétés de α, elles sont identiques à un renommage des variables liées près. Par conséquent à l'aide du lemme 16 et du dernier point du lemme 17, il est immédiat que $\vdash \theta \Leftrightarrow \theta'$. On élimine donc ces « duplications » de formules de Θ sans changer le fait que $\vdash \bigvee_{\theta\in\Theta} \neg\theta$.

Une fois ces duplications éliminées, il existe une unique formule $\theta_0 \in \Theta$ t.q. $n(\theta_0)$ soit maximal. Par conséquent la variable $x_{n(\theta_0)}$ n'a aucune occurrence libre dans une autre formule de Θ. Par généralisation,
$\vdash \forall x_{n(\theta_0)}(\bigvee_{\theta\in\Theta} \neg\theta)$
A l'aide de l'axiome d'inversion,
$\vdash \bigvee_{\theta\in\Theta\setminus\{\theta_0\}} \neg\theta \vee \forall x_{n(\theta_0)}\neg\theta_0$
A l'aide de l'axiome d'équivalence des quantificateurs,
$\vdash \bigvee_{\theta\in\Theta\setminus\{\theta_0\}} \neg\theta \vee \neg\exists x_{n(\theta_0)}\theta_0$

Posons $n_{2i} \equiv n(\theta_0)$. Choisissons une variable $x_{n_{2i-1}}$ qui n'a pas d'occurrence libre dans θ_0 et posons

$$\theta_1 \equiv \forall x_{n_{2i-1}} \exists x_{n_{2i}} \forall x_{n_{2i+1}} \exists x_{n_{2i+2}} \ldots \forall x_{n_{2k-1}} \exists x_{n_{2k}}$$
$$\psi(\{x_j \leftarrow x_{n_j}\}_{2i-1\leq j\leq 2k} \cup \{x_{2j+1} \leftarrow t_j\}_{1\leq j<i-1} \cup \{x_{2j+2} \leftarrow x_{\alpha(t_1,\ldots,t_j)}\}_{1\leq j<i-1})$$

En raison des contraintes sur les variables des formules de Θ, on peut appliquer l'axiome d'instanciation avec la substitution $x_{n_{2i-1}} \leftarrow t_i$ qui nous donne :
$\vdash \theta_1 \Rightarrow \exists x_{n(\theta_0)}\theta_0$
Par contraposée,
$\vdash \neg\exists x_{n(\theta_0)}\theta_0 \Rightarrow \neg\theta_1$
En remplaçant Θ par $\Theta \setminus \{\theta_0\} \cup \{\theta_1\}$, les mêmes conditions sont vérifiées avec (au moins) une variable libre éliminée.

Lorsqu'il n'y a plus de variable libre dans Θ et après élimination des duplications, Θ est réduit à une unique formule θ équivalente à φ à un renommage des variables liées près. Puisque $\vdash \neg\theta$ et $\vdash \theta \Leftrightarrow \varphi$, on a $\vdash \neg\varphi$.

c.q.f.d. $\Diamond\Diamond\Diamond$

3.3.4 Formes de Skolem

Nous avons montré comment transformer une formule en une formule équivalente (du point de vue déductif) sous forme prénexe (polie ou alternée). Nous allons poursuivre ce type de transformation. Soit une formule φ, nous allons démontrer qu'il existe une formule φ' sous forme prénexe et sans quantificateur existentiel t.q. φ est satisfaisable ssi φ' est satisfaisable. Observons qu'il s'agit d'une équivalence sémantique et notons qu'elle exige d'augmenter le support de symboles fonctionnels.

Définition 29 *Soit $\varphi \equiv Q_1x_1Q_2x_2\ldots Q_kx_k\psi$ une formule sous forme prénexe polie, t.q. les indices $n_1 < \ldots < n_l$ correspondent aux variables quantifiées existentiellement. On note $I_i = \{j \mid j < n_i \wedge \forall j', j \neq n_{j'}\}$. L'extension de support associée à φ consiste en un ensemble de fonctions $\{f_i\}_{i\leq l}$ t.q. la fonction f_i ait pour arité $n_i - i = |I_i|$.*

La forme de Skolem de φ, est définie par :

$$\varphi' \equiv \forall \ldots \psi(\{x_{n_i} \leftarrow f_i(\ldots, x_j, \ldots)\}_{i \leq l})$$

dans laquelle j parcourt I_i et les quantificateurs existentiels de φ ont été supprimés.

Exemple. Une forme de Skolem de la formule $\forall x_1\ \forall x_2\ \exists x_3\ \forall x_4\ \exists x_5\ p(x_1, x_3) \vee q(x_2, x_4, x_5)$ s'écrit $\forall x_1\ \forall x_2\ \forall x_4\ p(x_1, f_1(x_1, x_2)) \vee q(x_2, x_4, f_2(x_1, x_2, x_4))$.

Lemme 21 *Soit $\varphi \equiv Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \psi$ une formule sous forme prénexe polie et φ' une forme de Skolem de φ. Soit ι une interprétation associée au support étendu de φ. Alors $(\varphi' \Rightarrow \varphi)^\iota = \mathtt{V}$.*

Lemme 22 *Soit $\varphi \equiv Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \psi$ une formule* close *sous forme prénexe polie et φ' une forme de Skolem de φ. Soit ι une interprétation associée au support de φ. Alors il existe une interprétation ι' qui étend ι sur le support étendu t.q. $(\varphi \Rightarrow \varphi')^{\iota'} = \mathtt{V}$.*

On obtient alors la proposition suivante comme corollaire immédiat des deux lemmes précédents.

Proposition 19 *Soit φ une formule close sous forme prénexe polie et φ' une forme de Skolem de φ. Il existe un modèle de φ ssi il existe un modèle de φ'.*

3.3.5 La méthode de résolution

La méthode de résolution prend en entrée une formule close sous forme de Skolem $\varphi \equiv \forall x_1 \forall x_2 \ldots \forall x_k \psi$ où ψ est sous forme normale conjonctive, *i.e.* ψ est une conjonction de clauses.

On remarque que lorsqu'on distribue les quantificateurs devant chacune des clauses on obtient une formule équivalente et que, moyennant un renommage déja justifié précédemment, on obtient une autre formule équivalente dont les clauses ne partagent pas leurs variables. On dit qu'elles sont *séparées*. Afin d'alléger les notations on ne représente plus les quantificateurs (tous universels) devant les clauses.

Nous introduisons la notion d'unification. Soit σ une substitution de variables par des termes. Pour toute expression exp, on note $\sigma(exp)$, le résultat de l'application de σ à exp.

Définition 30 (Unification) *Une substitution σ unifie deux formules atomiques $p(t_1, \ldots, t_n)$ et $p'(t_1, \ldots, t'_{n'})$ si $\sigma(p(t_1, \ldots, t_n)) = \sigma(p'(t_1, \ldots, t'_{n'}))$.*

Naturellement, l'unification requiert que $p = p'$. Si deux atomes sont unifiables, il existe une substitution « minimale » σ t.q. toute autre substitution σ' qui unifie ces mêmes atomes peut s'exprimer $\sigma' = \mu \circ \sigma$. L'algorithme 1 explique comment construire cette unification principale. Notez que dans cet algorithme les constantes sont vues comme des fonctions 0-aires, on insère des paires de termes différents et une variable n'est le second élément d'une paire que si le premier élément est aussi une variable (les deux derniers points étant assurés par la fonction `Inserer`).

Le point clef de la méthode de résolution est la règle de résolution. Soient deux clauses $\bigvee_{i=1}^{m} \neg A_i \vee \bigvee_{j=1}^{n} B_j$ et $\bigvee_{i=1}^{m'} \neg A'_i \vee \bigvee_{j=1}^{n'} B'_j$ telle que $\exists i^*, j^*, B_{j^*}, C_{i^*}$

Algorithme 1 : Algorithme d'unification

Input : L une liste de paires de termes différents à unifier
Lorsqu'une paire contient au moins une variable,
le premier terme de la paire est toujours une variable

Output : un booléen indiquant si l'unification est possible
et la substitution associée à l'unification principale
Data : $Lsub$ une liste de subsitutions de variables par des termes

```
while L ≠ NULL do
    (t, t') ← Extraire(L)
    if t est une variable then
        if t apparaît dans t' then return F
        Appliquer({t ← t'}, Lsub)
        Inserer({t ← t'}, Lsub)
        Appliquer({t ← t'}, L)
    else
        // t ≡ f(t_1, ..., t_n)
        if t' ≡ f(t'_1, ..., t'_n) then
            for i from 1 to n do if t_i ≠ t'_i then Inserer ((t_i, t'_i), L)
        else return F
    end
end
return V, Lsub
```

qui s'unifient *via* σ. Alors on ajoute la clause produite par cette résolution :
$\bigvee_{i=1}^{m} \neg\sigma(A_i) \vee \bigvee_{j=1,\sigma(B_j)\neq\sigma(B_{j^*})}^{n} \sigma(B_j) \vee \bigvee_{i=1,\sigma(A'_i)\neq\sigma(A'_{i^*})}^{m'} \neg\sigma(A'_i) \vee \bigvee_{j=1}^{n'} \sigma(B'_j)$

On procède ensuite à un renommage pour séparer cette clause des précédentes. Le lemme suivant justifie l'emploi de cette règle. Notez que l'on n'introduit pas de règle de simplification mais qu'on procède à une simplification implicite lors de la résolution.

Lemme 23 *Soient φ et ψ deux clauses, et θ une clause obtenue par résolution de ces deux clauses. Alors tout modèle de φ et ψ est un modèle de θ.*

On introduit maintenant des modèles « universels ». Dans un tel modèle ι, l'ensemble de base est constitué par l'ensemble des termes. Les fonctions se définissent naturellement par $f^{\iota}(t_1, \ldots, t_n) \equiv f(t_1, \ldots, t_n)$ et chaque modèle est caractérisé par la valeur de vérité des formules atomiques. Comme pour la méthode de Herbrand, une clause peut s'interpréter comme une formule propositionnelle sur la logique L_{At} dont les propositions sont les atomes.

A cette fin, nous introduisons l'ensemble $\mathcal{C}(\varphi)$ où φ est une conjonction de clauses de logique de premier ordre et $\mathcal{C}(\varphi)$ est l'ensemble des clauses $\sigma(\psi)$ de L_{At} où ψ est l'une des clauses de φ et σ est une substitution des variables de ψ par des termes quelconques.

Le lemme suivant établit le lien entre la méthode des coupures et la méthode de résolution.

Lemme 24 *Soit φ, conjonction de clauses de logique de premier ordre, soit θ une clause quelconque. Si dans L_{At}, θ peut être obtenue à partir de $\mathcal{C}(\varphi)$ par la méthode des coupures, alors dans la logique du premier ordre, il existe une clause θ' obtenue à partir de φ par la méthode de résolution et une substitution σ t.q. tout atome positif (resp. négatif) de $\sigma(\theta')$ soit un atome positif (resp. négatif) de θ.*

La proposition suivante établit la complétude de la méthode de résolution.

Proposition 20 *Soit φ une conjonction de clauses (universelles) non satisfaisable alors la méthode de résolution produit la clause vide.*

3.4 Logique égalitaire

Il est naturel en logique de se doter du symbole $=$ et d'exiger que toute interprétation de ce prédicat binaire corresponde à l'égalité. On appelle interprétation *égalitaire* (resp. une modèle égalitaire d'une théorie) une interprétation (un modèle) dans laquelle le prédicat $=$ est interprété par l'identité. La question se pose alors de savoir s'il est possible de « forcer » cette restriction. En un certain sens, la réponse est positive ainsi que le montre la proposition suivante.

Proposition 21 *Soit T une théorie à partir d'un support comprenant le symbole $=$ alors il existe $T' \supset T$ une théorie telle que pour tout modèle égalitaire de T il existe un modèle de T' qui satisfait les mêmes formules close et* vice versa. *De plus, si T admet un nombre fini de fonctions et de prédicats alors $T' \setminus T$ est fini et calculable.*

Preuve
Par abus de langage, on note le prédicat $=$ sous forme infixe. La théorie T' est construite ainsi. On ajoute à T les formules suivantes :
$\forall x\ x = x$ (réflexivité de $=$)
Pour chaque fonction n-aire, f la formule :
$\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (\bigwedge_{i=1}^{n} x_i = y_i) \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
Pour chaque prédicat n-aire p (y compris $=$), la formule :
$\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (\bigwedge_{i=1}^{n} x_i = y_i \wedge p(x_1, \ldots, x_n)) \Rightarrow p(y_1, \ldots, y_n)$
Il est aisé de démontrer que cette dernière formule avec la réflexivité entraîne la symétrie et la transitivité du prédicat $=^{\iota}$ pour tout modèle ι de T'.

Condition nécessaire. Soit ι un modèle de T t.q. $=^{\iota}$ est l'égalité dans E_ι alors ι est un modèle de T' (il suffit de vérifier la validité des nouvelles formules).

Condition suffisante. Soit ι' un modèle de T'. En vertu des formules de T', $=^{\iota'}$ est une relation d'équivalence sur $E_{\iota'}$. On note $=^{\iota'}$ $(\overline{e}, \overline{f})$, si pour tout n, $=^{\iota'}$ (e_n, f_n).

On construit maintenant ι le modèle de T. E_ι est l'ensemble des classes d'équivalence de la relation $=^{\iota'}$.

Soit f une fonction n-aire, $f^{\iota}(a_1, \ldots, a_n)$ est la classe d'équivalence de $f^{\iota'}(c_1, \ldots, c_n)$ en prenant $c_i \in a_i$ quelconque. D'après la formule de T' sur les fonctions ceci ne dépend pas du choix des c_i.

Soit p un prédicat n-aire, $p^{\iota}(a_1, \ldots, a_n) = \mathtt{V}$ ssi $p^{\iota'}(c_1, \ldots, c_n) = \mathtt{V}$ en prenant $c_i \in a_i$ quelconque. D'après la formule de T' sur les prédicats ceci ne dépend pas du choix des c_i.

Démontrons par induction que pour toute formule φ, $\varphi^{\iota}(\overline{e}) = \varphi^{\iota'}(\overline{f})$ où pour tout i, $f_i \in e_i$.

On montre d'abord par induction que pour tout terme t, $t^{\iota'}(\overline{e})$ est la classe d'équivalence de $t^{\iota'}(\overline{f})$. Le cas de base (variable ou constante) est immédiat et l'induction découle de la construction de ι. Par conséquent, pour toute formule atomique φ, $\varphi(\overline{e}) = \mathtt{V}$ ssi $\varphi(\overline{f}) = \mathtt{V}$ en vertu de la de la construction de ι.

Le cas des formules générales s'obtient par induction sur les constructeurs. Seule la preuve sur les opérateurs $\forall$ et $\exists$ présente un intérêt. Nous développons celle du $\exists$, (l'autre preuve étant similaire). $(\exists x_n\ \varphi)(\overline{e}) = \mathtt{V}$ ssi il existe e' tel que $\varphi(\overline{e}[e', n]) = \mathtt{V}$ ssi il existe $e' \in E_\iota$ et il existe $f' \in e'$ tel que $\varphi(\overline{f}[f', n]) = \mathtt{V}$ (par hypothèse d'induction) ssi il existe $f' \in E_{\iota'}$ tel que $\varphi(\overline{f}[f', n]) = \mathtt{V}$ ssi $(\exists x_n\ \varphi)(\overline{f}) = \mathtt{V}$.

En appliquant ce résultat aux formules closes de T, on conclut.

c.q.f.d. $\Diamond\Diamond\Diamond$

3.5 Indécidabilité

Puisque le calcul des prédicats admet un système formel, le problème de savoir si une formule est une tautologie (*i.e.*, un théorème) est semi-décidable. Nous démontrons dans cette section, que ce problème est indécidable contrairement au calcul propositionnel.

Au premier chapitre, nous avons prouvé que l'arrêt d'un programme est un problème indécidable. Ce programme peut être exprimé dans n'importe quel langage qui peut exprimer les constructeurs `si`, `tant que` et peut incrémenter et décrémenter un compteur (ou capable de simuler ces opérations). C'est le cas des machines de Turing dont nous décrivons ci-dessous l'une des nombreuses variantes.

Définition 31 *Une machine de Turing (déterministe à une bande unidirectionnelle)* $\mathcal{M} = \langle \Sigma, Q, \delta \rangle$ *est définie par :*

- Σ*, un alphabet contenant le symbole « blanc »* $\mathtt{b}$ *et le symbole* \$ *;*
- Q*, un ensemble fini d'états contenant* q_0*, l'état initial et* q_f *l'état final ;*
- δ*, la fonction de transition de* $(Q \setminus \{q_f\}) \times \Sigma$ *dans* $Q \times \Sigma \times \{\mathtt{g}, \mathtt{d}\}$*. Afin d'assurer que la machine n'essaie jamais de déplacer sa tête de lecture en deça de sa bande on fait l'hypothèse que* $\delta(-, \$) = \langle -, \$, \mathtt{d} \rangle$ *et que* $\delta(-, \alpha) \neq \langle -, \$, - \rangle$ *pour* $\alpha \neq \$$*.*

Donnons la sémantique d'exécution d'une machine de Turing.

Définition 32 *Une configuration d'une machine de Turing est un triplet* $\langle q, w, pos \rangle$ *où* q *est l'état courant,* w *un mot infini de* Σ^∞ *représentant l'état de la bande (*w *est tel qu'il existe* $u \in (\Sigma \setminus \{\$\})^*$ *un mot fini avec* $w = \$u\mathtt{b}^\infty$*) et* $pos \in \mathbb{N}$*, la position de la tête de lecture.*

La configuration initiale de la bande est $\langle q_0, \$\mathtt{b}^\infty, 0 \rangle$*.*

Définition 33 *Soit $conf = \langle q, w, pos\rangle$, une configuration d'une machine de Turing. Alors la prochaine configuration $conf' = \langle q', w', pos'\rangle$ est obtenue comme suit.*

- *Si $q = q_f$ alors $conf' = conf$ (la machine est arrêtée).*
- *Sinon on note $\delta(q, w_{pos}) = \langle q^*, \alpha, dep\rangle$. Alors $q' = q^*$ (on change d'état), $w' = w_{1:pos-1}\alpha w_{pos+1:\infty}$ (on écrit le caractère α sur la bande). Si $dep = \mathsf{g}$ alors $pos' = pos - 1$ sinon $pos' = pos + 1$ (on déplace la tête de lecture).*

Choisissons un support *Supp* pour représenter les exécutions d'une machine de Turing en expliquant l'interprétation visée.

- La constante 0 représente à la fois le premier instant de l'exécution et la première position.
- Q et Σ sont des ensembles de constantes correspondant à leur signification originelle.
- f unaire représente la fonction « successeur » dans $\mathbb{N}$. inf est le prédicat binaire d'ordre sur les entiers.
- $state(t)$ est la fonction qui donne à l'instant t l'état de la machine.
- $tape(x, t)$ est la fonction qui donne le caractère de la bande à la position x et à l'instant t.
- $pos(t)$ est la fonction qui donne à l'instant t la position de la tête de lecture.

Nous écrivons une formule $\varphi_{\mathcal{M}}$ correspondant à une machine de Turing $\mathcal{M}$. Cette formule est une conjonction des formules suivantes.

- Une formule $q \neq q'$ par paire d'états différents et une formule $\alpha \neq \alpha'$ par paire de lettres différentes.
- Une formule pour les « entiers »
  $inf(0, f(0)) \wedge \forall x\ (inf(x, f(x)) \Rightarrow inf(f(x), f(f(x))))$
  $\wedge \forall x\ \forall y\ \forall z\ (inf(x, y) \wedge inf(x, z)) \Rightarrow inf(x, z)\ \wedge \forall x\ \forall y (inf(x, y) \Rightarrow x \neq y)$.
- Une formule pour l'état initial de la machine $state(0) = q_0 \wedge \forall x\ tape(x, 0) = \mathsf{b} \wedge pos(0) = 0$
- Une formule par transition. On note $\delta(q, \alpha) = \langle q', \alpha', dep\rangle$
  Cette formule dépend du déplacement.
  Cas $dep = g$**.** $\forall t\ state(t) = q \wedge tape(pos(t), t) = \alpha \Rightarrow$
  $state(f(t)) = q' \wedge \forall x\ (f(x) = post(t) \Rightarrow pos(f(t)) = x)$
  $\wedge tape(pos(t), f(t)) = \alpha'$
  $\wedge \forall x\ x \neq pos(t) \Rightarrow tape(x, f(t)) = tape(x, t)$
  Cas $dep = d$**.** $\forall t\ state(t) = q \wedge tape(pos(t), t) = \alpha \Rightarrow$
  $state(f(t)) = q' \wedge pos(f(t)) = f(pos(t)) \wedge tape(pos(t), f(t)) = \alpha'$
  $\wedge \forall x\ x \neq pos(t) \Rightarrow tape(x, f(t)) = tape(x, t)$
- $\forall t\ state(t) \neq q_f$, (la machine ne s'arrête jamais.)

Proposition 22 *$\varphi_{\mathcal{M}}$ admet un modèle égalitaire ssi $\mathcal{M}$ ne s'arrête pas.*

Preuve

Supposons que $\mathcal{M}$ ne s'arrête pas. On construit le modèle égalitaire ι de la formule en choisissant $E_\iota = \mathbb{N} \cup \Sigma \cup Q$ et en choisissant l'interprétation intuitive décrite ci-dessus. Pour les termes qui n'ont pas de sens, on choisit de manière arbitraire en évitant certaines valeurs. Ainsi $\forall t, t' \notin \mathbb{N}\ state^\iota(t) \notin Q \wedge f^\iota(t) \notin \mathbb{N} \wedge inf(t, t') = \mathtt{F}$, etc.

Supposons que $\varphi_{\mathcal{M}}$ ait un modèle égalitaire. On a $(f^\iota)^n(0) \neq (f^\iota)^m(0)$ pour tout $m \neq n$ en raison des formules associées au prédicat inf. On démontre par

récurrence que
$\langle state^\iota((f^\iota)^n(0)), tape^\iota(0,(f^\iota)^n(0))tape^\iota(f^\iota(0),(f^\iota)^n(0))\ldots, pos^\iota(f^n(0))\rangle$
est la configuration de la machine (sans ambiguïté en raison de la première sous-formule de $\varphi_{\mathcal{M}}$) après n pas d'exécution et par conséquent que la machine ne rencontre jamais l'état final.

c.q.f.d. $\Diamond\Diamond\Diamond$

En appliquant la proposition 21, on obtient immédiatement le corollaire suivant.

Corollaire 4 *Le problème de décider si une formule du calcul des prédicats est un théorème est indécidable.*

Preuve
Appelons φ' la conjonction de φ et des *axiomes* (finis) de l'égalité relatifs aux fonctions et prédicats apparaissant dans φ. Alors $\neg\varphi'_{\mathcal{M}}$ est un théorème ssi $\mathcal{M}$ s'arrête.

c.q.f.d. $\Diamond\Diamond\Diamond$

3.6 TD n°2

Question n°1. Démontrer le lemme 16.

Question n°2. Démontrer le lemme 17.

Question n°3. Démontrer le lemme 18.

Question n°4. Démontrer la proposition 17.

Question n°5. Démontrer le lemme 19.

Question n°6. Démontrer la proposition 18.

3.7 TD n°3

Question n°1. Démontrer le lemme 21.

Question n°2. Démontrer le lemme 22.

Question n°3. Démontrer le lemme 23.

Question n°4. Démontrer le lemme 24.

Question n°5. Démontrer la proposition 20.

Chapitre 4

Quelques théories décidables

Nous utiliserons indifféremment $\models$ ou $\vdash$ en vertu de la complétude sémantique de la logique du premier ordre.

4.1 Elimination des quantificateurs

La plupart des résultats que nous établirons dans ce chapitre reposent sur l'élimination des quantificateurs.

Définition 34 *Une théorie T permet l'élimination des quantificateurs si pour toute formule φ, il existe une formule sans quantificateur φ' t.q. $T \models \varphi \Leftrightarrow \varphi'$*

Le lemme suivant simplifie l'élimination des quantificateurs.

Lemme 25 *Soit T une théorie qui permet l'élimination des quantificateurs de toute formule $\varphi \equiv \exists x \bigwedge_{i\in I} \psi_i$ où ψ_i est soit un atome (appelé atome positif), soit la négation d'un atome (appelé atome négatif). Alors T permet l'élimination des quantificateurs.*

Preuve
On sait déjà que toute formule φ est équivalente à une formule φ' dont les opérateurs sont $\neg, \vee, \wedge, \exists$. La preuve se fait par induction sur le nombre d'opérateurs de φ'. L'hypothèse de récurrence est l'existence d'une formule équivalente sans quantificateur φ'' sous forme normale disjonctive.
Si ce nombre est nul, alors il n'y a rien à prouver. Supposons l'hypothèse de récurrence vérifiée. Les cas où le constructeur le plus externe de φ' est propositionnel se traitent avec la normalisation asssociée à la méthode des coupures (voir section 2.4.3). Par exemple, si $\varphi' \equiv \varphi_1 \wedge \varphi_2$ et si $T \models \varphi_i \Leftrightarrow \bigvee_{k\leq m_i} \psi_{i,k}$ (où les $\psi_{i,k}$ sont des clauses conjonctives) alors $T \models \varphi' \Leftrightarrow \bigvee_{k\leq m_1, l\leq m_2} \psi_{1,k} \wedge \psi_{2,l}$. Soit $\varphi' \equiv \exists x\psi$ avec $T \models \psi \Leftrightarrow \bigvee_{i\in I} \psi_i$ où les ψ_i sont des conjonctions d'atomes positifs ou négatifs. Il est immédiat que φ' est équivalente à $\bigvee_{i\in I} \exists x\psi_i$. Il suffit alors d'appliquer l'hypothèse d'élimination du lemme.

c.q.f.d. $\Diamond\Diamond\Diamond$

Nous montrons maintenant que l'élimination des quantificateurs entraine la complétude syntaxique d'une certaine théorie. A cette fin, nous introduisons deux nouvelles notions.

Notation. Dans une logique égalitaire, $\neg t = t'$ sera notée $t \neq t'$.

Définition 35 *Soit ι une interprétation d'une logique égalitaire de support Supp, alors :*

- *Le support $Supp^\iota$ est obtenu à partir de Supp en ajoutant aux constantes de Supp, les éléments de E_ι* (on fait l'hypothèse que ces éléments n'étaient pas déjà des constantes).
- *La théorie T^ι, appelée diagramme de ι est définie sur le support $Supp^\iota$ par :*
  1. *pour chaque paire d'éléments distincts de E_ι, (e, e') la formule $e \neq e'$.*
  2. *pour chaque fonction n-aire f, et chaque tuple $(e_1, \ldots, e_{n+1}) \in E_\iota^{n+1}$ la formule $e_{n+1} = f(e_1, \ldots, e_n)$ si $e_{n+1} = f^\iota(e_1, \ldots, e_n)$ ou la formule $e_{n+1} \neq f(e_1, \ldots, e_n)$ sinon.*
  3. *pour chaque prédicat n-aire p, et chaque tuple $(e_1, \ldots, e_n) \in E_\iota^n$ la formule $p(e_1, \ldots, e_n)$ si $p^\iota(e_1, \ldots, e_n) = \mathtt{V}$ ou la formule $\neg p(e_1, \ldots, e_n)$ sinon.*

Par construction, le diagramme T^ι est satisfait par l'interprétation ι étendue par $e^\iota = e$ pour tout $e \in E_\iota$.

Définition 36 *Soient ι et ι' deux interprétations sur un même support. On dit que ι' étend ι si :*

- *$E^\iota \subseteq E^{\iota'}$ et pour chaque constante c, $c^{\iota'} = c^\iota$,*
- *pour chaque fonction n-aire f, et chaque tuple $(e_1, \ldots, e_n)$ de E_ι^n, $f^{\iota'}(e_1, \ldots, e_n) = f^\iota(e_1, \ldots, e_n)$,*
- *pour chaque prédicat n-aire f, et chaque tuple $(e_1, \ldots, e_n)$ de E_ι^n, $p^{\iota'}(e_1, \ldots, e_n) = p^\iota(e_1, \ldots, e_n)$.*

Lemme 26 *Soit φ une formule sans quantificateur et ι, ι' deux interprétations t.q. ι' étend ι. Alors $\forall \overline{e} \in E_\iota^{Var}, \varphi^{\iota'}(\overline{e}) = \varphi^\iota(\overline{e})$.*

Preuve
La preuve se fait par induction sur les contructeurs de la formule. Le cas d'une formule atomique est une conséquence immédiate de la définition d'une extension.

Nous traitons le cas de la négation et laissons en exercice le cas de l'implication. Soit $\varphi \equiv \neg\psi$, soit $\overline{e} \in E_\iota^{Var}$, alors $\varphi^{\iota'}(\overline{e}) = \mathtt{V}$ ssi $\psi^{\iota'}(\overline{e}) = \mathtt{F}$ ssi (par hypothèse d'induction) $\psi^\iota(\overline{e}) = \mathtt{F}$ ssi $\varphi^\iota(\overline{e}) = \mathtt{V}$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Théorème 3 *Soit T une théorie qui permet l'élimination des quantificateurs et ι un modèle de T. Alors $T \cup T^\iota$ est une théorie syntaxiquement complète.*

Preuve
Démontrons d'abord que si T permet l'élimination des quantificateurs sur le support $Supp$, elle le permet aussi pour le support $Supp^\iota$. Soit φ une formule sur le support $Supp^\iota$, nous remplaçons dans φ chaque nouvelle constante c_i de $Supp^\iota$ qui y apparaît par une variable différente x_i absente de φ, ce qui nous donne φ'. Nous appliquons l'élimination des quantificateurs à cette formule ce qui nous

φ''. Il est alors immédiat que si on fait la substitution inverse dans φ'' la formule obtenue φ^* est équivalente à φ. En effet, $T \vdash \varphi' \Leftrightarrow \varphi''$. D'où $T \vdash \forall x_i \ldots \varphi' \Leftrightarrow \varphi''$ par généralisation. D'où $T \vdash \varphi'(\{x_i \leftarrow c_i\}) \Leftrightarrow \varphi''(\{x_i \leftarrow c_i\})$ par instanciation. Ce qui n'est rien d'autre que $T \vdash \varphi \Leftrightarrow \varphi^*$.

Soit ι' un modèle de $T \cup T^\iota$, alors ι' est (isomorphe à) une extension de ι puisqu'il est un modèle de T^ι. Soit maintenant φ une formule close de $Supp^\iota$. Puisque T permet l'élimination des quantificateurs, il existe une formule sans quantificateur φ' équivalente à φ. D'après le lemme 26, pour tout $\overline{e} \in E_\iota^{Var}$ $\varphi^{\iota'}(\overline{e}) = \varphi'^{\iota'}(\overline{e}) = \varphi'^{\iota}(\overline{e}) = \varphi^{\iota}(\overline{e})$. Mais puisque φ est une formule close, son interprétation est une constante identique pour toute interprétation. Donc soit $T \cup T^\iota \models \varphi$, soit $T \cup T^\iota \models \neg\varphi$. La complétude syntaxique s'obtient alors par la complétude (sémantique) de la logique du premier ordre.

c.q.f.d. $\Diamond\Diamond\Diamond$

4.2 Ordre dense avec premier et dernier élément

On considère le support suivant : deux constantes 0 et 1, deux prédicats binaires l'égalité (et ses axiomes présentés au chapitre précédent) et l'inégalité stricte $<$. La théorie T est définie par :

(T1) $\forall x\ \neg x < x$

(T2) $\forall x\ \forall y\ \forall z\ (x < y \wedge y < z) \Rightarrow x < z$

(T3) $\forall x\ \forall y\ x < y \vee y < x \vee x = y$

(T4) $\forall x\ \forall y\ \exists z\ x < y \Rightarrow (x < z \wedge z < y)$

(T5) $\forall x\ x = 0 \vee 0 < x$

(T6) $\forall x\ x = 1 \vee x < 1$

(T7) $0 \neq 1$

Les formules $T1, T2, T3$ sont les axiomes de l'ordre total. La formule $T4$ exprime la densité. Les formules $T5$ et $T6$ définissent les premiers et derniers éléments. $T7$ précise que le premier et le dernier élément sont différents.

Théorème 4 *La théorie T de l'ordre dense avec premier et dernier élément permet l'élimination des quantificateurs.*

Preuve
Intéressons-nous aux atomes positifs ou négatifs. L'atome $\neg t_1 < t_2$ peut être remplacé par la formule $t_2 < t_1 \vee t_1 = t_2$ en vertu de $T3$. On peut donc supposer que les atomes sont de la forme $t_1 < t_2, t_1 = t_2, t_1 \neq t_2$.

Soit une formule $\varphi \equiv \exists x\ \bigwedge_{1 \leq i \leq r} \psi_i$. On opère par récurrence sur r. Dans le cas $r = 1$, si x n'apparait pas dans ψ_1, l'élimination est immédiate. Nous traitons quelques cas significatifs de formules ψ_1 et laissons les autres en exercice. Si $\psi_1 \equiv x < 1$ alors φ est équivalente à `V` en vertu de $T6$ et $T7$. Si $\psi_1 \equiv 1 < x$ alors φ est équivalente à `F` en vertu de $T1$, $T2$ et $T6$. Si $\psi_1 \equiv x = y$ alors φ est équivalente à `V` en vertu des axiomes de l'égalité. Si $\psi_1 \equiv x \neq y$ alors φ est équivalente à `V` en vertu de $T1$ et $T7$.

Nous traitons l'étape d'induction. Si x n'apparaît pas dans un atome, alors cet atome peut être déplacé à l'extérieur de la portée du quantificateur. On considère

donc une formule $\varphi \equiv \exists x \ \bigwedge_{1\leq i\leq k} x < t_i \wedge \bigwedge_{1\leq i\leq l} u_i < x \wedge \bigwedge_{1\leq i\leq m} x = v_i$ Si l'un des termes t_i, u_i, v_i est x alors soit on élimine le terme (cas $x = x$) soit la formule est équivalente à F (cas $x < x$).

Si $k > 1$ alors φ est équivalente à
$(t_1 < t_2 \wedge \exists x \ \bigwedge_{1\leq i\leq k, i\neq 2} x < t_i \wedge \bigwedge_{1\leq i\leq l} u_i < x \wedge \bigwedge_{1\leq i\leq m} x = v_i)$
$\vee(\neg t_1 < t_2 \wedge \exists x \ \bigwedge_{2\leq i\leq k} x < t_i \wedge \bigwedge_{1\leq i\leq l} u_i < x \wedge \bigwedge_{1\leq i\leq m} x = v_i)$
Le cas $l > 1$ se traite de manière analogue.

Si $k = l = 1$ et $m > 0$, φ est équivalente à $v_1 = \ldots = v_m \wedge u_1 < v_1 < t_1$. Si $k = l = 1$ et $m = 0$, φ est équivalente à $u_1 < t_1$. Les derniers cas se traitent de manière analogue.

c.q.f.d. ◊◊◊

Puisque les transformations sont effectives, nous avons établi le corollaire suivant.

Corollaire 5 *Le problème de savoir si φ est une conséquence de la théorie T de l'ordre dense avec premier et dernier élément est décidable.*

Preuve
On peut considérer que φ est une formule close en raison du lemme 8. La transformation ne crée pas de variables donc la formule équivalente est une combinaison booléenne des formules $0 < 1$, $1 < 0$ et $0 = 1$ respectivement équivalentes à V, F, F. L'évaluation de cette expression booléenne permet de conclure.

c.q.f.d. ◊◊◊

Remarque. L'intervalle réel $[0, 1]$ est un modèle de T de même que l'ensemble des rationnels de cet intervalle. Autrement dit, ces deux modèles bien que non isomorphes sont indiscernables par les formules de cette théorie.

4.3 Ordre discret sans premier ni dernier élément

On considère le support suivant : une fonction unaire s, deux prédicats binaires l'égalité (et ses axiomes présentés au chapitre précédent) et l'inégalité stricte $<$. La théorie T est définie par :

(T1) $\forall x \ \neg x < x$

(T2) $\forall x \ \forall y \ \forall z \ (x < y \wedge y < z) \Rightarrow x < z$

(T3) $\forall x \ \forall y \ x < y \vee y < x \vee x = y$

(T4) $\forall x \ \forall y \ \ x < y \Leftrightarrow (y = s(x) \vee s(x) < y)$

(T5) $\forall x \ \exists y \ x = s(y)$

Les formules $T1, T2, T3$ sont les axiomes de l'ordre total. La formule $T4$ exprime le caractère discret de l'ordre. La formule $T5$ garantit l'existence d'un prédécesseur.

Observation. Naturellement $\mathbb{Z}$ est un modèle de T. Nous laissons en exercice la recherche d'autres modèles de T.

Afin d'alléger les formules, on note $s^p t$ le terme obtenu par p applications de la fonction successeur à t.

Théorème 5 *La théorie T de l'ordre discret sans premier ni dernier élément permet l'élimination des quantificateurs.*

Preuve
Intéressons-nous aux atomes positifs ou négatifs. L'atome $\neg t_1 < t_2$ peut être remplacé par la formule $t_2 < t_1 \vee t_1 = t_2$ en vertu de $T3$. On peut donc supposer que les atomes sont de la forme $t_1 < t_2, t_1 = t_2, t_1 \neq t_2$.

Soit une formule $\varphi \equiv \exists x \bigwedge_{1 \leq i \leq r} \psi_i$. On opère par récurrence sur r. Dans le cas $r = 1$, si x n'apparait pas dans ψ_1, l'élimination est immédiate. Nous traitons quelques cas significatifs de formules ψ_1 et laissons les autres en exercice. Si $\psi_1 \equiv x < s^p x$ avec $p \geq 1$ alors φ est équivalente à `V` en vertu de $T4$. Si $\psi_1 \equiv x < s^p y$ avec $p \geq 0$ alors φ est équivalente à `V` en vertu de $T4$ et $T5$. Si $\psi_1 \equiv x \neq y$ alors φ est équivalente à `V` en vertu de $T1$, $T4$ et $T5$.

Nous traitons l'étape d'induction. Si x n'apparaît pas dans un atome, alors cet atome peut être déplacé à l'extérieur de la portée du quantificateur. Si x apparaît dans le deux termes de l'atome, l'élimination est immédiate. Par exemple, $s^p x < s^q x$ peut être supprimé si $p < q$. Dans le cas contraire, φ est équivalente à `F`.

Afin de simplifier la suite du raisonnement, on écrit parfois un terme $s^p y \sim t$ (resp. $t \sim s^p y$) sous la forme $y \sim s^{-p} t$ (resp. $s^{-p} t \sim y$). On considère donc une formule $\varphi \equiv \exists x \bigwedge_{1 \leq i \leq k} x < t_i \wedge \bigwedge_{1 \leq i \leq l} u_i < x \wedge \bigwedge_{1 \leq i \leq m} x = v_i$ où les termes t_i, u_i, v_i sont de la forme $s^p y$ et $p \in \mathbb{Z}$.

Si $k > 1$ alors φ est équivalente à
$(t_1 < t_2 \wedge \exists x \bigwedge_{1 \leq i \leq k, i \neq 2} x < t_i \wedge \bigwedge_{1 \leq i \leq l} u_i < x \wedge \bigwedge_{1 \leq i \leq m} x = v_i)$
$\vee (\neg t_1 < t_2 \wedge \exists x \bigwedge_{2 \leq i \leq k} x < t_i \wedge \bigwedge_{1 \leq i \leq l} u_i < x \wedge \bigwedge_{1 \leq i \leq m} x = v_i)$
Le cas $l > 1$ se traite de manière analogue.

Si $k = l = 1$ et $m > 0$, φ est équivalente à $v_1 = \ldots = v_m \wedge u_1 < v_1 < t_1$. Si $k = l = 1$ et $m = 0$, φ est équivalente à $u_1 < t_1$. Les derniers cas se traitent de manière analogue.

c.q.f.d. $\Diamond\Diamond\Diamond$

Puisque les transformations sont effectives, nous avons établi le corollaire suivant.

Corollaire 6 *Le problème de savoir si φ est une conséquence de la théorie T de l'ordre discret sans premier ni dernier élément est décidable.*

Preuve
On peut considérer que φ est une formule close en raison du lemme 8. La transformation ne crée pas de variables donc la formule équivalente est nécessairement équivalente à `V` ou à `F`.

c.q.f.d. $\Diamond\Diamond\Diamond$

4.4 Groupes commutatifs ordonnés discrets

On considère le support suivant : deux constantes 0 et 1, une fonction unaire $-$, une fonction binaire $+$, un prédicat unaire >0 représentant le caractère positif et l'égalité. Le terme $1 + \ldots + 1$ (resp. $t + \ldots + t$), n fois, est noté n (resp. nt). Le terme $t + (-t')$ est noté $t - t'$. La théorie T est définie par :

(T1) $\forall x\ x + 0 = x$

(T2) $\forall x\ x - x = 0$

(T3) $\forall x\ \forall y\ x + y = y + x$

(T4) $\forall x\ \forall y\ \forall z\ (x + y) + z = x + (y + z)$

(T5) $\forall x\ x = 0 \vee x > 0 \vee -x > 0$

(T6) $\forall x\ \neg(x > 0 \wedge -x > 0)$

(T7) $\forall x\ \forall y\ (x > 0 \wedge y > 0) \Rightarrow x + y > 0$

(T8) $\forall x\ x > 0 \Leftrightarrow (x = 1 \vee x - 1 > 0)$

Les formules $T1, T2, T3, T4$ sont les axiomes des groupes commutatifs. Les formules $T5, T6, T7$ exprime la compatibilité de la positivité et de l'addition. La formule $T8$ souligne le caractère discret de l'ordre. En raison des axiomes de groupe commutatif on peut démontrer pour tout terme t une égalité $t = a_1x_1 + \ldots + a_nx_n + b$ où $x_1, \ldots, x_n$ sont des variables distinctes et $a_1, \ldots, a_n, b \in \mathbb{Z}$.

Exemples. D'après $T1$, $T2$ et $T3$, $-0 = 0$. Par conséquent d'après $T6$, $\neg 0 > 0$. Montrons que $T \models kx = ky \Rightarrow x = y$. Si $y - x > 0$ alors l'emploi répété de $T7$ entraine que $k(y - x) > 0$. De même si $x - y > 0$ alors $k(y - x) > 0$. D'après $T5$, $T \models x \neq y \Rightarrow y - x > 0 \vee x - y > 0$. En appliquant le raisonnement précédent, $T \models x \neq y \Rightarrow k(y - x) > 0 \vee k(x - y) > 0$. Puisque $\neg 0 > 0$, $T \models x \neq y \Rightarrow k(x - y) \neq 0$. On conclut par contraposition.

Cependant cette théorie ne permet pas l'élimination des quantificateurs.

Théorème 6 *La théorie T des groupes commutatifs ordonnés discrets ne permet pas l'élimination des quantificateurs.*

Preuve

Raisonnons par l'absurde. Supposons que T permette l'élimination des quantificateurs. Soit ι le modèle « naturel » de T pour lequel $\mathbb{Z}$ est l'ensemble de base. D'après le théorème 3, $T \cup T^\iota$ est syntaxiquement complète.

Soit maintenant l'interprétation ι' dont l'ensemble de base est le groupe additif (composante par composante) $\{(x, y) \mid x \in \mathbb{Z} \wedge y \in \mathbb{Z}\}$ et dont la relation d'ordre est définie par $(x, y) > 0$ ssi $x > 0$ ou $x = 0 \wedge y > 0$. ι' est un modèle de $T \cup T^\iota$ avec l'interprétation $x^{\iota'} = (0, x)$.

Soit maintenant la formule close $\varphi \equiv \forall x\ \exists y\ x = 2y \vee x + 1 = 2y$ on a $\iota \models \varphi$ (tout nombre est pair ou impair) et $\iota \models \neg\varphi$ (prendre pour x, la paire $(1, 1)$). Par conséquent ni $T \cup T^\iota \models \varphi$ ni $T \cup T^\iota \models \neg\varphi$ contrairement à l'hypothèse.

c.q.f.d. $\Diamond\Diamond\Diamond$

Afin de remédier à ce problème, nous allons élargir le support et la théorie (cette méthode n'est pas toujours possible) afin d'éliminer les quantificateurs tandis que $\mathbb{Z}$ restera un modèle de cette théorie. Nous ajoutons un ensemble dénombrable de prédicats unaires $n|x$ pour $n > 1$ et les formules suivantes :

(T9) $\forall x\ (n|x \Leftrightarrow \exists y x = ny)$

(T10) $\forall x\ n|x \vee n|x+1 \vee \ldots \vee n|x+n-1$

Exemple. On peut démontrer que pour tout $0 < i < n$, on a $\neg n|i$. En effet si $i = yn$ pour un certain y alors $y > 0$ et $y \neq 1$ puis (en utilisant $T8$), $(y-1)n > 0$ et par conséquent $yn - i = (y-1)n + (n-i) > 0$. D'où une contradiction.

Théorème 7 *La théorie T des groupes commutatifs ordonnés discrets avec divisibilité permet l'élimination des quantificateurs.*

Preuve

Nous réduisons d'abord les formules atomiques possibles. $t_1 = t_2$ équivaut à $t_1 - t_2 = 0$ et $t \neq 0$ équivaut à $t > 0 \vee -t > 0$. De même $\neg t > 0$ équivaut à $t = 0 \vee -t > 0$. Enfin $\neg n|t$ équivaut à $n|t+1 \vee \ldots \vee n|t+n-1$ (en se servant du résultat de l'exemple). Par conséquent, les formules atomiques que nous considérons sont de la forme $t = 0$, $t > 0$ et $n|t$. Par souci de lisibilité, nous noterons parfois $t_1 - t_2 > 0$, $t_1 > t_2$.

Soit donc une formule
$\varphi \equiv \exists x\ \bigwedge_{i \leq k} p_i x > t_i \wedge \bigwedge_{i \leq l} q_i x = u_i \wedge \bigwedge_{i \leq m} n_i | r_i x - v_i$
où les $p_i, q_i, r_i \in \mathbb{Z}$ et les t_i, u_i, v_i sont des termes ne contenant pas x.

On procède à une simplification supplémentaire : $n_i|r_i x - v_i$ équivaut à $(n_i|r_i x \wedge n_i|v_i) \vee \ldots \vee (n_i|r_i x + n_i - 1 \wedge n_i|v_i + n_i - 1)$. Par conséquent dans la formule ci-dessus, on considère que les v_i sont des entiers.

On va raisonner par induction sur un ordre bien fondé défini par $\varphi' < \varphi$ ssi
- Soit $k > k'$
- Soit $k = k' \wedge \sum |p_i| > \sum |p'_i|$
- Soit $k = k' \wedge \sum |p_i| = \sum |p'_i| \wedge \sum |q_i| > \sum |q'_i|$
- Soit $k = k' \wedge \sum |p_i| = \sum |p'_i| \wedge \sum |q_i| = \sum |q'_i| \wedge \sum |n_i r_i| > \sum |n'_i r'_i|$

Supposons l'élimination des quantificateurs démontrée pour les formules ψ t.q. $\psi < \varphi$.

Supposons qu'il existe $p_i > 0$ et $p_{i'} > 0$ $(i \neq i')$. Alors $p_i x > t_i \wedge p_{i'} x > t_{i'}$ équivaut à $(p_i x > t_i \wedge p_{i'} t_i > p_i t_{i'}) \vee (p_{i'} x > t_{i'} \wedge \neg p_{i'} t_i > p_i t_{i'})$. Un raisonnement similaire s'applique dans le cas où il existe $p_i < 0$ et $p_{i'} < 0$ $(i \neq i')$. On peut donc supposer qu'il existe au plus un $p_i > 0$ et au plus un $p_i < 0$. Nous traitons le cas où chacun de ces p_i existe et laissons les autres cas en exercice.
$\varphi \equiv \exists x\ p_1 x > t_1 \wedge p_2 x > t_2 \wedge \bigwedge_{i \leq l} q_i x = u_i \wedge \bigwedge_{i \leq m} n_i | r_i x - v_i$
avec $p_1 > 0$ et $p_2 < 0$
Cette formule est équivalente à
$\exists x\ p_1 p_2 x < p_2 t_1 \wedge p_1 p_2 x > p_1 t_2 \wedge \bigwedge_{i \leq l} p_1 p_2 q_i x = p_1 p_2 u_i \wedge \bigwedge_{i \leq m} -p_1 p_2 n_i | p_1 p_2 r_i x - p_1 p_2 v_i$
Par « changement de variable », φ est aussi équivalente à
$\exists x\ x < p_2 t_1 \wedge x > p_1 t_2 \wedge \bigwedge_{i \leq l} q_i x = p_1 p_2 u_i \wedge \bigwedge_{i \leq m} -p_1 p_2 n_i | r_i x - p_1 p_2 v_i \wedge -p_1 p_2 | x$
Supposons qu'il existe q_i et $q_{i'}$ $(i \neq i')$ avec $|q_i| \leq |q_{i'}|$. Alors $q_i x = u_i \wedge q_{i'} x = u_{i'}$ équivaut à $q_i x = u_i \wedge (q_{i'} - q_i) x = u_{i'} - u_i$. On peut donc maintenant supposer que $l \leq 1$. Si $l = 1$ on peut supposer que $q_1 > 0$ (en multipliant par -1 l'égalité si nécessaire). L'élimination des quantificateurs est immédiate car φ équivaut à (en reprenant les notations initiales) :
$u_1 < q_1 t_1 \wedge u_1 > q_1 t_2 \wedge \bigwedge_{i \leq m} q_1 n_i | r_i u_1 - q_1 v_i$
On peut donc supposer que $l = 0$.

Soit n_i t.q. $n_i = nn'$ avec n et n' premiers entre eux alors $n_i|r_ix - v_i$ est équivalent à $n|r_ix - v_i \wedge n'|r_ix - v_i$. Par conséquent, on peut supposer que les n_i sont des puissances de nombre premier. Soit maintenant $\{a_1, \ldots, a_\alpha\}$ l'ensemble des entiers de l'intervalle $[0, n_i - 1]$ t.q. $n_i|r_ia_j - v_i$, alors $n_i|r_ix - v_i$ est équivalent à $n_i|x - a_1 \vee \ldots \vee n_i|x - a_\alpha$. Supposons que que $n_i = p^\alpha$ et $n_{i'} = p^{\alpha'}$ avec $\alpha \leq \alpha'$ alors $n_i|x - v_i \wedge n_{i'}|x - v_{i'}$ est équivalent à $n_i|v_i - v_{i'} \wedge n_{i'}|x - v_{i'}$ On est donc ramené au cas où les r_i sont tous égaux à 1 et les n_i sont tous premiers entre eux. On obtient donc une formule du type suivant :
$\varphi \equiv \exists x\ x < t_1 \wedge x > t_2 \wedge n_1|x - v_1 \wedge \ldots \wedge n_m|x - v_m$
avec $n_1, \ldots, n_m$ premiers entre eux.

L'élimination des quantificateurs est immédiate. En effet la formule est équivalente à :
$\bigvee_{\alpha \in [1..n_1 \times \ldots \times n_m]} t_2 + \alpha < t_1 \wedge n_1|t_2 + \alpha - v_1 \wedge \ldots \wedge n_m|t_2 + \alpha - v_m$

c.q.f.d. ◊◊◊

Puisque les transformations sont effectives, nous avons établi le corollaire suivant.

Corollaire 7 *Le problème de savoir si φ, une formule, est une conséquence de la théorie T des groupes commutatifs ordonnés discrets avec divisibilité est décidable.*

Preuve
On peut considérer que φ est une formule close en raison du lemme 8. La transformation ne crée pas de variables donc la formule équivalente est nécessairement une combinaison linéaire d'atomes $t = 0$, $t > 0$ ou $n|t$ avec t un terme sans variable (i.e. un élément de $\mathbb{Z}$). Chacun de ces termes a une valeur connue. d'où le résultat.

c.q.f.d. ◊◊◊

Remarques. Etant donné un modèle fixé de la théorie (par exemple $\mathbb{Z}$), l'évaluation d'une formule est décidable si les prédicats et les fonctions sont calculables sur ce modèle (ici l'addition et le caractère positif).

4.5 Corps algébriquement clos

On considère le support suivant : deux constantes 0 et 1, une fonction unaire $-$, deux fonctions binaires $+, \times$ et l'égalité. Le terme $1 + \ldots + 1$ (resp. $t + \ldots + t$), n fois, est noté n (resp. nt). Le terme $t + (-t')$ est noté $t - t'$. Le terme $t \times t'$ est noté tt'. La théorie T est définie par d'une part les axiomes d'un corps commutatif :

(T1) $\forall x\ x + 0 = x$

(T2) $\forall x\ x - x = 0$

(T3) $\forall x\ \forall y\ x + y = y + x$

(T4) $\forall x\ \forall y\ \forall z\ (x + y) + z = x + (y + z)$

(T5) $\forall x\ \forall y\ \forall z\ (xy)z = x(yz)$

(T6) $\forall x\ \forall y\ xy = yx$

(T7) $\forall x\ 1x = x$

(T8) $\forall x\ x = 0 \vee (\exists y\ xy = 1)$

(T9) $\forall x\ \forall y\ \forall z\ x(y+z) = xy + xz$

Les formules $T1, T2, T3, T4$ sont les axiomes des groupes commutatifs. Les formules $T5, T6, T7, T8, T9$ sont relatives à la loi de la multiplication dans un corps commutatif. En raison des axiomes de corps commutatif on peut démontrer pour tout terme t une égalité $t = pol(x_1, \ldots, x_n)$ où pol est un polynôme à coefficients dans $\mathbb{Z}$ et dont les variables sont $x_1, \ldots, x_n$.

Nous introduisons maintenant la propriété de clôture algébrique qui s'exprime par un ensemble dénombrable de formules, une par degré de polynomes ($n \geq 1$).

$$\texttt{(T10)} \quad \forall x_0 \ldots \forall x_{n-1}\ \exists x\ x_0 + x_1 x + \ldots + x_{n-1}x^{n-1} + x^n = 0$$

Les atomes de ce support (moyennant une éventuelle soustraction) s'écrivent $t = 0$ et $t \neq 0$ où t un polynôme multivariable à coefficients dans $\mathbb{Z}$. Avant d'entreprendre l'élimination des quantificateurs, nous allons effectuer quelques développements mathématiques.

4.5.1 Division euclidienne et PGCD

Notations et rappels. Soit $P \equiv a_p X^p + \ldots + a_0$ (avec $a_p \neq 0$) un polynôme à coefficients dans un anneau intègre $\mathbb{D}$ (dont $\mathbb{K}$ son corps des fractions est lui-même inclus dans $\mathbb{C}$ un corps algébriquement clos).

- Le degré de P noté $deg(P)$ est p. Par convention le degré du polynôme nul est $-\infty$.
- Soient $P, Q \in \mathbb{D}[X]$ deux polynômes, Q est un diviseur de P s'il existe $A \in \mathbb{K}[X]$ t.q. $P = AQ$.
- Si $Q \neq 0$ alors le reste de la division euclienne de P par Q est l'unique polynôme $R \in \mathbb{K}[X]$ tel que $deg(R) < deg(Q)$ et qu'il existe $A \in \mathbb{K}[X]$ avec $P = AQ + R$. R le reste est noté $Rem(P, Q)$ et A le quotient est noté $Quo(P, Q)$.
- Un pgcd $G \in \mathbb{K}[X]$ de P, Q est un diviseur de P et Q t.q. tout diviseur de P et Q divise G. Un pgcd est unique à un facteur multiplicatif de $\mathbb{K}^*$ près. On notera un quelconque de ces pgcd $pgcd(P, Q)$. Un ppcm $G \in \mathbb{K}[X]$ de P, Q est un multiple de P et Q t.q. tout multiple de P et Q est un multiple de G. Un ppcm est unique à un facteur multiplicatif de $\mathbb{K}^*$ près. On notera un quelconque de ces ppcm $ppcm(P, Q)$. Si P ou Q est différent de 0 alors $\frac{PQ}{pgcd(P,Q)}$ est un ppcm de P et Q.

Définition 37 *La définition d'un pgcd se généralise à une famille finie de polynômes $\mathcal{P}$ inductivement par :*

- $pgcd(\emptyset) \equiv 0$
- $pgcd(\{P\}) \equiv P$
- $pgcd(\mathcal{P} \cup \{P\}) \equiv pgcd(P, pgcd(\mathcal{P}))$

Cette opération est bien définie car l'opération binaire « pgcd » est associative et commutative.

Autres rappels. Un corps algébriquement clos est infini. En effet si $\mathbb{C} = \{x_1, \ldots, x_n\}$ alors $P \equiv (\prod_{i \leq n}(x - x_i)) + 1$ n'admet pas de racines. Un polynôme P de degré $n \leq 0$, admet au plus n racines. Plus précisément, par clôture algébrique et application répétée de la division euclienne, $P = c(\prod_{i \leq n}(x - a_i))$ où $c \neq 0$ et où certains a_i peuvent être égaux.

Nous rappelons maintenant l'algorithme d'Euclide dont nous utiliserons les résultats dans cette section et la suivante. Soient $P, Q \in \mathbb{K}[X]$. Alors la *séquence des restes signés* notée $SRemS(P, Q)$ est une suite de polynômes $SRemS_0(P, Q), \ldots, SRemS_k(P, Q)$ définie itérativement par :

- $SremS_0(P, Q) = P, SremS_1(P, Q) = Q$
- $\forall 2 \leq i \leq k$ $SremS_i(P, Q) = -Rem(SremS_{i-2}(P, Q), SremS_{i-1}(P, Q))$ avec la condition $SremS_i(P, Q) \neq 0$
- $SremS_{k+1}(P, Q) = -Rem(SremS_{k-1}(P, Q), SremS_k(P, Q))$ avec la condition $SremS_{k+1}(P, Q) = 0$

Le signe $-$ introduit dans la séquence est sans importance dans le cas des corps algébriquement clos mais il deviendra significatif à la prochaine section. Bien entendu, $SRemS_k(P, Q)$ est un pgcd de P et Q.

Afin de bénéficier de l'algorithme d'Euclide dans le cadre de $\mathbb{D}[X]$, il nous faut introduire un pseudo-reste de la division de $P \equiv \sum_{i \leq p} a_i X^i$ par $Q \equiv \sum_{j \leq q} b_j X^j$. Ce pseudo-reste, noté $PRem(P, Q)$, est défini par $PRem(P, Q) \equiv Rem(b_q^d P, Q)$ où d est le plus petit entier pair supérieur ou égal à $p - q + 1$ (la parité ne sert qu'à la prochaine section). L'intérêt de cette définition provient du fait que $PRem(P, Q) \in \mathbb{D}[X]$. La *séquence des restes signés* notée $SPRemS(P, Q)$ est une suite de polynômes $SPRemS_0(P, Q), \ldots, SPRemS_k(P, Q)$ définie itérativement par :

- $SPRemS_0(P, Q) = P, SPRemS_1(P, Q) = Q$
- $\forall 2 \leq i \leq k$, $SPRemS_i(P, Q) = -PRem(SPRemS_{i-2}(P, Q), SPRemS_{i-1}(P, Q))$ avec la condition $SPRemS_i(P, Q) \neq 0$
- $SPRemS_{k+1}(P, Q) = -Rem(SPRemS_{k-1}(P, Q), SPRemS_k(P, Q))$ avec la condition $SPRemS_{k+1}(P, Q) = 0$

Illustrons cette définition avec $\mathbb{D} = \mathbb{Z}[a, b, c]$. Soit $P \equiv X^4 + aX^2 + bX + c$ et $Q \equiv 4X^3 + 2aX + b$ ($Q = P'$). Alors :
$SPREMS_2(P, Q) = -Rem(4^2 P, Q) = -8aX - 12bX - 16c$
$SPREMS_3(P, Q) = -Rem(64a^2 Q, SPREMS_2(P, Q))$
$= 64((8ac - 9b^2 - 2a^3)X - b(12c + a^2))$

$SPREMS_4(P, Q)$
$= -Rem(4096(8ac - 9b^2 - 2a^3)^2 SPREMS_2(P, Q), SPREMS_3(P, Q))$
$= 16384a^2(256c^3 - 128a^2c^2 + 144ab^2c + 16a^4c - 27b^4 - 4a^3b^2)$

$SPREMS_5(P, Q) = 0$

Poursuivons l'étude de cet exemple. Si on remplace les variables a, b, c par des valeurs de $\mathbb{C}$, on peut s'interroger si la séquence des restes signés est la même (à un facteur multiplicatif non nul près). Ceci ne dépend que du coefficient de plus haut degré de chaque polynôme de la suite. Notons :

$S_i \equiv SPRemS_i(P,Q)$, $s = 8ac - 9b^2 - 2a^3$
et $\delta = 256c^3 - 128a^2c^2 + 144ab^2c + 16a^4c - 27b^4 - 4a^3b^2$

Nous pouvons affirmer que pour tout triplet $(\overline{a}, \overline{b}, \overline{c}) \in C^3$ tel que :
$\overline{a} \neq 0 \wedge s(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) \neq 0 \wedge \delta(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) \neq 0$
le pgcd de $P(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\})$ et $Q(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\})$ est
$S_4(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\})$.

Qu'en est-il pour les autres triplets ? Examinons d'abord le cas $\overline{a} = 0$. Il y a en fait différents cas selon le degré possible de S_2 :

- $deg(S_2)(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) = 1$ ssi $\overline{a} = 0 \wedge \overline{b} \neq 0$
- $deg(S_2)(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) = 0$ ssi $\overline{a} = 0 \wedge \overline{b} = 0 \wedge \overline{c} \neq 0$
- $deg(S_2)(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) = -\infty$ ssi $\overline{a} = 0 \wedge \overline{b} = 0 \wedge \overline{c} = 0$

Développons le cas où $deg(S_2)(\{a \leftarrow \overline{a}, b \leftarrow \overline{b}, c \leftarrow \overline{c}\}) = 1$. Définissons $Tru_1(S_2) \equiv -12bX - 16$ et appelons $u \equiv -PRem(Q, Tru_1(S_2))$. Un calcul élémentaire nous fournit :

$$u = 768b(-27b^4 + 72acb^2 + 256c^3)$$

Nous avons maintenant les éléments nécessaires pour étudier tous les cas possibles de $pgcd(P,Q)$ selon les valeurs du triplet $(\overline{a}, \overline{b}, \overline{c})$.

La figure 4.1 représente sous forme d'un arbre tous les cas possibles (avec $t = b(12c + a^2)$). Nous y avons formalisé la notion de troncature précédemment introduite.

Définition 38 *Soit $P \equiv \sum_{i \leq p} a_i X^i \in \mathbb{D}[X]$ un polynôme non nul. La troncature de P en $j \in \{-\infty, 0, \ldots, p\}$, notée $Tru_j(P)$ est définie par :*

$$Tru_j(P) \equiv \sum_{i \leq j} a_i X^i$$

L'ensemble $Tru(P)$ des troncatures de $P \in \mathbb{D}[X_1, \ldots, X_n][X]$ non nul est défini récursivement par :

- $Tru(P) \equiv \{P\}$ *si* $a_p \in \mathbb{D}$;
- $Tru(P) \equiv \{P, 0\}$ *si* $a_p \notin \mathbb{D}$ *et* $deg(P) = 0$;
- $Tru(P) \equiv \{P\} \cup Tru(Tru_{p-1}(P))$ *sinon.*

Définition 39 *L'arbre des séquences des pseudo-restes signés de deux polynômes $P, Q \in \mathbb{D}[X_1, \ldots, X_n][X]$, noté $TRems(P,Q)$, est un arbre dont la racine est P. Les fils de la racine sont étiquetés par les éléments de $Tru(Q)$. Chaque noeud nd est étiqueté par un polynôme $Pol(nd)$ de $\mathbb{D}[X_1, \ldots, X_n][X]$. Un noeud est une feuille ssi le polynôme associé est nul. Sinon l'ensemble des fils des nd est étiqueté par l'ensemble des troncatures de $-PRem(Pol(p(nd)), Pol(nd))$ où $p(nd)$ est le père de nd. Chaque noeud associé à un polynôme $Tru_i(R) \in Tru(R)$ où $Tru(R)$ n'est pas un singleton est aussi étiqueté par la formule (avec $R \equiv \sum_{i \leq r} a_i X^i$)*

- $a_r \neq 0$ *si* $i = r$
- $a_r = 0 \wedge \ldots \wedge a_{i+1} = 0 \wedge a_i \neq 0$ *si* $0 \leq i < r$
- $a_r = 0 \wedge \ldots \wedge a_0 = 0$ *si* $i = -\infty$

Soit *path* une branche de cet arbre, $pgcd(path)$ est le polynôme associé au père de la feuille de *path*. La formule φ_{path} est la conjonction des formules qui étiquettent les noeuds de *path*.

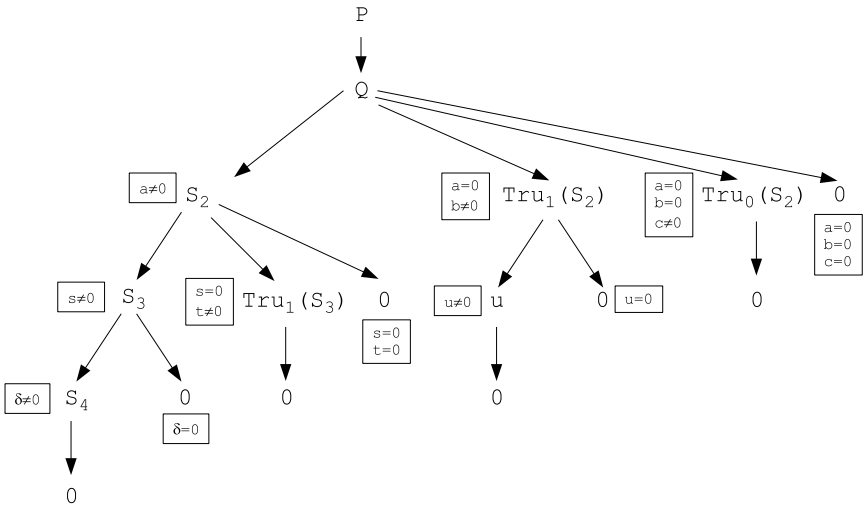

FIG. 4.1: L'arbre des pgcd possibles

L'intérêt de cet arbre est décrit dans la proposition suivante dont la preuve découle immédiatement de la construction de l'arbre.

Proposition 23 *Soit P, Q deux polynômes de $\mathbb{D}[X_1, \ldots, X_n][X]$.*

- *Pour tout tuple $(\overline{x}_1, \ldots, \overline{x}_n) \in \mathbb{C}^n$, il existe une et une seule formule φ_{path} où path est une branche de $TRems(P, Q)$ t.q. $\varphi_{path}(\{X_i \leftarrow \overline{x}_i\}) = \texttt{true}$.*
- *Soit path le chemin associé à $(\overline{x}_1, \ldots, \overline{x}_n)$, alors $pgcd(path)(\{X_i \leftarrow \overline{x}_i\})$ est un pgcd de $P(\{X_i \leftarrow \overline{x}_i\})$ et $Q(\{X_i \leftarrow \overline{x}_i\})$*
  et le degré de $pgcd(path)(\{X_i \leftarrow \overline{x}_i\})$ est le degré de $pgcd(path)$ en X.

4.5.2 Elimination des quantificateurs

Le lemme suivant est la clef de l'élimination des quantificateurs avec l'arbre des séquences des pseudo-restes signés.

Lemme 27 *Soient $\mathcal{P}$ et $\mathcal{Q}$, deux sous-ensembles finis de $\mathbb{D}[X]^*$ t.q. $\mathcal{P} \cup \mathcal{Q} \neq \emptyset$. Notons*

$$Sol(\mathcal{P}, \mathcal{Q}) \equiv \{x \in \mathbb{C} \mid \forall P \in \mathcal{P}, P(x) = 0 \wedge \forall Q \in \mathcal{Q}, Q(x) \neq 0\}$$

Alors

Cas $\mathcal{Q} = \emptyset$. $Sol(\mathcal{P}, \mathcal{Q}) \neq \emptyset$ *ssi* $deg(pgcd(\mathcal{P})) \neq 0$
Cas $\mathcal{P} = \emptyset$. $Sol(\mathcal{P}, \mathcal{Q}) \neq \emptyset$
Cas $\mathcal{P} \neq \emptyset \wedge \mathcal{Q} \neq \emptyset$. *Notons* $d = deg(pgcd(\mathcal{P}))$.
$Sol(\mathcal{P}, \mathcal{Q}) \neq \emptyset$ *ssi* $deg(pgcd(pgcd(\mathcal{P}), \prod_{Q \in \mathcal{Q}} Q^{d+1})) \neq d$

Preuve
Cas $\mathcal{Q} = \emptyset$. Soit $x \in \mathbb{C}$. Alors $\forall P \in \mathcal{P}, P(x) = 0$ ssi $pgcd(\mathcal{P})(x) = 0$. Donc

il existe un tel x ssi $deg(pgcd(\mathcal{P})) \neq 0$ en vertu de la propriété de clôture algébrique.

Cas $\mathcal{P} = \emptyset$**.** Soit $x \in \mathbb{C}$. Alors $\forall Q \in \mathcal{Q}, Q(x) \neq 0$ ssi $\prod_{Q\in\mathcal{Q}} Q(x) \neq 0$. Puisque $0 \notin \mathcal{Q}$, $\prod_{Q\in\mathcal{Q}} Q(x)$ n'est pas nul et admet donc un nombre fini de racines alors que $\mathbb{C}$ est infini.

Cas $\mathcal{P} \neq \emptyset \wedge \mathcal{Q} \neq \emptyset$**.** Soit $x \in \mathbb{C}$. $x \in Sol(\mathcal{P}, \mathcal{Q})$ ssi $pgcd(\mathcal{P})(x) = 0 \wedge \prod_{Q\in\mathcal{Q}} Q^{d+1}(x) \neq 0$ puisque $d \geq 0$. Soient $x_1, \ldots, x_k$ les racines de $pgcd(\mathcal{P})$ avec leur multiplicité $n_1, \ldots, n_k$ t.q. $n_1 + \ldots + n_k = d$. Si pour tout x_i, il existe un $Q \in \mathcal{Q}$ t.q. $Q(x) = 0$ alors $pgcd(P)$ divise $\prod_{Q\in\mathcal{Q}} Q^{d+1}$ (grace à la puissance $d+1$). S'il existe un x_i t.q. pour tout $Q \in \mathcal{Q}$, $Q(x) \neq 0$, alors $x - x_i$, et donc $pgcd(P)$, ne divise pas $\prod_{Q\in\mathcal{Q}} Q^{d+1}$.

c.q.f.d. ◊◊◊

Théorème 8 *La théorie T des corps algébriquement clos permet l'élimination des quantificateurs.*

Preuve
Soit une formule,
$\varphi \equiv \exists X \ \bigwedge_{P\in\mathcal{P}}, P = 0 \wedge \bigwedge_{Q\in\mathcal{Q}} Q \neq 0$ avec $P, Q \in \mathbb{Z}[X_1, \ldots, X_n][X]$.
En itérant la construction de $Trems$, on produit un ensemble de formules sans quantificateur $\{\varphi_i\}_{i\in I}$ mutuellement exclusives t.q. $\bigvee_{i\in I} \varphi_i = \texttt{true}$ et de polynômes $\{P_i\}_{i\in I}$ t.q. pour tout tuple $(\overline{x}_1, \ldots \overline{x}_n)$, $\varphi_i(\{X_i \leftarrow \overline{x}_i\})$ implique que $P_i(\{X_i \leftarrow \overline{x}_i\})$ est un $pgcd(\mathcal{P}(\{X_i \leftarrow \overline{x}_i\})$ et que le degré de $P_i(\{X_i \leftarrow \overline{x}_i\})$ est le degré de P_i en X.

Nous traitons ensuite chacun des trois cas du lemme précédent pour obtenir une disjonction de formules comportant uniquement les variables $X_1, \ldots, X_n$ équivalente à φ. Les deux premiers cas sont triviaux. Pour le troisième cas, on applique encore la construction de $TRems$ avec chacun des P_i et $\prod_{Q\in\mathcal{Q}} Q^{d+1}$ où d est le degré de P_i ce qui nous conduit à des polynômes $P'_{i,j}$ et des formules $\varphi'_{i,j}$. Les disjonctions de ce cas sont $\varphi_i \wedge \varphi_{i,j}$ pour tout (i, j) t.q. $deg(P'_{i,j}) \neq d$.

c.q.f.d. ◊◊◊

A titre d'exemple, nous donnons une application de ce résultat.

Proposition 24 *Soient ι un modèle de corps algébriquement clos et φ une fomule de cette logique comportant une unique variable libre x. Alors soit l'ensemble $E^\iota_\varphi \equiv \{a \in E_\iota \mid \varphi^\iota(x \leftarrow a) = \texttt{true}\}$ soit son complémentaire est fini.*

Preuve
Sans perte de généralité, nous considérons une formule φ sans quantificateur. S'il s'agit d'un atome $P = 0$, E^ι_φ est l'ensemble fini des racines.

On démontre la propriété par induction. Si $\varphi \equiv \neg\varphi'$ la propriété est trivialement conservée. Si $\varphi \equiv \varphi' \wedge \varphi''$, soit $E^\iota_{\varphi'}$ soit $E^\iota_{\varphi''}$ est fini et leur intersection est aussi finie. Dans le cas contraire le complémentaire, union de deux ensembles finis, est fini.

c.q.f.d. ◊◊◊

Corollaire 8 *Le problème de savoir si φ, une formule est une conséquence de la théorie T des corps algébriquement clos est décidable.*

Preuve
On peut considérer que φ est une formule close en raison du lemme 8. La transformation ne crée pas de variables donc la formule équivalente est nécessairement une combinaison booléenne d'atomes $t = 0$ avec t un terme sans variable (i.e. un élément de $\mathbb{Z}$). Sa valeur de vérité dépend de la caractéristique du corps. Aussi on évalue la formule pour toutes les caractéristiques (i.e. 0 et les nombres premiers) inférieures à la plus grande valeur absolue des entiers apparaissant dans la formule. La formule est une conséquence de la théorie ssi la formule est vraie pour toutes ces caractéristiques (les autres caractéristiques sont équivalentes à 0 pour cette formule).

c.q.f.d. $\Diamond\Diamond\Diamond$

Remarque. L'élimination des quantificateurs n'introduit ni variable ni constante. Par conséquent, toute formule close **sans constante** a la même valeur de vérité quelque soit le corps algébriquement clos. Il n'en est pas de même pour les formules closes avec constantes. Ainsi $p = 0$ avec p premier, sera vraie pour les corps de caractéristique p et fausse pour les autres corps.

4.6 Corps réel fermé

On considère le support suivant : deux constantes 0 et 1, une fonction unaire $-$, deux fonctions binaires $+, \times$, l'égalité et la relation unaire de positivité >0. Le terme $1 + \ldots + 1$ (resp. $t + \ldots + t$), n fois, est noté n (resp. nt). Le terme $t + (-t')$ est noté $t - t'$. Le terme $t \times t'$ est noté tt'. La théorie T est définie par d'une part les axiomes d'un corps commutatif :

(T1) $\forall x\ x + 0 = x$

(T2) $\forall x\ x - x = 0$

(T3) $\forall x\ \forall y\ x + y = y + x$

(T4) $\forall x\ \forall y\ \forall z\ (x + y) + z = x + (y + z)$

(T5) $\forall x\ \forall y\ \forall z\ (xy)z = x(yz)$

(T6) $\forall x\ \forall y\ xy = yx$

(T7) $\forall x\ 1x = x$

(T8) $\forall x\ x = 0 \vee (\exists y\ xy = 1)$

(T9) $\forall x\ \forall y\ \forall z\ x(y + z) = xy + xz$

D'autre part, les axiomes d'un corps ordonné.

(T10) $\forall x\ x = 0 \vee x > 0 \vee -x > 0$

(T11) $\forall x\ \neg(x > 0 \wedge -x > 0)$

(T12) $\forall x\ \forall y\ (x > 0 \wedge y > 0) \Rightarrow x + y > 0$

(T13) $\forall x\ \forall y\ (x > 0 \wedge y > 0) \Rightarrow xy > 0$

Enfin les axiomes des corps réels fermés qui affirment que tout nombre positif est un carré et que tout polynôme de degré impair a une racine (en fait une famille de théorèmes par degré impair).

(T14) $\forall x\ \exists y\ x = y^2 \vee -x = y^2$

(T15) $\forall x_0 \ldots \forall x_{2n}\ \exists x\ x_0 + x_1 x + \ldots + x_{2n} x^{2n} + x^{2n+1} = 0$

4.6.1 Rappels algébriques

Afin d'obtenir l'élimination des quantificateurs, nous allons effectuer quelques développements algébriques plus conséquents que ceux de la section précédente. Nous supposons connus les résultats élémentaires sur les anneaux, les idéaux et les corps. Entre autres, un idéal $\mathbb{I}$ d'un anneau $\mathbb{A}$ est un idéal maximal ssi l'anneau quotient $\mathbb{A}/\mathbb{I}$ est un corps. On rappelle aussi le lemme de Zorn (de la théorie des ensembles) qui s'énonce ainsi :

Soit un ensemble partiellement ordonné non vide dont toute partie totalement ordonnée admet un majorant alors cet ensemble admet un élément maximal.

Lemme 28 (Krull) *Soit $\mathbb{A}$ un anneau et $\mathbb{I}$ un idéal strictement inclus dans $\mathbb{A}$. Alors il existe un idéal maximal $\mathbb{I}'$ qui contient $\mathbb{I}$.*

Preuve
Soit $\mathcal{I}$, l'ensemble (non vide) des idéaux propres de $\mathbb{A}$ contenant $\mathbb{I}$ ordonné par l'inclusion. Soit $\{\mathbb{I}_f\}_{f\in F}$ une partie totalement ordonnée de $\mathcal{I}$. On vérifie facilement que $\bigcup_{f\in F} \mathbb{I}_f$ est un idéal. Les conditions du lemme de Zorn sont vérifiées. Soit $\mathbb{I}'$ un élément maximal alors $\mathbb{I}'$ est un idéal maximal de $\mathbb{A}$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Une extension algébrique $\mathbb{K}'$ d'un corps $\mathbb{K}$ est un corps t.q. $\mathbb{K} \subseteq \mathbb{K}'$ et tout élément de $\mathbb{K}'$ annule un polynôme à coefficients dans $\mathbb{K}$. Une clôture algébrique d'un corps $\mathbb{K}$ est une extension algébrique de $\mathbb{K}$ qui est algébriquement close. Nous laissons en exercice le fait que si $\mathbb{K}''$ est une extension algébrique de $\mathbb{K}'$ et $\mathbb{K}'$ est une extension algébrique de $\mathbb{K}$ alors $\mathbb{K}''$ est une extension algébrique de $\mathbb{K}$.

Lemme 29 *Soit $\mathbb{K}$ un corps et soit $\mathcal{P}$ une famille finie de polynômes non constants de $\mathbb{K}[X]$. Alors il existe $\mathbb{K}'$, une extension algébrique de $\mathbb{K}$ t.q. tout polynôme de $\mathcal{P}$ admet une racine dans $\mathbb{K}'$.*

Preuve
La preuve se fait par récurrence sur le nombre de polynômes de $\mathcal{P}$. Soit $P \in \mathcal{P}$, Si P admet une racine dans $\mathbb{K}$, il n'y a rien à faire. Sinon soit Q, un facteur irréductible de P alors $\mathbb{I}(Q)$ l'idéal engendré par P est maximal et par conséquent $\mathbb{K}[X]/\mathbb{I}(Q)$ est un corps qui est une extension algébrique de $\mathbb{K}$ engendré par les images de $1, X, \ldots, X^{deg(Q)-1}$ et t.q. l'image de X est une racine de Q donc de P.

c.q.f.d. $\Diamond\Diamond\Diamond$

Lemme 30 *Soit $\mathbb{K}$ un corps et soit $\mathcal{P}$ une famille quelconque de polynômes non constants de $\mathbb{K}[X]$. Alors il existe $\mathbb{K}'$, une extension algébrique de $\mathbb{K}$ tq. tout polynôme de $\mathcal{P}$ admet une racine dans $\mathbb{K}'$.*

Preuve
Soit $\mathbb{A} \equiv \mathbb{K}[\{X_P\}_{P\in\mathcal{P}}]$ la $\mathbb{K}$-algèbre de polynômes sur les variables $\{X_P\}_{P\in\mathcal{P}}$. Soit $\mathbb{I}$ l'idéal engendré par $\{P(X \leftarrow X_P)\}_{P\in\mathcal{P}}$. Montrons que I est un idéal propre de $\mathbb{A}$. Si ce n'est pas le cas, il existe une égalité $\sum_{i\leq n} Q_i P_i(X \leftarrow X_{P_i}) = 1$. Soit $\mathbb{K}'$ l'extension algébrique de $\mathbb{K}$ dans laquelle chaque P_i admet une racine a_i. L'égalité précédente est aussi vraie dans $\mathbb{K}'[\{X_P\}_{P\in\mathcal{P}}]$ mais dans ce cas le polynôme $(\sum_{i\leq n} Q_i P_i(X \leftarrow X_{P_i}))(\{X_{P_i} \leftarrow a_i\})$ est égal à la fois à 0 et à 1.
D'après le lemme de Krull, $\mathbb{I}$ est inclus dans un idéal maximal $\mathbb{I}_m$. Soit le corps $\mathbb{K}' \equiv \mathbb{A}/\mathbb{I}_m$, pour tout $P \in \mathcal{P}$, $P(X \leftarrow X_P) \in \mathbb{I} \subseteq \mathbb{I}_m$. En notant $\overline{X}_P$, l'image de X_P dans $\mathbb{K}'$, on a $P(\overline{X}_P) = 0$. Puisque les $\overline{X}_P$ sont algébriques sur $\mathbb{K}$ et qu'ils engendrent $\mathbb{K}'$, $\mathbb{K}'$ est une extension algébrique de $\mathbb{K}$.

c.q.f.d. ◊◊◊

Théorème 9 (Steinitz) *Tout corps $\mathbb{K}$ admet une clôture algébrique.*

Preuve
Nous construisons $\{\mathbb{K}_i\}_{i\in\mathbb{N}}$ une suite de corps croissante pour l'inclusion. $\mathbb{K}_0 \equiv \mathbb{K}$ Supposons $\mathbb{K}_i$ construit, alors $\mathbb{K}_{i+1}$ est l'extension algébrique du lemme précédent dans laquelle tous les polynômes de $\mathbb{K}_i[X]$ admettent une racine. Par récurrence, $\mathbb{K}_i$ est une extension algébrique de $\mathbb{K}$.
Soit le corps $\mathbb{K}' = \bigcup_{i\in\mathbb{N}} \mathbb{K}_i$. C'est une extension algébrique de $\mathbb{K}$ puisque chaque $\mathbb{K}_i$ est une extension algébrique de $\mathbb{K}$. Soit un polynôme de $P \in \mathbb{K}'[X]$, il existe un $\mathbb{K}_i$ t.q. $P \in \mathbb{K}_i[X]$. Par conséquent P admet une racine dans $\mathbb{K}_{i+1} \subseteq \mathbb{K}'$.

c.q.f.d. ◊◊◊

La clôture algébrique est unique à un $\mathbb{K}$-isomorphisme près mais nous n'utiliserons pas ce résultat.

Nous introduisons maintenant les polynômes symétriques essentiels pour établir un résultat fondamental des corps réels fermés.

Définition 40 *Soit $P \in \mathbb{D}[X_1, \ldots, X_k]$ un polynôme à coefficients dans $\mathbb{D}$ un anneau intègre. P est dit symétrique si pour toute permutation $\sigma \in \mathcal{S}_k$ (les permutations de $\{1, \ldots, k\}$),*

$$P[\{X_i \leftarrow X_{\sigma(i)}\}] = P$$

Le polynôme symétrique E_i est défini par

$$E_i \equiv \sum_{1\leq j_1<\ldots<j_i\leq k} X_{j_1} \ldots X_{j_i}$$

Rappelons le lemme classique qui s'obtient par un simple développement.

Lemme 31

$$\prod_{i\leq k}(X - X_i) = X^k + \sum_{1\leq i\leq k} (-1)^i E_i X^i$$

On dote $\mathbb{N}^k$ de l'ordre (total) lexicographique gradué par $(\alpha_1, \ldots, \alpha_k) <_{grlex} (\beta_1, \ldots, \beta_k)$ ssi $\sum_{i\leq k} \alpha_i < \sum_{i\leq k} \beta_i$ ou $(\sum_{i\leq k} \alpha_i = \sum_{i\leq k} \beta_i$ et $(\alpha_1, \ldots, \alpha_k) <_{lex} (\beta_1, \ldots, \beta_k))$ avec $<_{lex}$ étant l'ordre lexicographique standard. Cet ordre s'applique immédiatement aux monômes $X_1^{\alpha_1} \ldots X_k^{\alpha_k}$ de $\mathbb{K}[X_1, \ldots, X_k]$. Remarquons que l'ordre lexicographique gradué est bien fondé.

Proposition 25 *Soit $P \in \mathbb{D}[X_1, \ldots, X_k]$ un polynôme symétrique alors il existe un polynôme $R \in \mathbb{D}[X_1, \ldots, X_k]$ t.q. $P = R[\{X_i \leftarrow E_i\}_{i \leq k}]$.*

Proposition 26 *Soit $P \in \mathbb{D}[X]$ un polynôme dont les racines dans un corps algébriquement clos contenant $\mathbb{D}$ sont $x_1, \ldots, x_k$. Soit un polynôme symétrique $Q \in \mathbb{D}[X_1, \ldots, X_k]$ alors $Q[\{X_i \leftarrow x_i\}_{i \leq k}] \in \mathbb{D}$.*

Lemme 32 *Soient $\mathbb{K} \subset \mathbb{K}'$ deux corps de caractéristique nulle tels que tout polynôme séparable de $\mathbb{K}[X]$ admet une racine dans $\mathbb{K}'$, alors tout polynôme de $\mathbb{K}[X]$ admet une racine dans $\mathbb{K}'$*

Théorème 10 *Soit $\mathbb{R}$ un corps réel fermé, alors :*
- *$\mathbb{R}[i] \equiv \mathbb{R}[X]/(X^2+1)$ est une clôture algébrique de $\mathbb{R}$.*
- *Soit $P \in \mathbb{R}[X]$ t.q. $P[X \leftarrow a]P[X \leftarrow b] < 0$ alors $\exists c \in \mathbb{R}\ a < c < b \wedge P[X \leftarrow c] = 0$.*

Remarque. Nous avons au passage obtenu une démonstration purement algébrique que le corps des complexes est algébriquement clos (moyennant le fait que le corps des réels est complet).

Nous démontrons maintenant un résultat (vrai dans les réels) qui nous autorise à parler du signe d'un polynôme à l'infini.

Lemme 33 *Soit $P \equiv \sum_{i \leq p} a_i X^i$ avec $a_p \neq 0$, un polynôme d'un corps réel fermé. Si $|x| > 2 \sum_{i \leq p} \frac{a_i}{a_p}$ alors $P(x)$ et $a_p x^p$ ont même signe.*

Lemme 34 *Soit $P \in \mathbb{R}[X]$ et $a < b \in \mathbb{R}$ t.q. $P[X \leftarrow a] = P[X \leftarrow b] = 0$. Alors il existe $a < c < b$ t.q. $P'[X \leftarrow c] = 0$.*

4.6.2 Comptage de racines

Le point clef de l'élimination des quantificateurs dans un corps réel fermé est la possibilité de compter les racines d'un polynôme P dans un intervalle. Par exemple, une formule t.q.

$$\varphi \equiv \exists x_1\ \exists x_2\ x_1 \neq x_2 \wedge P[X \leftarrow x_1] = 0 \wedge P[X \leftarrow x_2] = 0 \wedge \forall x_3 (P[X \leftarrow x_3] = 0 \Rightarrow (x_3 = x_1 \vee x_3 = x_2))$$

est vraie ssi le polynôme P a exactement deux racines.

En fait nous avons besoin d'une notion plus forte qui est liée au signe d'un polynôme Q lorsqu'il est évalué pour les racines de P. Puisque le nombre de racines d'un polynôme est fini, on peut définir (grâce au théorème des valeurs intermédiaires) le signe d'un polynôme à droite et à gauche d'une valeur quelconque. On parle aussi du signe d'un polynôme en ∞ ou $-\infty$ déterminé par le monôme de plus haut degré du polynôme.

Définition 41 (Indice de Cauchy) *Soit $P, Q \in \mathbb{R}[X]$ deux polynômes, soit x une racine de P. On dit que la fraction Q/P* saute *de $-\infty$ (resp. ∞) à ∞ (resp. $-\infty$) en x si la multiplicité μ de x comme racine de P est supérieure à la multiplicité de ν de x comme racine de P, $\mu - \nu$ est impair et le signe de Q/P à droite de x est positif (resp. négatif).*

Soit $]a, b[$ un intervalle non vide (et éventuellement non borné), l'indice de Cauchy *de Q/P en $]a, b[$, noté $Ind(Q/P, a, b)$ est le nombre de sauts de P/Q de*

$-\infty$ à ∞ dans l'intervalle $]a,b[$ moins le nombre de sauts de ∞ à $-\infty$ dans l'intervalle $]a,b[$.

Lorsque $]a,b[=]-\infty,\infty[$, on note plus simplement l'indice $Ind(Q/P)$. Notre objectif est de calculer cet indice sans chercher les racines. Nous introduisons à cet effet la notion de *variation de signe*.

Définition 42 *Soit $a = a_0, a_1, \ldots, a_d$ une suite de valeurs non nulles, $Var(a)$ la variation de signe de cette séquence est définie par :*
- *$Var(a_0) = 0$*
- *Si $a_0 a_1 > 0$ Alors $Var(a_0, a_1 \ldots a_d) = Var(a_1 \ldots a_d)$*
  Sinon $Var(a_0, a_1 \ldots a_d) = Var(a_1 \ldots a_d) + 1$

Cette notion s'étend à toute suite de valeurs en supprimant les zéros de la suite et en posant $Var(\emptyset) = 0$.

Soit $\mathcal{P} = P_0, P_1, \ldots P_d$ une suite de polynômes (à une variable) et $x \in \mathbb{R} \cup \{-\infty, \infty\}$ la variation de signe de $\mathcal{P}$ en x, notée $Var(\mathcal{P}; x)$ est définie par : $Var(\mathcal{P}; x) = Var(P_0(x), P_1(x) \ldots P_d(x))$.

Soit $\mathcal{P} = P_0, P_1 \ldots P_d$ une suite de polynômes (à une variable) et $]a,b[$ un intervalle non vide, la variation de signe de $\mathcal{P}$ dans $]a,b[$, notée $Var(\mathcal{P}; a, b)$ est définie par $Var(\mathcal{P}; a, b) = Var(\mathcal{P}; a) - Var(\mathcal{P}; b)$.

Comme déjà précisé, le signe d'un polynôme à l'infini est bien défini et par conséquent la définition ci-dessus est correcte. Etant donné $x \in \mathbb{R} \cup \{-\infty, \infty\}$ et P un polynôme, $\sigma_P(x)$ dénotera le signe de $P(x)$.

Lemme 35 *Soient P, Q deux polynômes non nuls et $a < b \in \mathbb{R} \cup \{-\infty, \infty\}$ t.q. ni a ni b ne sont des racines des polynômes de la suite $SRemS(P, Q)$. On pose $R \equiv Rem(P, Q)$. Alors :*
- *Si $\sigma_{PQ}(a)\sigma_{PQ}(b) = 1$ Alors*
  $Var(SRemS(P, Q); a, b) = Var(SRemS(Q, -R); a, b)$
- *Si $\sigma_{PQ}(a)\sigma_{PQ}(b) = -1$ Alors*
  $Var(SRemS(P, Q); a, b) = Var(SRemS(Q, -R); a, b) + \sigma_{PQ}(b)$

Preuve
Soit x qui n'est ni une racine de P ni une racine de Q.
Si $P(x)Q(x) < 0$ alors $Var(SRemS(P, Q); x) = Var(SRemS(Q, -R); x) + 1$.
Sinon $Var(SRemS(P, Q); x) = Var(SRemS(Q, -R); x)$.
Le résultat en découle en examinant tous les cas possibles.

c.q.f.d. $\Diamond\Diamond\Diamond$

Le lemme et le théorème qui suivent expliquent le choix du signe dans la séquence $SRemS$.

Lemme 36 *Soient P, Q deux polynômes non nuls et $a < b \in \mathbb{R} \cup \{-\infty, \infty\}$ t.q. ni a ni b ne sont des racines des polynômes de la suite $SRemS(P, Q)$. On pose $R \equiv Rem(P, Q)$. Alors :*
- *Si $\sigma_{PQ}(a)\sigma_{PQ}(b) = 1$ Alors $Ind(Q/P; a, b) = Ind(-R/Q; a, b)$*
- *Si $\sigma_{PQ}(a)\sigma_{PQ}(b) = -1$ Alors $Ind(Q/P; a, b) = Ind(-R/Q; a, b) + \sigma_{PQ}(b)$*

Preuve
Nous pouvons supposer que P et Q sont premiers entre eux car en posant $D \equiv pgcd(P,Q), P_1 \equiv P/D, Q-1 \equiv Q/D, R_1 \equiv Rem(P_1,Q_1) = R/D$, on a :
$Q_1/P_1 = Q/P, -R_1/Q_1 = -R/Q$
et les signes de $P(x)Q(x)$ et $P_1(x)Q_1(x)$ coïncident en tout point x qui n'est pas une racine de PQ.

Notons n_{-+} (resp. n_{+-}) le nombre de variations de -1 à $+1$ (resp. de $+1$ à -1) du signe de $PQ[X \leftarrow x]$ lorsque x varie de a à b. On remarque que si $\sigma_{PQ}(a)\sigma_{PQ}(b) = 1$ alors $n_{-+} - n_{+-} = 0$ et que sinon $n_{-+} - n_{+-} = \sigma_{PQ}(b)$.

La définition de l'indice de Cauchy entraine que :
$Ind(Q/P;a,b) + Ind(P/Q;a,b) = n_{-+} - n_{+-}$
En remarquant que $Ind(P/Q;a,b) = Ind(R/Q;a,b)$, on conclut.

c.q.f.d. $\Diamond\Diamond\Diamond$

Théorème 11 *Soit $P \neq 0, Q$ deux polynômes et $a < b \in \mathbb{R} \cup \{-\infty, \infty\}$ t.q. ni a ni b ne sont pas des racines de P. Alors :*

$$Var(SRemS(P,Q);a,b) = Ind(Q/P;a,b)$$

Preuve
On laisse le cas évident $Q = 0$ en exercice et on suppose dans la preuve que $Q \neq 0$. Le théorème est presque une conséquence des deux lemmes précédents (moyennant une induction sur la longueur de $SRemS(P,Q)$) à ceci près qu'on exige uniquement de a et de b qu'ils ne soient pas racine de P. Choisissons maintenant a' et b' t.q $a \leq a' < b' \leq b$, $a' = a$ (resp. $b' = b$) si a (resp. b) n'est racine d'aucun polynôme de $SRemS(P,Q)$ et sinon $]a,a']$ (resp. $[b',b[$) ne contient aucune racine d'un polynôme de $SRemS(P,Q)$.

Pour conclure nous allons prouver que $Ind(Q/P;a,b) = Ind(Q/P;a',b')$ et que $Var(SRemS(P,Q);a,b) = Var(SRemS(P,Q);a',b')$. La première égalité provient immédiatement des contraintes sur a', b'.

Notons $SRemS(P,Q) \equiv SRemS_0, SRemS_1, \ldots$. Sachant que a n'est pas une racine de P, on établit par une récurrence trivale que a n'est jamais simultanément racine de deux polynômes consécutifs de cette suite. Lorsque a n'est pas racine d'un polynôme $SRemS_i$ de la suite, a et a' $SRemS_i[X \leftarrow a]$ et $SRemS_i[X \leftarrow a']$ ont même signe. Supposons que a soit racine de $SRemS_i$ alors l'égalité :

$$SRemS_{i+1} = -SRemS_{i-1} + Quo(SRemS_{i-1}, SRemS_i)SRemS_i$$

entraine que $SRemS_{i-1}SRemS_{i+1}[X \leftarrow a] < 0$ et par conséquent

$$Var(SRemS_{i-1}, SRemS_i, SRemS_{i+1}; a) = Var(SRemS_{i-1}, SRemS_{i-1}, SRemS_{i-1}; a')$$

Un raisonnement similaire pour b et b' permet de conclure.

c.q.f.d. $\Diamond\Diamond\Diamond$

Nous en arrivons maintenant à la quantité clef pour l'élimination des quantificateurs.

Définition 43 (Question de Tarski) *Soit $P \neq 0$ et Q deux polynômes. La* question de Tarski *sur Q relative à P dans $]a, b[$ avec $a < b \in \mathbb{R} \cup \{-\infty, \infty\}$ est le nombre :*

$$TaQ(Q, P; a, b) \equiv \sum_{x \in]a,b[, P(x)=0} \sigma_Q(x)$$

On note $TaQ(Q, P) \equiv TaQ(Q, P; -\infty, \infty)$. Remarquons que $TaQ(1, P; a, b)$ est le nombre de racines de P dans l'intervalle $]a, b[$.

La proposition suivante nous fournit ce que nous recherchions.

Théorème 12 (Théorème de Tarski)

$$TaQ(Q, P; a, b) = Ind(P'Q/P; a, b) = Var(SREmS(P, P'Q); a, b)$$

Preuve
La deuxième égalité est celle du théorème 11. Il nous reste à prouver la première égalité. La multiplicité (μ) de x en P est strictement supérieure à celle de $P'Q$ ssi x n'est pas racine de Q et dans ce cas, la différence est égale à 1. On remarque que la fraction rationnelle $P'Q/P$ s'écrit $\mu Q/(X - x) + R$ où R est une fraction rationnelle définie en x. Par conséquent $P'Q/P$ fait un saut de $-\infty$ à ∞ (resp. de ∞ à $-\infty$) si $Q(x) > 0$ (resp. $Q(x) < 0$). D'où le résultat.

c.q.f.d. ◊◊◊

A partir du théorème de Tarski, on peut facilement obtenir une information sur le signe de $Q(x)$ lorsque x parcourt les racines de P. Dans la suite, on se limite à l'intervalle $]-\infty, \infty[$.

Notations. Soient $P \neq 0, Q$ deux polynômes et $\alpha \in \{-1, 0, 1\}$. On note $\sharp(P, Q, \alpha) \equiv |\{P(x) = 0 \wedge \sigma_Q(x) = \alpha\}|$.

Proposition 27

$$\sharp(P, Q, 0) = TaQ(1, P) - TaQ(Q^2, P)$$

$$\sharp(P, Q, 1) = (TaQ(Q^2, P) + TaQ(Q, P))/2$$

$$\sharp(P, Q, -1) = (TaQ(Q^2, P) - TaQ(Q, P))/2$$

Preuve
On résout le système d'équations suivant.

$$TaQ(1, P) = \sharp(P, Q, -1) + \sharp(P, Q, 0) + \sharp(P, Q, 1)$$

$$TaQ(Q, P) = -\sharp(P, Q, -1) + \sharp(P, Q, 1)$$

$$TaQ(Q^2, P) = \sharp(P, Q, -1) + \sharp(P, Q, 1)$$

c.q.f.d. ◊◊◊

On veut généraliser ce résultat lorsque Q est remplacé par une famille de polynômes $\mathcal{Q}$.

Notations. Soient P un polynôme non nul, $\mathcal{Q} = \{Q_1, \ldots, Q_s\}$ une famille finie de polynômes et $\alpha : \{1, \ldots, s\} \mapsto \{-1, 0, 1\}$.
On note :

- $Z(P, \mathcal{Q}, \alpha) \equiv \{P(x) = 0 \wedge \bigwedge_{i \leq s} \sigma_{Q_i}(x) = \alpha(i)\}$
- $\sharp(P, \mathcal{Q}, \alpha) \equiv |Z(P, \mathcal{Q}, \alpha)|$

$\sharp(P, \mathcal{Q})$ est le vecteur dont les composantes sont $\sharp(P, \mathcal{Q}, \alpha)$ indicées par les α ordonnés lexicographiquement.
On généralise aussi la question de Tarski. Soit $\beta : \{1, \ldots, s\} \mapsto \{0, 1, 2\}$. On définit le polynôme $\mathcal{Q}^\beta \equiv \prod_{j \leq s} Q_j^{\beta(j)}$. $TaQ(\mathcal{Q}, P)$ est le vecteur dont les composantes sont $TaQ(\mathcal{Q}^\beta, P)$ indicées par les β ordonnés lexicographiquement.

On remarque qu'à l'intérieur d'un $Z(P, \mathcal{Q}, \alpha)$ le signe d'un polynôme $\mathcal{Q}^\beta$ est fixé. On note $M_s[\alpha, \beta]$ ce signe. La matrice M_s peut se construire itérativement aisi que le lemme suivant l'indique.

Lemme 37 *Soit M_s la matrice associée à la famille $\{Q_1, \ldots, Q_s\}$ et M_{s-1} la matrice associée à la famille $\{Q_1, \ldots, Q_{s-1}\}$. Soit $\alpha : \{2, \ldots, s\} \mapsto \{-1, 0, 1\}$ et $\beta : \{2, \ldots, s\} \mapsto \{0, 1, 2\}$. On a :*

- $M_s[(0, \beta), (-1, \alpha)] = M_{s-1}[\beta, \alpha]$, $M_s[(0, \beta), (0, \alpha)] = M_{s-1}[\beta, \alpha]$
  $M_s[(0, \beta), (1, \alpha)] = M_{s-1}[\beta, \alpha]$
- $M_s[(1, \beta), (-1, \alpha)] = -M_{s-1}[\beta, \alpha]$, $M_s[(1, \beta), (0, \alpha)] = 0$
  $M_s[(1, \beta), (1, \alpha)] = M_{s-1}[\beta, \alpha]$
- $M_s[(2, \beta), (-1, \alpha)] = M_{s-1}[\beta, \alpha]$, $M_s[(2, \beta), (0, \alpha)] = 0$
  $M_s[(2, \beta), (1, \alpha)] = M_{s-1}[\beta, \alpha]$

Autrement dit, la matrice M_s se décompose par bloc comme suit :

M_s	$(-1, -)$	$(0, -)$	$(1, -)$
$(0, -)$	M_{s-1}	M_{s-1}	M_{s-1}
$(1, -)$	$-M_{s-1}$	0	M_{s-1}
$(2, -)$	M_{s-1}	0	M_{s-1}

Preuve
La preuve laissée en exercice s'obtient par un examen cas par cas.

c.q.f.d. $\Diamond\Diamond\Diamond$

On a déjà observé que la matrice M_1 est définie par :

M_1	(-1)	(0)	(1)
(0)	1	1	1
(1)	-1	0	1
(2)	1	0	1

On peut formaliser la construction itérative du lemme précédent à l'aide de l'opérateur tensoriel $\otimes$ des matrices.

Définition 44 *Soient M une matrice $m \times n$ et M' une matrice $m' \times n'$ la matrice $M \otimes M'$ $mm' \times nn'$ est définie par $M \otimes M'[(i, i'), (j, j')] \equiv M[i, j]M'[i', j']$. Autrement dit, la matrice bloc associée à i et j est définie par $M[i, j]M'$.*

Avec cette formulation, on observe que $M_s = M_1 \otimes M_{s-1}$. L'intérêt de cette formulation est le résultat élementaire suivant : si M et M' sont inversibles alors $M \otimes M'$ est inversible d'inverse $M^{-1} \otimes M'^{-1}$. On en conclut que M_s est inversible. Une conséquence de ce résultat est mise en évidence par le lemme suivant.

Lemme 38

$$TaQ(\mathcal{Q}, P) = M_s \cdot \sharp(P, \mathcal{Q})$$

et par conséquent

$$\sharp(P, \mathcal{Q}) = M_s^{-1} \cdot TaQ(\mathcal{Q}, P)$$

Preuve
On remarque que les zéros de P sont répartis dans la partition $\{Z(P, \mathcal{Q}, \alpha)\}_\alpha$. La matrice M_s donnant le signe d'un polynôme $\mathcal{Q}^\beta$ en fonction d'un élement de la partition, le résultat s'ensuit immédiatement.

c.q.f.d. ◊◊◊

La proposition suivante (dont la preuve évidente est laissé en exercice) synthétise les résultats obtenus.

Proposition 28 *Soient P un polynôme non nul, $\mathcal{Q} = \{Q_1, \ldots, Q_s\}$ une famille finie de polynômes et $\alpha : \{1, \ldots, s\} \mapsto \{-1, 0, 1\}$. $\sharp(P, \mathcal{Q}, \alpha)$ dépend de manière effective des degrés des polynômes et du signe de leur plus haut coefficient des séquences $SRemS(P, P'\mathcal{Q}^\beta)$ pour tous les β possibles.*

Il nous reste à traiter le cas où le polynôme P est nul et α est à valeurs dans $\{-1, 1\}$.

Proposition 29 *Soient $\mathcal{Q} = \{Q_1, \ldots, Q_s\}$ une famille finie de polynômes et $\alpha : \{1, \ldots, s\} \mapsto \{-1, 1\}$. Posons $C \equiv \prod_{i \leq s} Q_i$. La vacuité de l'ensemble $Z(\mathcal{Q}, \alpha) \equiv \{x \mid \bigwedge_{i \leq s} \sigma_{Q_i}(x) = \alpha_i\}$ dépend de manière effective des degrés des polynômes de $\mathcal{Q}$ et du signe de leur plus haut coefficient des degrés des polynômes et du signe de leur plus haut coefficient de la séquence $SRemS(C, C')$ et des séquences $SRemS(C', C''\mathcal{Q}^\beta)$ pour tous les β possibles.*

Preuve
Rappelons que le nombre de racines de C est déterminé par les degrés des polynômes et le signe de leur plus haut coefficient de la séquence $SRemS(C, C')$.

- Si C n'a pas de racine alors tout $Q \in \mathcal{Q}$ a un signe constant déterminé par son plus haut coefficient.
- Si C a une unique racine alors il y a uniquement deux α possibles déterminés par le degré de chaque polynôme $Q \in \mathcal{Q}$ et le signe du coefficient de plus haut degré.
- Si C a au moins deux racines alors entre deux racines de C, il y a au moins une racine de C'. Par conséquent les α possibles sont déterminés en $-\infty$ et en ∞ par le degré de chaque polynôme $Q \in \mathcal{Q}$ et le signe du coefficient de plus haut degré et dans les intervalles entre les racines par les degrés des polynômes et du signe de leur plus haut coefficient des séquences $SRemS(C', C''\mathcal{Q}^\beta)$ pour tous les β possibles.

c.q.f.d. ◊◊◊

4.6.3 Elimination des quantificateurs

Théorème 13 *La théorie T des corps réels fermés permet l'élimination des quantificateurs.*

Preuve
Soit une formule,
$\varphi \equiv \exists X \ \bigwedge_{P\in\mathcal{P}}, P = 0 \wedge \bigwedge_{Q\in\mathcal{Q}} Q > 0$ avec $P, Q \in \mathbb{Z}[X_1, \ldots, X_n][X]$.

On remplace la formule φ par une disjonction de formules sans quatificateurs portant sur les variables $X_1, \ldots, X_n$.

- Le premier type de formule correspond à la conjonction des atomes $a_i = 0$ (où les a_i sont les coefficients de P) et d'une formule obtenue en appliquant la proposition 29 qui garantit que les signes de $Q \in \mathcal{Q}$ sont positifs en au moins un point. Ces formules se calculent à l'aide de la construction des arbres $TRemS$ appliqués aux séquences $PSRemS(C', C''\mathcal{Q}^\beta)$ pour tous les β possibles .
- Le deuxième type de formule correspond à une conjonction d'atomes qui fixe le degré de $P \neq 0$ et d'une formule obtenue en appliquant le lemme 38 qui garantit que les signes de $Q \in \mathcal{Q}$ sont positifs en au moins un zéro de P. Ces formules se calculent à l'aide de la construction des arbres $TRemS$ appliqués aux séquences $PSRemS(P, P'\mathcal{Q}^\beta)$ pour tous les β possibles .

c.q.f.d. ◊◊◊

Corollaire 9 *Le problème de savoir si φ, une formule est une conséquence de la théorie T des corps algébriquement clos est décidable.*

Preuve
On peut considérer que φ est une formule close en raison du lemme 8. La transformation ne crée pas de variables donc la formule équivalente est nécessairement une combinaison booléenne d'atomes $t = 0$ ou $t > 0$ avec t un terme sans variable (i.e. un élément de $\mathbb{Z}$). Puisque la caractéristique d'un corps réel fermé est nulle (vérifiez-le) les atomes clos s'évaluent comme dans $\mathbb{Z}$.

c.q.f.d. ◊◊◊

A titre d'exemple, nous donnons une application de ce résultat.

Proposition 30 *Soient ι un modèle de corps réel fermé et φ une fomule de cette logique comportant une unique variable libre x. Alors l'ensemble $E^\iota_\varphi \equiv \{a \in E_\iota \mid \varphi^\iota(x \leftarrow a) = \texttt{true}\}$ est une union finie de points et d'intervalles ouverts.*

Preuve
Sans perte de généralité, nous considérons une formule φ sans quantificateur. S'il s'agit d'un atome $P = 0$, E^ι_φ est l'ensemble fini des racines. S'il s'agit d'un atome $P > 0$ alors E^ι_φ est l'union des intervalles ouverts délimités par les racines (et $-\infty, \infty$) sur lesquels P est positif.

On démontre la propriété par induction. Si $\varphi \equiv \varphi' \wedge \varphi''$, en inversant l'intersection et les unions, cela revient à démontrer que l'intersection de deux points ou intervalles est encore soit un point soit un intervalle soit l'ensemble vide (ce qui est trivial). Soit $\varphi \equiv \neg\varphi'$. D'une part, le complémentaire d'un point est une

union de deux intervalles. D'autre part, le complémentaire d'un intervalle est une union d'au plus deux intervalles et d'au plus deux points. Pour conclure, il suffit de distribuer la complémentation ce qui nous ramène au cas de l'intersection déjà traité.

c.q.f.d. ◊◊◊

Il existe des corps réels fermés non isomorphes au corps des réels. Une série de Puiseux est soit 0 soit une série de la forme $\sum_{i \geq k} a_i \varepsilon^{i/q}$ avec $i, k \in \mathbb{Z}$, $a_i \in \mathbb{R}$, $q \in \mathbb{N}^*$ et $a_k \neq 0$. Une série de Puiseux est positive si $a_k > 0$. On peut démontrer (mais c'est difficile) que l'ensemble des series de Puiseux, noté $R\langle\langle\varepsilon\rangle\rangle$, est un corps réel fermé. On remarque que $\varepsilon < r$ pour tout $r \in \mathbb{R}^{+*}$. Par conséquent ce corps n'est pas archimédien. On obtient donc :

Proposition 31 *La pseudo-formule* $\forall x\ x > 0 \Rightarrow \bigvee_{n \in \mathbb{N}} nx - 1 > 0$ *n'est équivalente à aucune formule (close) dans la théorie des corps réels fermés.*

Preuve
Les formules closes de la théorie des corps réels fermés ont même valeur de vérité pour tout corps réel fermé ce qui n'est pas le cas de la pseudo-formule.

c.q.f.d. ◊◊◊

4.7 TD n°4

Question n°1. Démontrer la proposition 25.

Question n°2. Démontrer la proposition 26.

Question n°3. Démontrer le lemme 32.

Question n°4. Démontrer le théorème 10.

Question n°5. Démontrer le lemme 33.

Question n°6. Démontrer le lemme 34.

Chapitre 5

Les théorèmes d'incomplétude de Gödel

Remarque. Dans ce chapitre, on considère que $\wedge, \vee, \Leftrightarrow, \exists$ sont des abréviations (par exemple $\varphi \vee \psi$ est une abréviation de $\neg\psi \Rightarrow \varphi$).

5.1 Le premier théorème de Gödel(-Tarski)

5.1.1 « Cet énoncé est faux »

Définition 45 *Soit Λ un alphabet fini et $L \subseteq \Lambda^*$ un langage. g, une bijection de L dans $\mathbb{N}$, est appelée une* numérotation de Gödel.

Notation. Pour n un entier quelconque, on notera $E_n = g^{-1}(n)$. Le nombre n est appelé le numéro de Gödel de E_n.

Le langage L qui nous intéresse sera un sur-ensemble d'un langage de formules de la logique du premier ordre relatif à l'arithmétique. On supposera dans la suite que $\{\Rightarrow, \neg, (,), \forall, v, ', f, \prime, =, 0\} \subseteq \Lambda$. L'addition $+$ (resp. la multiplication $\times$) est une abréviation de $(f_\prime)$ (resp. $(f_{\prime\prime})$) et sera notée en infixé. La fonction successeur $'$ sera notée en postfixé. On abrégera $0'\ldots'$ où le symbole $'$ est répété n fois par $\underline{n}$. De même, on abrégera $(v_\prime \ldots \prime)$ où le symbole $\prime$ est répété $n(>0)$ fois par v_n. Afin de retrouver une notation usuelle de variables x, y, z désignent v_1, v_2, v_3.

On dira qu'un support est *relatif à* $\mathbb{N}$ s'il contient la constante 0 et la fonction successeur $'$ notée en postfixé. Un support est relatif à l'arithmétique s'il contient 0, $'$, $+$ et $\times$.

Etant donnés $E, t \in \Lambda^*$, on note $E[t]$ le mot $\forall x(x = t \Rightarrow E)$ *(notez bien les crochets)*. Soient φ une formule et t un terme clos, remarquons que $\varphi[t]$ est logiquement équivalente à $\varphi(v_1 \leftarrow t)$ (*i.e.* $\varphi[t] \Leftrightarrow \varphi(v_1 \leftarrow t)$ est un théorème de la logique du premier ordre). L'avantage de la construction de formule $\varphi[t]$ est qu'elle ne fait pas appel à la notion d'occurrence libre de variable.

On suppose que si $E \in L$ alors $E[\underline{n}] \in L$. Ceci nous conduit à la définition suivante en supposant g fixée.

Définition 46 *d, la fonction de $\mathbb{N}$ dans $\mathbb{N}$ qui à n associe le numéro de Gödel de $E_n[\underline{n}]$ est appelée la (fonction) diagonale de Gödel. Soit $A \subseteq \mathbb{N}$, alors $A^d \subseteq \mathbb{N}$ est défini par $d^{-1}(A)$.*

On se donne une interprétation ι dans $\mathbb{N}$ de manière standard $0^\iota = 0$, $x' = x + 1$, $x +^\iota y = x + y$, $x \times^\iota y = xy$. Dans la suite, lorsqu'on écrira $\mathbb{N} \models \varphi$ il sera sous-entendu que cette satisfaction est relative à cette interprétation. On obtient déjà un premier résultat *diagonal*. Ce résultat est en réalité indépendant de l'interprétation choisie pour $\mathbb{N}$.

Définition 47 *Etant donnée une formule φ avec au plus x comme unique variable libre sur un support Supp relatif à $\mathbb{N}$ et $A \subseteq \mathbb{N}$, on dit que A est* exprimé *par φ si :*

$$n \in A \text{ ssi } \mathbb{N} \models \varphi[\underline{n}]$$

Par exemple, l'ensemble des nombres pairs supérieurs ou égaux à 17 est exprimable par la formule $\exists y\ \exists z\ x = \underline{2} \times y \wedge \underline{17} + z = x$.

Notation. Soit A un sous-ensemble d'un ensemble fixé alors $\overline{A}$ est son complémentaire.

Théorème 14 *Soit $\mathcal{TRUE}$ l'ensemble des numéros de Gödel des formules sur un support quelconque relatif à $\mathbb{N}$ satisfaites par $\mathbb{N}$, alors $(\overline{\mathcal{TRUE}})^d$ n'est pas exprimable par une formule de ce support.*

Preuve
Remarquons que $(\overline{\mathcal{TRUE}})^d = \{m \mid E_m \notin F_1 \vee \mathbb{N} \not\models E_m[\underline{m}]\}$.
Rappelons que $d(n)$ est le numéro de Gödel de $E_n[\underline{n}]$. Supposons maintenant qu'il existe E_n une formule qui exprime $(\overline{\mathcal{TRUE}})^d$.
Alors :
$\mathbb{N} \models E_n[\underline{n}]$ ssi *(par le choix de E_n)*
$n \in (\overline{\mathcal{TRUE}})^d$ ssi *(par la définition de $()^d$)*
$d(n) \in \overline{\mathcal{TRUE}}$ ssi *(par complémentation)*
$d(n) \notin \mathcal{TRUE}$ ssi *(par la définition de $\mathcal{TRUE}$ et de $d()$)*
$\mathbb{N} \not\models E_n[\underline{n}]$ *(car E_n est une formule)*
Ce qui est absurde.

c.q.f.d. $\Diamond\Diamond\Diamond$

Remarques.
- Informellement, la formule $E_n[\underline{n}]$ exprime « Cet énoncé est faux » qui conduit à un paradoxe évident.
- Nous n'avons utilisé aucune hypothèse sur $\mathbb{N}$ et pratiquement aucune hypothèse sur la numérotation de Gödel. Il s'agit donc d'un résultat essentiellement logique.

5.1.2 Une numéroration de Gödel

L'alphabet $\Lambda = \{', 0, (,), f, \iota, v, \neg, \Rightarrow, \forall, =, \flat, \sharp\}$ est constitué de 13 caractères. Les deux derniers caractères sont introduits « en cas de besoin ». Le fait que 13 est premier simplifiera certains développements.

Le langage d'expressions $L = (\Lambda \setminus \{'\})\Lambda^* \cup \{'\}$ est constitué des chaînes non vides ne commençant pas $'$ exceptée la chaîne réduite à $'$.

Soit une expression $w = w_0 w_1 \dots w_n$, alors $g(w) = \sum_{i=1}^{n} 13^{n-i} g(w_i)$ avec :
$g(') = 0$, $g(0) = 1$, $g(() = 2$, $g()) = 3$ $g(f) = 4$
$g(,) = 5$, $g(v) = 6$, $g(\neg) = 7$, $g(\Rightarrow) = 8$
$g(\forall) = 9$, $g(=) = 10$, $g(\flat) = 11$ $g(\sharp) = 12$.

Nous laissons le soin au lecteur de vérifier que g est bien une bijection. Par exemple $g(v = 0) = 4 \times 13^2 + 10 \times 13^2 + 1$. Si on prend le soin de compter les chiffres en base 13 en introduisant les symboles $\eta, \varepsilon, \delta$ pour $10, 11, 12$ alors la numéroration de Gödel n'est rien d'autre que le nombre obtenu en concaténant les numéros de Gödel des symboles de l'expression ! Ainsi $g(v = 0) = 2\eta 1$.

Notations. La formule $t \leq t'$ est une abréviation de $\exists v_i (t + v_i = t')$ pour n'importe quel v_i n'apparaissant ni dans t ni dans t'. La formule $t \neq t'$ est une abréviation de $\neg t = t'$.

Nous introduisons des sous-ensembles de formules dont le rôle s'avérera primordial dans les différents théorèmes d'incomplétude.

Définition 48 (Formules Σ_{0^-}) *L'ensemble de formules Σ_{0^-} est défini inductivement par :*

- *Si a, b et c désignent un nombre ($\underline{n}$) ou une variable alors $a+b = c, a \times b = c, a = b, a \leq b$ appartiennent à Σ_{0^-}.*
- *Si φ et ψ appartiennent à Σ_{0^-} alors $\neg\varphi$ et $(\varphi \Rightarrow \psi)$ appartiennent à Σ_{0^-}.*
- *Si φ appartient à Σ_{0^-}, v_i est une variable et t désigne soit un nombre ($\underline{n}$), soit une variable* différente *de v_i alors $\forall v_i (v_i \leq t \Rightarrow \varphi)$ appartient à Σ_{0^-}.*

On introduit deux nouvelles abréviations : $[\forall v_i \leq t](\varphi)$ comme abréviation de $\forall v_i (v_i \leq t \Rightarrow \varphi)$ et $[\exists v_i \leq t](\varphi)$ comme abréviation de $\neg[\forall v_i \leq t](\neg\varphi)$.

Définition 49 (Formules Σ_1) *L'ensemble de formules Σ_1 est défini par :*
$\Sigma_1 = \{\exists v_i(\varphi) \mid \varphi \in \Sigma_{0^-}\}$

Définition 50 (Formules Σ) *L'ensemble de formules Σ est défini inductivement par :*

- *Toute formule Σ_{0^-} est une formule Σ.*
- *Si φ et ψ appartiennent à Σ alors $(\varphi \vee \psi)$ et $(\varphi \wedge \psi)$ appartiennent à Σ.*
- *Si φ appartient à Σ_{0^-} et ψ appartient à Σ alors $(\varphi \Rightarrow \psi)$ appartient à Σ.*
- *Si φ appartient à Σ, v_i est une variable et t désigne soit un nombre ($\underline{n}$), soit une variable* différente *de v_i alors $[\forall v_i \leq t](\varphi)$ et $[\exists v_i \leq t](\varphi)$ appartiennent à Σ.*
- *Si φ appartient à Σ et v_i est une variable alors $\exists v_i(\varphi)$ appartient à Σ.*

On reprend (en la spécialisant) la notion d'ensemble et de relation *exprimable*.

Définition 51 *Soit φ une formule dont l'ensemble des variables libres est contenu dans $\{v_1, \dots, v_n\}$, φ exprime R une relation n-aire de $\mathbb{N}$ si :*

$$\mathbb{N} \models \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \in 1..n}) \text{ ssi } R(k_1, \dots, k_n)$$

Soit φ une formule dont l'ensemble des variables libres est contenu dans $\{v_1, \ldots, v_n, v_{n+1}\}$, φ exprime f une fonction n-aire de $\mathbb{N}$ si :

$$\mathbb{N} \models \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \in 1..n+1}) \text{ ssi } k_{n+1} = f(k_1, \ldots, k_n)$$

Soit φ une formule avec au plus v_1 comme variable libre φ exprime $\mathcal{A} \subseteq \mathbb{N}$ un ensemble si :

$$\mathbb{N} \models \varphi(v_1 \leftarrow \underline{k}) \text{ ssi } k \in A$$

On dira que R une relation (resp. f une fonction, A un ensemble) est *arithmétique* s'il existe une formule l'exprimant.

Si la formule utilisée appartient à Σ_{0^-} (resp. Σ_1, Σ) on dira que la relation est Σ_{0^-}-arithmétique (resp. Σ_1-arithmétique, Σ-arithmétique) ou encore de manière abrégée qu'elle est Σ_{0^-} (resp. Σ_1, Σ).

On sera très souvent amené à vérifier que des relations, fonctions ou ensembles sont arithmétiques. Dans ce qui suit, *b désigne un nombre premier.*

Définition 52 *Voici quelques exemples de relations et fonctions arithmétiques.*

La relation binaire $x < y$ se définit par $x \leq y \wedge x \neq y$.

La relation binaire $div(x, y)$ qui indique que x divise y se définit par : $[\exists z \leq y](y = x \times z)$.

La relation unaire $Puiss_b(x)$ qui dénote le fait que x est une puissance de b se définit par $x \neq 0 \wedge [\forall y \leq x](y \neq \underline{1} \wedge div(y, x) \Rightarrow div(\underline{b}, y))$

La fonction qui à x associe $y = b^l$ où l est la longueur en base b de x, notée $pl_b(x)$ se définit par $y = pl_b(x)$ si :

$$(0 = x \wedge \underline{b} = y) \vee$$
$$(0 < x \wedge Puiss_b(y) \wedge x < y \wedge [\forall z \leq y]((z < y \wedge Puiss_b(z)) \Rightarrow z \leq x))$$

La fonction binaire $x *_b y = z$ représentant le nombre obtenu par la concaténation en base b de x suivi de y se définit par $x *_b y = z$ si :

$$[\exists v_4 \leq z][\exists v_5 \leq z](v_5 = pl_b(y) \wedge z = v_4 + y \wedge v_4 = x \times v_5)$$

Observons qu'on a utilisé implicitement le fait qu'on peut introduire une relation ou une fonction arithmétique pour en définir d'autres. D'autre part, toutes les relations et fonctions définies jusqu'ici sont Σ_{0^-} *(vérifiez-le).*

Remarquons que $*_b$ n'est pas associative. Nous prenons la convention que $v_1 *_b v_2 *_b \ldots *_b v_n$ est une abréviation de $(\ldots(v_1 *_b v_2) *_b v_3) *_b \ldots) *_b v_n$.

L'intérêt de l'opérateur $*_b$ vis à vis de la numérotation de Gödel est évident. Soit $E = w_1 \ldots w_n$ une expression de L, alors $g(E) = g(w_1) *_{13} \ldots *_{13} g(w_n)$. Par une récurrence immédiate, on obtient :

Proposition 32 *La fonction n-aire $v_1 *_b v_2 *_b \ldots *_b v_n$ est arithmétique (Σ_{0^-}).*

On peut aussi définir arithmétiquement de propriétés de chaines de chiffres.

Proposition 33 *Les relations suivantes sont arithmétiques (Σ_{0^-}).*

- *xC_by qui indique que l'écriture du nombre x débute l'écriture du nombre y en base b.*
- *xT_by qui indique que l'écriture du nombre x termine l'écriture du nombre y en base b.*
- *xP_by qui indique que l'écriture du nombre x est une partie de l'écriture du nombre y en base b.*

Preuve
Nous établissons uniquement le cas C_b qui s'exprime par la formule :

$$x = y \vee (x \neq 0 \wedge [\exists z \leq y][\exists v_4 \leq y] Puiss_b(v_4) \wedge (x \times v_4) *_b z = y)$$

Pour comprendre l'intérêt de la variable v_4, traitez le cas $x = 3$ et $y = 3001$.

c.q.f.d. ◊◊◊

Nous allons maintenant construire une relation arithmétique d'une grande importance. Notez bien que la proposition exige simplement qu'elle ait une « bonne » propriété. D'autres définitions que celles de la preuve sont donc possibles. Informellement, cette relation permet de coder des suites de couples à l'aide d'un nombre.

Proposition 34 *Il existe une relation arithmétique Σ_{0^-} que nous noterons $K(x, y, z)$ qui possède les deux propriétés suivantes :*

- *Pour toute suite de couples $(a_1, b_1), \ldots, (a_n, b_n)$, il existe un nombre z tel que $K(x, y, z)$ est vrai ssi $\exists i \leq n, (a_i, b_i) = (x, y)$.*
- *Si $K(x, y, z)$ est vrai alors $x \leq z$ et $y \leq z$.*

Preuve
Comme précédemment, on raisonne sur une représentation des nombres en base $b = 13$. On omet l'indice b dans les relations relations P_b, C_b, T_b et on omet la fonction $*_b$. Autrement dit, xy signifie $(x *_b y)$.

Dans cette preuve, on appelle *séparateur* un nombre de la forme $21 \ldots 12$. $un(x)$ qui dénote le fait que x s'écrit comme une suite de 1 est une relation Σ_{0^-} : $un(x) \equiv x \neq 0 \wedge [\forall y \leq x](yPx \Rightarrow \underline{1}Py)$.

Etant donnée une suite $(a_1, b_1), \ldots, (a_n, b_n)$ les « z » recherchés s'écriront sous la forme $wwa_1wb_1ww \ldots wwa_nwb_nww$ avec w un séparateur plus grand que tous les séparateurs présents dans les a_i et les b_i. Pour repérer ce séparateur, on introduit la relation Σ_{0^-}, $sm(x, y)$ qui indique que x est le séparateur maximal présent dans y.

$$sm(x, y) \equiv [\exists z \leq x] un(z) \wedge x = \underline{2}z\underline{2} \wedge xPy \wedge \neg[\exists v_4 \leq y](un(v_4) \wedge \underline{2}zv_4\underline{2}Py)$$

La formule associée à K est maintenant facile à définir :

$$K(x, y, z) \equiv [\exists v_4 \leq x] sm(v_4, z) \wedge v_4v_4xv_4yv_4v_4Pz \wedge \neg v_4Px \wedge \neg v_4Py$$

La deuxième propriété de la proposition est obtenue par construction de K.

c.q.f.d. ◊◊◊

A l'aide de la relation K, on en déduit deux nouvelles relations utiles pour la démonstration du théorème de Gödel-Tarski.

Proposition 35 *La relation $x^y = z$ est Σ_1.*

Preuve
L'idée est de « coder » la suite $(0, 1), (1, x), (2, x^2), \ldots, (y, x^y)$ à l'aide de la relation K et d'une contrainte sur tout couple (a, b) qui apparaît dans la suite : soit $(a, b) = (0, 1)$ soit il existe un autre couple (c, d) tel que $a = c+1$ et $b = d \times x$. Autrement dit :

$$\exists v_4 \; K(y, z, v_4) \wedge [\forall v_5 \leq v_4][\forall v_6 \leq v_4] K(v_5, v_6, v_4) \Rightarrow$$
$$(v_5 = 0 \wedge v_6 = \underline{1}) \vee ([\exists v_7 \leq v_4][\exists v_8 \leq v_4](K(v_7, v_8, v_4) \wedge v_5 = v_7 + \underline{1} \wedge v_6 = v_8 \times x))$$

c.q.f.d. ◊◊◊

Gödel avait obtenu d'une autre manière une fonction β similaire à celle de la prochaine proposition qui permet de coder des suites de nombres.

Proposition 36 *Il existe une fonction arithmétique Σ_{0^-} que nous noterons $\beta(x,y)$ qui possède la propriété suivante : pour toute suite de nombres $a_0,\ldots,a_n$, il existe un nombre w tel que $\forall i \leq n, \beta(w,i) = a_i$.*

Preuve
Etant donnée une suite de nombres $a_0,\ldots,a_n$, l'idée est de « coder » la suite $(0,a_0),(1,a_1),(2,a_2),\ldots,(n,a_n)$ à l'aide de la relation K et de récupérer a_i comme étant le plus petit nombre c (en fait unique) tel que le couple (i,c) apparaisse dans la suite.

$$y = \beta(z,x) \equiv K(x,y,z) \wedge [\forall v_4 \leq y](v_4 = y \vee \neg K(x,v_4,z))$$
$$\vee([\forall v_4 \leq z]\neg K(x,v_4,z) \wedge y = 0)$$

Dans le cas contraire, β est nulle.

c.q.f.d. $\Diamond\Diamond\Diamond$

Soit maintenant $r(m,n)$ la fonction qui fournit le numéro de Gödel de l'expression $E_m[\underline{n}] \equiv \forall v_1(v_1 = \underline{n} \Rightarrow E_m)$. On a :
$r(m,n) = g(\forall v_1(v_1 =) *_{13} 13^n *_{13} g(\Rightarrow) *_{13} m *_{13} g())$
Par conséquent r est arithmétique et la diagonale de Gödel $d(x) \equiv r(x,x)$ est aussi arithmétique. D'où le théorème de Tarski suivant.

Théorème 15 *$\mathcal{TRUE}$ l'ensemble des numéros de Gödel des formules satisfaites par $(\mathbb{N},+,\times)$ n'est pas arithmétique.*

Preuve
Supposons que $\mathcal{TRUE}$ soit exprimable par la formule φ. alors $\overline{\mathcal{TRUE}}$ est exprimable par la formule $\neg\varphi$. Appelons φ_d la formule qui exprime $x = d(y)$. Sans perte de généralité, on suppose que les occurrences liées de variables dans φ sont toutes différentes de y. Alors $(\overline{\mathcal{TRUE}})^d$ est exprimable par la formule :
$\exists y\ \varphi_d \wedge \neg\varphi(x \leftarrow y)$
Ce qui contredit le théorème 14.

c.q.f.d. $\Diamond\Diamond\Diamond$

5.1.3 Relations Σ, Σ_1 et récursivement énumérables

Par construction, une relation Σ est une relation Σ_1. La proposition suivante montre que ces deux notions coïncident.

Proposition 37 *Toute relation Σ est une relation Σ_1.*

Proposition 38 *Toute relation récursivement énumérable est une relation Σ.*

Proposition 39 *Toute relation Σ_{0^-} est récursive. Toute relation Σ_1 est récursivement énumérable.*

Notations. Soit T une théorie du premier ordre, on note $\mathcal{TH}_T$ l'ensemble des numéros de Gödel des théorèmes de T.

Nous prouvons maintenant la première version du théorème de Gödel.

Théorème 16 (1er Théorème d'incomplétude - version Tarski) *Soit T une théorie arithmétique (i.e. sur le support $0, ', +, \times$) récursivement énumérable telle que $\mathbb{N}$ soit un modèle de T. Alors T n'est pas syntaxiquement complet.*

Preuve
Puisque $\mathbb{N}$ est un modèle de T, $\mathcal{TH}_T \subseteq \mathcal{TRUE}$. D'après le lemme 3 du chapitre 1, les théorèmes de T sont récursivement énumérables. Puisque la numérotation de Gödel est récursive, $\mathcal{TH}_T$ est récursivement énumérable. Donc d'après la proposition 38, $\mathcal{TH}_T$ est arithmétique. Puisque $\mathcal{TRUE}$ n'est pas arithmétique (théorème 15). $\mathcal{TRUE} \setminus \mathcal{TH}_T \neq \emptyset$. Soit $\varphi \in \mathcal{TRUE} \setminus \mathcal{TH}_T$, $T \not\vdash \varphi$ et $\mathbb{N} \models \varphi$. Soit φ' une cloture universelle de φ, on a aussi $T \not\vdash \varphi'$ et $\mathbb{N} \models \varphi'$. Donc $\mathbb{N} \not\models \neg\varphi'$ et par conséquent $T \not\vdash \neg\varphi'$. T est donc syntaxiquement incomplet.

c.q.f.d. $\Diamond\Diamond\Diamond$

5.2 Le premier théorème de Gödel

L'objectif de cette section est d'obtenir des conditions portant sur une théorie relative à l'arithmétique qui conduisent à l'incomplétude syntaxique sans même exiger que $\mathbb{N}$ soit un modèle de cette théorie.

5.2.1 « Cet énoncé n'est pas démontrable »

Etant donnée une théorie T, on dira qu'une formule est *prouvable* dans T si c'est un théorème de T et *réfutable* dans T si sa négation est un théorème de T.

Il nous faut un analogue à la notion de « relation exprimable ».

Définition 53 *Soient T une théorie et φ une formule dont l'ensemble des variables libres est contenu dans $\{v_1, \ldots, v_n\}$,* φ représente *R une relation n-aire de $\mathbb{N}$ dans T si :*

$$T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \in 1..n}) \text{ ssi } R(k_1, \ldots, k_n)$$

Soient T une théorie et φ une formule avec au plus v_1 comme variable libre φ représente $\mathcal{A} \subseteq \mathbb{N}$ un ensemble dans T si :

$$T \vdash \varphi(v_1 \leftarrow \underline{k}) \text{ ssi } k \in A$$

On dira qu'une relation (ou un ensemble) est *représentable* dans T s'il existe une formule qui la représente dans T.

Proposition 40 *Soit T une théorie cohérente et supposons que $\neg E_n$ représente $(\mathcal{TH}_T)^d$. Alors $E_n[\underline{n}]$ (et par conséquent $E_n(x \leftarrow \underline{n})$) n'est ni prouvable, ni réfutable dans T.*

Preuve
Supposons que $E_n[\underline{n}]$ soit réfutable dans T, alors puisque $\neg E_n$ représente $(\mathcal{TH}_T)^d$, $n \in (\mathcal{TH}_T)^d$ et, par définition de $()^d$, $E_n[\underline{n}]$ est prouvable. Ce qui contredit la cohérence.

Supposons que $E_n[\underline{n}]$ soit prouvable dans T, alors, par cohérence $\neg E_n[\underline{n}]$ n'est pas prouvable. Puisque $\neg E_n$ représente $(\mathcal{TH}_T)^d$, $n \notin (\mathcal{TH}_T)^d$ et, par définition de $()^d$, $E_n[\underline{n}]$ n'est pas prouvable. Ce qui contredit l'hypothèse.

c.q.f.d. $\Diamond\Diamond\Diamond$

Informellement $E_n[\underline{n}]$ signifie « Cet énoncé n'est pas démontrable ». D'où le résultat. Le corollaire immédiat qui nous intéresse est le suivant.

Proposition 41 *Soit T une théorie cohérente telle que $(\mathcal{TH}_T)^d$ soit représentable alors T est syntaxiquement incomplet.*

Preuve
Soit φ la formule qui représente $(\mathcal{TH}_T)^d$, alors $\neg\neg\varphi$ représente aussi $(\mathcal{TH}_T)^d$ (complétude sémantique du calcul propositionnel). Soit $E_n \equiv \neg\varphi$, d'après la proposition précédente $E_n[\underline{n}]$ n'est ni démontrable ni réfutable.

c.q.f.d. $\Diamond\Diamond\Diamond$

Le lemme suivant explicite un résultat déjà utilisé.

Lemme 39 *Soit T une théorie arithmétique récursivement énumérable alors $(\mathcal{TH}_T)^d$ est récursivement énumérable.*

Preuve
Ceci découle du fait que l'ensemble des théorèmes d'une théorie r.e. est r.e., que la numérotation et la fonction diagonale de Gödel sont récursives. Par exemple, dans une boucle infinie à l'étape n, on énumère les n premiers numéros de théorèmes, l'image de $[1, n]$ par d et on affiche leur intersection.

c.q.f.d. $\Diamond\Diamond\Diamond$

Aussi pour exhiber des conditions d'incomplétude syntaxique, nous allons chercher quelles conditions sur une théorie assurent que tout ensemble Σ_1 est représentable.

5.2.2 Ensembles énumérables

Nous introduisons d'abord une nouvelle notion liée à l'expression des ensembles.

Définition 54 *Soient T une théorie et φ une formule dont l'ensemble des variables libres est contenu dans $\{v_1, \ldots, v_n, v_{n+1}\}$,* φ énumère *R une relation n-aire de $\mathbb{N}$ dans T si :*
- *Si $R(k_1, \ldots, k_n)$ alors $\exists k_{n+1} \in \mathbb{N}$ tel que $T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$*
- *Si $\neg R(k_1, \ldots, k_n)$ alors $\forall k_{n+1} \in \mathbb{N}$ on a $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$*

Soient T une théorie arithmétique et φ une formule avec au plus x, y comme variables libres φ énumère *$\mathcal{A} \subseteq \mathbb{N}$ un ensemble dans T si :*
- *Si $k \in A$ alors $\exists n \in \mathbb{N}$ tel que $T \vdash \varphi(\{x \leftarrow \underline{k}, y \leftarrow \underline{n}\})$*
- *Si $k \notin A$ alors $\forall n \in \mathbb{N}$ on a $T \vdash \neg\varphi(\{x \leftarrow \underline{k}, y \leftarrow \underline{n}\})$*

L'intérêt de l'énumérabilité est justifié par ce lemme.

Lemme 40 *Soit T une théorie arithmétique qui démontre toutes les formules closes Σ_{0^-} satisfaites par $\mathbb{N}$, alors toutes les relations (et les ensembles) Σ_1 sont énumérables.*

Preuve
Soit R une relation n-aire Σ_1 et $\exists v_{n+1}\varphi$ une formule qui l'exprime avec l'ensemble des variables libres de la formule Σ_{0^-} φ contenu dans $\{v_1, \ldots, v_{n+1}\}$.

Supposons que $R(k_1, \ldots, k_n)$ soit vérifiée, alors il existe k_{n+1} tel que $\mathbb{N} \models \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$. Par hypothèse du lemme $T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$.

Supposons que $R(k_1, \ldots, k_n)$ ne soit pas vérifiée, alors pour tout k_{n+1}, $\mathbb{N} \models \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$. Par hypothèse du lemme $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$.

Par conséquent, φ énumère R.

c.q.f.d. $\Diamond\Diamond\Diamond$

Arrivé à ce point du développement, le lecteur attentif aura compris que pour parvenir au résultat d'incomplétude, il nous faut :

- une condition sur la théorie qui assure que toutes les formules closes Σ_{0^-} satisfaites par $\mathbb{N}$ sont prouvables ;
- une condition sur la théorie qui assure que l'énumérabilité implique la représentativité.

C'est l'objet des deux sous-sections qui suivent.

5.2.3 Le système R^-

Le système d'axiomes R^- est le suivant.

Ω_1 : Toutes les formules $\underline{m} + \underline{n} = \underline{k}$ avec $m + n = k$
Ω_2 : Toutes les formules $\underline{m} \times \underline{n} = \underline{k}$ avec $mn = k$
Ω_3 : Toutes les formules $\underline{m} \neq \underline{n}$ avec $m \neq n$
Ω_4 : Toutes les formules $\forall x\ x \leq \underline{n} \Leftrightarrow (x = 0 \vee \ldots \vee x = \underline{n})$

Dans un souci de lisibilité, nous avons conservé les abréviations $(\Leftrightarrow, \vee, \neq, \leq)$.

Proposition 42 *Soit T une théorie arithmétique qui démontre les formules de R^- et φ une formule close Σ_{0^-} alors :*

- *Si $\mathbb{N} \models \varphi$ alors $T \vdash \varphi$.*
- *Si $\mathbb{N} \not\models \varphi$ alors $T \vdash \neg\varphi$.*

Preuve
Dans cette preuve lorsqu'on parle de complétude sémantique, il s'agit de celle de la logique du premier ordre. Démontrons la proposition par induction sur la taille de la formule φ.

Cas $\varphi \equiv \underline{m} = \underline{n}$. Si $\mathbb{N} \models \underline{m} = \underline{n}$, cela signifie que $m = n$. Puisque $\vdash x = x$, par substitution on obtient $\vdash \varphi$. Si $\mathbb{N} \not\models \underline{m} = \underline{n}$, cela signifie que $m \neq n$. Alors $T \vdash \neg\varphi$ qui est une forme de l'axiome Ω_3.

Cas $\varphi \equiv \underline{m} + \underline{n} = \underline{p}$. Si $\mathbb{N} \models \varphi$, cela signifie que φ est une forme de l'axiome Ω_1. Donc $T \vdash \varphi$ Si $\mathbb{N} \not\models \varphi$, cela signifie que $m + n = q \neq p$. Alors $T \vdash \underline{m} + \underline{n} = \underline{q}$ et $T \vdash \underline{q} \neq \underline{p}$. Par complétude sémantique, on obtient $T \vdash \neg\varphi$. Le cas $\underline{m} \times \underline{n} = \underline{p}$ se traite de manière identique avec l'axiome Ω_2.

Cas $\varphi \equiv \underline{m} \leq \underline{n}$. Si $\mathbb{N} \models \underline{m} \leq \underline{n}$, cela signifie que $m \leq n$. Puisque $\vdash \underline{m} = \underline{m}$ et $T \vdash (\underline{m} = 0 \vee \ldots \vee \underline{m} = \underline{n}) \Rightarrow \underline{m} \leq \underline{n}$ (axiome Ω_4 suivi d'une substitution), on obtient par complétude sémantique $T \vdash \varphi$. Si $\mathbb{N} \not\models \underline{m} \leq \underline{n}$, cela signifie que $m > n$. Par conséquent $T \vdash \underline{m} \neq \underline{k}$ pour tout $k \leq n$. A l'aide d'une contraposée de l'axiome Ω_4 suivi d'une substitution, on obtient $T \vdash (\underline{m} \neq 0 \wedge \ldots \wedge \underline{m} \neq \underline{n}) \Rightarrow \neg \underline{m} \leq \underline{n}$. Par complétude sémantique, $T \vdash \neg\varphi$.

Cas $\varphi \equiv \neg\psi$. Si $\mathbb{N} \models \varphi$ alors $\mathbb{N} \not\models \psi$ et par hypothèse de récurrence $T \vdash \varphi$. Si $\mathbb{N} \not\models \varphi$ alors $\mathbb{N} \models \psi$ et par hypothèse de récurrence $T \vdash \psi$. Par complétude sémantique, $T \vdash \neg\neg\psi$.

Cas $\varphi \equiv \psi \Rightarrow \chi$. Si $\mathbb{N} \models \varphi$ alors $\mathbb{N} \not\models \psi$ ou $\mathbb{N} \models \chi$. Par hypothèse de récurrence, $T \vdash \neg\psi$ ou $T \vdash \chi$. Par complétude sémantique, $T \vdash \varphi$. Si $\mathbb{N} \not\models \varphi$ alors $\mathbb{N} \models \psi$ et $\mathbb{N} \not\models \chi$. Par hypothèse de récurrence, $T \vdash \neg\psi$ ou $T \vdash \neg\chi$. Par complétude sémantique, $T \vdash \neg\varphi$.

Cas $\varphi \equiv \forall v_i(v_i \leq \underline{n} \Rightarrow \psi)$. Si $\mathbb{N} \models \varphi$ alors $\mathbb{N} \models \psi(v_i \leftarrow \underline{k})$ pour tout $k \leq n$. Par hypothèse de récurrence $T \vdash \psi(v_i \leftarrow \underline{k})$ pour tout $k \leq n$. Par complétude sémantique, $T \vdash v_i = \underline{k} \Rightarrow \psi$ pour tout $k \leq n$. Par complétude sémantique, $T \vdash (v_i = 0 \vee \ldots \vee v_i = \underline{n}) \Rightarrow \psi$ L'axiome Ω_4 avec v_i substitué à x, nous donne $T \vdash v_i \leq \underline{n} \Rightarrow (v_i = 0 \vee \ldots \vee v_i = \underline{n})$. Par complétude sémantique, $T \vdash (v_i \leq \underline{n} \Rightarrow \psi)$. La règle de généralisation donne alors $T \vdash \varphi$.

Si $\mathbb{N} \not\models \varphi$ alors $\mathbb{N} \not\models \psi(v_i \leftarrow \underline{k})$ pour un certain $k \leq n$. Donc $T \vdash \neg\psi(v_i \leftarrow \underline{k})$ et $T \vdash \underline{k} \leq \underline{n}$ par hypothèse de récurrence. Autrement dit, $T \vdash \neg\psi(v_i \leftarrow \underline{k}) \wedge \underline{k} \leq \underline{n}$. Par complétude sémantique, $T \vdash \exists v_i \neg\psi \wedge v_i \leq \underline{n}$ ce qui permet de conclure, de nouveau par complétude sémantique, que $T \vdash \neg\varphi$.

c.q.f.d. $\Diamond\Diamond\Diamond$

5.2.4 L'ω-cohérence

L'ω-cohérence est une propriété qu'on peut raisonnablement attendre d'une théorie axiomatisant l'arithmétique.

Définition 55 *Soit T une théorie relative à l'arithmétique, T est dite ω-cohérente si pour toute formule φ avec au plus v_i comme variable libre, on a :*

Si $T \vdash \exists v_i\ \varphi$ alors $\exists n\ \in \mathbb{N}$ tel que $T \nvdash \neg\varphi(v_i \leftarrow \underline{n})$

Pour justifier cette définition, nous établissons la proposition suivante.

Proposition 43 *Soit T une théorie relative à l'arithmétique qui admet $\mathbb{N}$ pour modèle, alors T est ω-cohérente.*

Preuve
Si $\exists v_i\ \varphi$ est un théorème de T alors $\mathbb{N} \models \exists v_i\ \varphi$, donc il existe n tel que $\mathbb{N} \models \varphi(v_i \leftarrow \underline{n})$. Par conséquent, $T \nvdash \neg\varphi(v_i \leftarrow \underline{n})$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Il se trouve que cette propriété est suffisante pour nos objectifs.

Proposition 44 *Soit T une théorie ω-cohérente et soit R une relation n-aire énumérable dans T alors R est représentable dans T.*

Preuve
Soit φ une formule qui énumère R dans T.

Supposons que $R(k_1, \ldots, k_n)$ soit vérifiée. Donc $\exists k_{n+1}$ tel que $T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$. Par complétude sémantique $T \vdash \exists v_{n+1} \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n})$.

Supposons que $R(k_1, \ldots, k_n)$ ne soit pas vérifiée. Donc pour tout $k_{n+1} \in \mathbb{N}$, $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n+1})$. Par ω-cohérence, $T \nvdash \exists v_{n+1} \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \leq n})$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Théorème 17 (1er Théorème d'incomplétude) *Soit T une théorie arithmétique récursivement énumérable ω-cohérente telle que les axiomes de R^- soient des théorèmes de T. Alors T n'est pas syntaxiquement complet.*

Preuve
Puisque les axiomes de R^- sont des théorèmes de T, toute formule Σ_{0^-} satisfaite dans $\mathbb{N}$ est démontrable dans T (proposition 42). Puisque toute formule Σ_{0^-} satisfaite dans $\mathbb{N}$ est démontrable dans T, toute relation Σ_1 est énumérable dans T (lemme 40). Puisque T est ω-cohérente, toute relation Σ_1 est représentable dans T (proposition 44). En particulier $(\mathcal{TH}_T)^d$ est représentable dans T (lemme 39). Donc T est syntaxiquement incomplet (proposition 41).

c.q.f.d. $\Diamond\Diamond\Diamond$

Cette version du théorème implique la première version du théorème. En effet, supposons qu'il existe une théorie syntaxiquement complète T qui admette $\mathbb{N}$ pour modèle. Alors puisque $\mathbb{N}$ satisfait les axiomes de R^-, ce sont des théorèmes de T. D'autre part, d'après la proposition 43, T est ω-cohérente. Donc T est syntaxiquement incomplet !

L'intérêt de la première version réside surtout dans le résultat intermédiaire de la non arithmicité des formules arithmétiques vraies.

5.3 Le premier théorème de Gödel(-Rosser)

L'hypothèse de l'ω-cohérence du théorème semble introduite artificiellement pour parvenir au résultat. Nous allons maintenant nous en débarasser au prix d'un renforcement des axiomes de R^-.

5.3.1 Séparabilité

Nous établissons une proposition qui est une amélioration de la proposition 40 de la section précédente.

Notation. $\mathcal{RF}_T$ désignera l'ensemble des numéros de Gödel des formules réfutables dans T.

Proposition 45 *Soit T une théorie et supposons que E_n représente un ensemble A tel que $(\mathcal{RF}_T)^d \subseteq A$ et $A \cap (\mathcal{TH}_T)^d = \emptyset$. Alors $E_n[\underline{n}]$ (et par conséquent $E_n(x \leftarrow \underline{n})$) n'est ni prouvable, ni réfutable dans T.*

Preuve
Supposons que $E_n[\underline{n}]$ soit réfutable dans T, *i.e.* $n \in (\mathcal{RF}_T)^d$. alors puisque $(\mathcal{RF}_T)^d \subseteq A$ et que E_n représente A, $E_n[\underline{n}]$ est prouvable, *i.e.* $n \in (\mathcal{TH}_T)^d$. Ce qui est impossible puisque $(\mathcal{RF}_T)^d \cap (\mathcal{TH}_T)^d = \emptyset$.

Supposons que $E_n[\underline{n}]$ soit prouvable dans T , *i.e.* $n \in (\mathcal{TH}_T)^d$. alors puisque $A \cap (\mathcal{TH}_T)^d = \emptyset$, $n \notin A$. Puisque E_n représente A, $E_n[\underline{n}]$ n'est pas prouvable, à nouveau impossible.

c.q.f.d. $\Diamond\Diamond\Diamond$

Ceci nous conduit à la définition suivante.

Définition 56 *Soient T une théorie et φ une formule dont l'ensemble des variables libres est contenu dans $\{v_1, \ldots, v_n\}$,* φ sépare R *et R' deux relations n-aires de $\mathbb{N}$ dans T si :*
- *Si $R(k_1, \ldots, k_n)$ alors $T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i \in 1..n})$.*
- *Si $R'(k_1, \ldots, k_n)$ alors $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i \in 1..n})$.*

Soient T une théorie et φ une formule avec au plus v_1 comme variable libre φ sépare $A, B \subseteq \mathbb{N}$ deux ensembles dans T si :
- *Si $k \in A$ alors $T \vdash \varphi(v_1 \leftarrow \underline{k})$.*
- *Si $k \in B$ alors $T \vdash \neg\varphi(v_1 \leftarrow \underline{k})$.*

Dans la proposition 45, la cohérence était garantie par les hypothèses *(prouvez-le)*. Nous la réintroduisons maintenant avec cette variante.

Proposition 46 *Soit T une théorie cohérente et supposons que E_n sépare $(\mathcal{RF}_T)^d$ et $(\mathcal{TH}_T)^d$. Alors $E_n[\underline{n}]$ (et par conséquent $E_n(x \leftarrow \underline{n})$) n'est ni prouvable, ni réfutable dans T.*

Preuve
Soit A, l'ensemble représenté par E_n. Par définition de la séparation, $(\mathcal{RF}_T)^d \subseteq A$. S'il existe $k \in A \cap (\mathcal{TH}_T)^d$ alors $E_n(v_1 \leftarrow \underline{k})$ est à la fois prouvable ($k \in A$) et réfutable ($k \in (\mathcal{TH}_T)^d$) contrairement à la cohérence. Donc les hypothèses de la proposition 45 sont satisfaites. Ce qui permet de conclure.

c.q.f.d. $\Diamond\Diamond\Diamond$

5.3.2 Le système R

Le système d'axiomes R (une extension de R^-) est le suivant.
- Ω_1 : Toutes les formules $\underline{m} + \underline{n} = \underline{k}$ avec $m + n = k$
- Ω_2 : Toutes les formules $\underline{m} \times \underline{n} = \underline{k}$ avec $mn = k$
- Ω_3 : Toutes les formules $\underline{m} \neq \underline{n}$ avec $m \neq n$
- Ω_4 : Toutes les formules $\forall x\ x \leq \underline{n} \Leftrightarrow (x = 0 \vee \ldots \vee x = \underline{n})$
- Ω_5 : Toutes les formules $\forall x\ x \leq \underline{n} \vee \underline{n} \leq x$

Proposition 47 *Soient T une théorie telle que les axiomes de R soient des théorèmes de T, P et P' deux relations énumérables, alors $P \wedge \neg P'$ et $P' \wedge \neg P$ sont séparables.*

Preuve
Soient φ et ψ les formules qui énumérent P et P'.
On suppose que v_{n+2} n'apparaît pas dans φ et dans ψ.

Supposons $P(k_1,\ldots,k_n) \wedge \neg P'(k_1,\ldots,k_n)$.
Puisque $P(k_1,\ldots,k_n)$, il existe k_{n+1} tel que $T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n+1})$.
D'autre part $\vdash v_{n+2} \leq \underline{k_{n+1}} \vee \neg v_{n+2} \leq \underline{k_{n+1}}$
A l'aide de l'axiome Ω_4, on en déduit
$T \vdash v_{n+2} = 0 \vee \ldots \vee v_{n+2} = \underline{k_{n+1}} \vee \neg v_{n+2} \leq \underline{k_{n+1}}$
Puisque $\neg P'(k_1,\ldots,k_n)$, pour tout $k \leq k_{n+1}$,
$T \vdash v_{n+2} = \underline{k} \Rightarrow \neg\psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\})$
En combinant (propositionnellement),
$T \vdash \neg\psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}) \vee \neg v_{n+2} \leq \underline{k_{n+1}}$
Par généralisation,
$T \vdash \forall v_{n+2}\ \neg\psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}) \vee \neg v_{n+2} \leq \underline{k_{n+1}}$
En incluant la première déduction,
$T \vdash \varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n+1}) \wedge$
$\forall v_{n+2}\ (\neg\psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}) \vee \neg v_{n+2} \leq \underline{k_{n+1}})$
Par complétude sémantique,
$T \vdash \exists v_{n+1}\ \varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}) \wedge$
$\forall v_{n+2}\ (\neg\psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}) \vee \neg v_{n+2} \leq v_{n+1})$

Posons $\theta \equiv \exists v_{n+1}\varphi \wedge \forall v_{n+2}\ (\neg\psi(\{v_{n+1} \leftarrow v_{n+2}\}) \vee \neg v_{n+2} \leq v_{n+1})$.
Nous venons de prouver que $T \vdash \theta(\{v_i \leftarrow \underline{k_i}\}_{i\leq n})$.
Notons que θ s'écrit aussi :

$$\boxed{\exists v_{n+1}\varphi \wedge \forall v_{n+2}\ (v_{n+2} \leq v_{n+1} \Rightarrow \neg\psi(\{v_{n+1} \leftarrow v_{n+2}\}))}$$

Remarquons que $\vdash \neg\theta \Leftrightarrow \forall v_{n+1}\neg\varphi \vee \exists v_{n+2}\ (\psi(\{v_{n+1} \leftarrow v_{n+2}\}) \wedge v_{n+2} \leq v_{n+1})$
Avec l'hypothèse sur v_{n+2},
$\vdash \neg\theta \Leftrightarrow \forall v_{n+1}\exists v_{n+2}\ \neg\varphi \vee (\psi(\{v_{n+1} \leftarrow v_{n+2}\}) \wedge v_{n+2} \leq v_{n+1})$
Ou encore,
$\vdash \neg\theta \Leftrightarrow \forall v_{n+1}\exists v_{n+2}\ (\neg\varphi \vee \psi(\{v_{n+1} \leftarrow v_{n+2}\})) \wedge (\neg\varphi \vee v_{n+2} \leq v_{n+1})$
Supposons $P'(k_1,\ldots,k_n) \wedge \neg P(k_1,\ldots,k_n)$.
Puisque $P'(k_1,\ldots,k_n)$, il existe k_{n+1} tel que $T \vdash \psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n+1})$.
D'où : $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}) \vee \psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n+1})$.
Notez que v_{n+1} n'a pas été substitué dans φ.

Puisque $\neg P(k_1,\ldots,k_n)$, pour tout k
$T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow \underline{k}\})$.
Par conséquent, pour tout $k \leq k_{n+1}$,
on a $T \vdash v_{n+1} = \underline{k} \Rightarrow \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n})$.
En combinant,
$T \vdash (v_{n+1} = \underline{0} \vee \ldots \vee v_{n+1} = \underline{k_{n+1}}) \Rightarrow \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n})$.
A l'aide de Ω_4, on obtient
$T \vdash v_{n+1} \leq \underline{k_{n+1}} \Rightarrow \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n})$.
Autrement dit, $T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}\}) \vee \neg v_{n+1} \leq \underline{k_{n+1}}$.
En utilisant l'axiome Ω_5, on obtient
$T \vdash \neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}\}) \vee \underline{k_{n+1}} \leq v_{n+1}$.
En combinant avec une déduction précédente,
$T \vdash (\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}) \vee \psi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n+1})) \wedge$
$(\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}\}) \vee \underline{k_{n+1}} \leq v_{n+1})$.

Par complétude sémantique,
$T \vdash \exists v_{n+2}(\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}) \vee \psi(\{\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}\}))\wedge$
$(\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}\}) \vee \underline{v_{n+2}} \leq v_{n+1})$.
Par généralisation,
$T \vdash \forall v_{n+1}\exists v_{n+2}(\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}) \vee \psi(\{\{v_i \leftarrow \underline{k_i}\}_{i\leq n} \cup \{v_{n+1} \leftarrow v_{n+2}\}\}))\wedge$
$(\neg\varphi(\{v_i \leftarrow \underline{k_i}\}_{i\leq n}\}) \vee \underline{v_{n+2}} \leq v_{n+1})$.
Autrement dit, $T \vdash \neg\theta(\{v_i \leftarrow \underline{k_i}\}_{i\leq n})$.

c.q.f.d. ◊◊◊

Théorème 18 (1er Théorème d'incomplétude - version Rosser)
Soit T une théorie arithmétique récursivement énumérable cohérente telle que les axiomes de R soient des théorèmes de T. Alors T n'est pas syntaxiquement complet.

Preuve
Puisque $(\mathcal{RF}_T)^d$ et $(\mathcal{TH}_T)^d$ sont des ensembles Σ_1, et que T démontre R^-, ils sont énumérables dans T (proposition 42). Puisque T est cohérente, ces ensembles sont disjoints. Puisque T démontre R, ils sont séparables dans T (proposition 47). Les hypothèses de la proposition 46 sont donc satisfaites et T est syntaxiquement incomplet.

c.q.f.d. ◊◊◊

5.4 Le second théorème de Gödel

L'objectif du second théorème de Gödel est de trouver une formule significative qu'une théorie n'arrive ni à démontrer. D'une manière plus précise, la formule à démontrer est l'une des formules représentant la cohérence et pourrait s'exprimer informellement ainsi : « T ne peut démontrer qu'il n'y a pas de démonstration dans T de $\neg 0 = 0$ ». Au vu de la preuve, on peut substituer n'importe quelle formule réfutable par T à la formule $\neg 0 = 0$.

Lemme 41 *Soit T une théorie arithmétique qui démontre R et $y = f(x)$ une fonction qui est une relation Σ_{0^-}. Alors il existe une formule φ dont x et y sont les uniques variables libres telle que pour tout m, $T \vdash \forall y\, \varphi(x \leftarrow \underline{m}) \Leftrightarrow y = \underline{f(m)}$.*

Preuve
Puisque T démontre R^-, nous savons qu'il existe une formule φ' telle que pour toute paire d'entiers m, n :
Si $n = f(m)$ alors $T \vdash \varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$ sinon $T \vdash \neg\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$
Sans perte de généralité, on suppose que z n'apparaît pas dans φ'.
Posons $\varphi \equiv \varphi' \wedge \forall z\ (z < y \Rightarrow \neg\varphi'(y \leftarrow z))$.
$T \vdash y = \underline{f(m)} \Rightarrow \varphi(x \leftarrow \underline{m})$ ssi $T \vdash \varphi(\{x \leftarrow \underline{m}, y \leftarrow \underline{f(m)}\})$.
Posons $n = f(m)$. Démontrons $T \vdash \varphi(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$.
Par définition de φ', on a :
$T \vdash \neg\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{0}\}) \wedge \ldots \wedge \neg\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n-1}\})$
$\wedge\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$
Puisque $R^- \vdash x < \underline{n} \Rightarrow x = \underline{0} \vee x = \underline{1} \ldots \vee x = \underline{n-1}$, $T \vdash \varphi(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$.
Démontrons maintenant $T \vdash \varphi(x \leftarrow \underline{m}) \Rightarrow y = \underline{f(m)}$.
Rappelons que pour tout n, $R \vdash y = \underline{0} \vee y = \underline{1} \ldots \vee y = \underline{n} \vee y > \underline{n}$.

$T \vdash \varphi \Rightarrow ((\varphi \wedge y = \underline{0}) \vee (\varphi \wedge y = \underline{1}) \ldots \vee (\varphi \wedge y = \underline{n}) \vee (\varphi \wedge y > \underline{n}))$
Puisque $T \vdash \neg\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{0}\}) \wedge \ldots \wedge \neg\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n-1}\})$
$\wedge\varphi'(\{x \leftarrow \underline{m}, y \leftarrow \underline{n}\})$
$T \vdash \neg(\varphi \wedge y = \underline{0})$, $T \vdash \neg(\varphi \wedge y = \underline{1})$, $\ldots$, $T \vdash \neg(\varphi \wedge y > \underline{n})$.
Par conséquent, $T \vdash \varphi \Rightarrow y = \underline{n}$ et on conclut par généralisation.

c.q.f.d. $\Diamond\Diamond\Diamond$

Notation. Dans cette section, on note $\psi\{\varphi\} \equiv \psi(x \leftarrow \underline{g(\varphi)})$.

Lemme 42 *Soit T une théorie arithmétique qui démontre R et ψ une formule dont x est l'unique variable libre. Alors il existe une formule close φ telle que $T \vdash \varphi \Leftrightarrow \psi\{\varphi\}$.*

Preuve
Rappelons que la fonction diagonale $y = d(x)$ est une relation Σ_{0^-}. En vertu du lemme 41, il existe une formule θ_d telle que pour tout m :

$$T \vdash \forall y\ \theta_d(x \leftarrow \underline{m}) \Leftrightarrow y = \underline{d(m)}$$

Soit v_i une variable n'apparaissant ni dans ψ ni dans θ_d.
Posons $\psi' \equiv \forall v_i\ (\theta_d(y \leftarrow v_i) \Rightarrow \psi(x \leftarrow v_i))$ et $\varphi \equiv \forall x\ (x = \underline{g(\psi')} \Rightarrow \psi')$.
Remarquons que $g(\varphi) = d(g(\psi'))$.
Par conséquent, $T \vdash \theta_d(y \leftarrow \underline{g(\varphi)})(x \leftarrow \underline{g(\psi')})$.

Par instanciation,
$\{\varphi\} \vdash \underline{g(\psi')} = \underline{g(\psi')} \Rightarrow \forall v_i\ (\theta_d(y \leftarrow v_i)(x \leftarrow \underline{g(\psi')}) \Rightarrow \psi(x \leftarrow v_i))$
Soit $\{\varphi\} \vdash \forall v_i\ (\theta_d(y \leftarrow v_i)(x \leftarrow \underline{g(\psi')}) \Rightarrow \psi(x \leftarrow v_i))$
Par instanciation, $\{\varphi\} \vdash \theta_d(y \leftarrow \underline{g(\varphi)})(x \leftarrow \underline{g(\psi')}) \Rightarrow \psi(x \leftarrow \underline{g(\varphi)})$
D'où par modus ponens, $T \cup \{\varphi\} \vdash \psi\{\varphi\}$
A l'aide du lemme de déduction, on obtient : $T \vdash \varphi \Rightarrow \psi\{\varphi\}$.

D'autre part,
$\{\neg\varphi\} \vdash \exists x\ x = \underline{g(\psi')} \wedge \neg\psi'$
$\{\neg\varphi\} \vdash \neg\psi'(x \leftarrow \underline{g(\psi')})$
$\{\neg\varphi\} \vdash \exists v_i\ \theta_d(y \leftarrow v_i) \wedge \neg\psi(x \leftarrow v_i))$
Or $T \vdash \forall y\ \theta_d(x \leftarrow \underline{g(\psi')}) \Rightarrow y = \underline{g(\varphi)}$
Par conséquent, $T \cup \{\neg\varphi\} \vdash \neg\psi\{\varphi\}$
A l'aide du lemme de déduction, on obtient : $T \vdash \neg\varphi \Rightarrow \neg\psi\{\varphi\}$.
Ce qui permet de conclure.

c.q.f.d. $\Diamond\Diamond\Diamond$

La définition qui suit est au coeur de la preuve du second théorème d'incomplétude de Gödel. On a déjà (implicitement) établi que pour une théorie ω-consistante qui démontre R^-, il existe une formule arithmétique satisfaisant le point a de la définition ci-dessous.

Définition 57 *Soit T une théorie arithmétique et soit* `Pr` *une formule arithmétique, dont x est l'unique variable libre.* `Pr` *est dite un prédicat de prouvabilité pour T si les trois conditions suivantes sont réunies.*

a. Pour toute formule close φ, $T \vdash \varphi$ ssi $T \vdash \mathtt{Pr}\{\varphi\}$
b. Pour toutes formules closes φ, ψ, $T \vdash \mathtt{Pr}\{\varphi \Rightarrow \psi\} \Rightarrow (\mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\psi\})$
c. Pour toute formule close φ, $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\mathtt{Pr}\{\varphi\}\}$

Théorème 19 (2ème Théorème d'incomplétude - version abstraite)
Soit T une théorie arithmétique cohérente qui démontre R et qui possède un prédicat de prouvabilité $\mathtt{Pr}$*, alors* $T \nvdash \neg\mathtt{Pr}\{\neg 0 = 0\}$.

Preuve
A l'aide des points a et b, on obtient que si $T \vdash \varphi \Rightarrow \psi$ alors $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\psi\}$
Nous désignerons cette assertion comme étant le point d.
Dans la suite, nous notons φ, la formule du lemme 42 associée à $\neg\mathtt{Pr}$.

1. $T \vdash \varphi \Leftrightarrow \neg\mathtt{Pr}\{\varphi\}$ *(par définition de φ)*
2. $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\neg\mathtt{Pr}\{\varphi\}\}$ *(à l'aide de 1 et d)*
3. $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\mathtt{Pr}\{\varphi\}\}$ *(c)*
4. $T \vdash \neg\mathtt{Pr}\{\varphi\} \Rightarrow (\mathtt{Pr}\{\varphi\} \Rightarrow \neg 0 = 0)$ *(tautologie propositionnelle)*
5. $T \vdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\varphi\}\} \Rightarrow \mathtt{Pr}\{\mathtt{Pr}\{\varphi\} \Rightarrow \neg 0 = 0\}$ *(à l'aide de 4 et d)*
6. $T \vdash \mathtt{Pr}\{\mathtt{Pr}\{\varphi\} \Rightarrow \neg 0 = 0\} \Rightarrow (\mathtt{Pr}\{\mathtt{Pr}\{\varphi\}\} \Rightarrow \mathtt{Pr}\{\neg 0 = 0\})$ *(b)*
7. $T \vdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\varphi\}\} \Rightarrow (\mathtt{Pr}\{\mathtt{Pr}\{\varphi\}\} \Rightarrow \mathtt{Pr}\{\neg 0 = 0\})$ *(à l'aide de 5 et 6)*
8. $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow (\mathtt{Pr}\{\mathtt{Pr}\{\varphi\}\} \Rightarrow \mathtt{Pr}\{\neg 0 = 0\})$ *(à l'aide de 2 et 7)*
9. $T \vdash \mathtt{Pr}\{\varphi\} \Rightarrow \mathtt{Pr}\{\neg 0 = 0\}$ *(à l'aide de 3 et 8)*
10. $T \vdash \neg\mathtt{Pr}\{\neg 0 = 0\} \Rightarrow \neg\mathtt{Pr}\{\varphi\}$ *(contraposée de 9)*
11. $T \vdash \neg\mathtt{Pr}\{\neg 0 = 0\} \Rightarrow \varphi$ *(à l'aide de 1 et 10)*
12. $T \vdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\neg 0 = 0\}\} \Rightarrow \mathtt{Pr}\{\varphi\}$ *(à l'aide de 11 et d)*
13. $T \vdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\neg 0 = 0\}\} \Rightarrow \mathtt{Pr}\{\neg 0 = 0\}$ *(à l'aide de 9 et 12)*

Supposons que $T \vdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\neg 0 = 0\}\}$ alors, par modus ponens, $T \vdash \mathtt{Pr}\{\neg 0 = 0\}$ et par conséquent $T \vdash \neg 0 = 0$ (à l'aide de a) contrairement à l'hypothèse de cohérence. D'où $T \nvdash \mathtt{Pr}\{\neg\mathtt{Pr}\{\neg 0 = 0\}\}$ ce qui entraine (à l'aide de a) que $T \nvdash \neg\mathtt{Pr}\{\neg 0 = 0\}$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Pour parvenir à une version concrète de ce théorème, il faut établir quel type de théorie garantit l'existence d'un prédicat de prouvabilité. C'est ce qu'Hilbert et Bernays ont étudié en détail dans le tome 2 de leur livre « Fondements des mathématiques ». *A suivre ...*

5.5 TD n°5

Question n°1. Démontrer la proposition 37.

Question n°2. Démontrer la proposition 38.

Question n°3. Démontrer la proposition 39.

Chapitre 6

Logique du second ordre

6.1 Syntaxe et sémantique

La logique du second ordre est syntaxique très proche de la logique du premier ordre. On adjoint d'abord au support des variables de fonctions et de prédicats.

Définition 58 *Un support* $Supp = \langle Var, Cst, \{Fct_i\}_{i>0}, \{Pred_i\}_{i\geq 0}\rangle$ *d'un calcul de prédicats est défini par :*

- Var*, un ensemble dénombrable de variables.*
- Cst*, un ensemble fini ou dénombrable de constantes.*
- $\{Fct_i\}_{i>0}$*, une famille d'ensembles (disjoints) finis ou dénombrables de fonctions.* Fct_i *désigne l'ensemble des fonctions d'arité* i*. On note* $Fct = \biguplus_{i>0} Fct_i$*.*
- $\{Pred_i\}_{i\geq 0}$*, une famille d'ensembles (disjoints) finis ou dénombrables de prédicats.* $Pred_i$ *désigne l'ensemble des prédicats d'arité* i*. On note* $Pred = \biguplus_{i\geq 0} Pred_i$*.*
- $\{VarFct_i\}_{i>0}$*, une famille d'ensembles (disjoints) dénombrables de variables de fonctions.* $VarFct_i$ *désigne l'ensemble des variables de fonctions d'arité* i*. On note* $VarFct = \biguplus_{i>0} VarFct_i$*.*
- $\{VarPred_i\}_{i\geq 0}$*, une famille d'ensembles (disjoints) dénombrables de variables de prédicats.* $VarPred_i$ *désigne l'ensemble des prédicats d'arité* i*. On note* $VarPred = \biguplus_{i\geq 0} VarPred_i$*.*

Tous les ensembles de la définition précédente sont supposés disjoints.

A l'aide de ces éléments, on définit d'abord des termes.

Définition 59 *Soit un support Supp, un terme est défini inductivement comme suit :*

- *Une variable ou une constante est un terme.*
- *Soit* $f \in Fct_i \cup VarFct_i$*,* $t_1, \ldots, t_i$ *des termes alors* $f(t_1, \ldots, t_i)$ *est un terme.*

A partir des termes, on définit les atomes.

Définition 60 *Un atome est défini comme suit :*
Soit $p \in Pred_i \cup VarPred_i$*,* $t_1, \ldots, t_i$ *des termes alors* $p(t_1, \ldots, t_i)$ *est un atome.*

Nous sommes maintenant prêts à définir les formules de la logique du second ordre.

Définition 61 *L'ensemble des formules F_2 de la logique du second ordre associée à un support Supp est défini inductivement comme suit :*
- *Un atome de Supp est une formule de F_2.*
- *Si $A, B \in F_2$ et $x \in Var \cup VarFct \cup VarPred$ alors $\neg A \in F_2$, $A \Rightarrow B \in F_2$, $\forall x A \in F_1$ et $\exists x A \in F_2$.*

Notations. Soit E un ensemble. Dans la suite $\overline{e}$ désignera une famille indexée par $Var \cup VarFct \cup VarPred$ t.q.
- Si $x \in Var$ alors $e_x \in E$
- Si $x \in VarFct_i$ alors $e_x \in E^i \mapsto E$
- Si $x \in VarPred_i$ alors $e_x \in E^i \mapsto \{\mathtt{V}, \mathtt{F}\}$

$\overline{e}[e', x]$ désigne la famille obtenue à partir de $\overline{e}$ en substituant à e' à e_x.

Définition 62 *Une interprétation ι associée à un support Supp est définie par :*
- *Un ensemble non vide E_ι.*
- *Pour chaque constante $c \in Cst$, un élement $c^\iota \in E_\iota$;*
- *Pour chaque $f \in Fct_i$, une fonction f^ι de E_ι^i dans E_ι.*
- *Pour chaque $p \in Pred_i$, une fonction p^ι de E_ι^i dans $\{\mathtt{V}, \mathtt{F}\}$.*

L'interprétation d'un terme t est une fonction t^ι de E_ι^{Var} dans E_ι où $t^\iota(\overline{e})$ est défini inductivement par :
- *Si $t = c \in Cst$ alors $t^\iota(\overline{e}) = c^\iota$*
- *Si $t = x \in Var$ alors $t^\iota(\overline{e}) = e_x$*
- *Si $t = f(t_1, \ldots, t_n)$ avec $f \in Fct_n$ alors $t^\iota(\overline{e}) = f^\iota(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$*
- *Si $t = f(t_1, \ldots, t_n)$ avec $f \in VarFct_n$ alors $t^\iota(\overline{e}) = e_f(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$*

L'interprétation d'un atome $p(t_1, \ldots, t_i)$ est une fonction $p(t_1, \ldots, t_i)^\iota$ de E_ι^{Var} dans $\{\mathtt{V}, \mathtt{F}\}$ définie par :
- *Si $p \in Pred_n$ alors $p(t_1, \ldots, t_n)^\iota(\overline{e}) = p^\iota(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$*
- *Si $p \in VarPred_n$ alors $p(t_1, \ldots, t_n)^\iota(\overline{e}) = e_p(t_1^\iota(\overline{e}), \ldots, t_n^\iota(\overline{e}))$*

L'interprétation d'une formule φ est alors définie inductivement comme une fonction φ^ι de E_ι^{Var} dans $\{\mathtt{V}, \mathtt{F}\}$ ainsi :

cas $\varphi = \neg\psi$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\psi^\iota(\overline{e}) = \mathtt{F}$.

cas $\varphi = \psi_1 \Rightarrow \psi_2$ **:** $\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi* $\psi_1^\iota(\overline{e}) = \mathtt{F}$ *ou* $\psi_2^\iota(\overline{e}) = \mathtt{V}$.

cas $\varphi = \forall x \psi$ **:** *avec $x \in Var$ (resp. $x \in VarFct_i$, $x \in VarFct_i$)*
$\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi*
$\forall e' \in E_\iota$ *(resp. $e' \in E_\iota^i \mapsto E_\iota$, $e' \in E_\iota^i \mapsto \{\mathtt{V}, \mathtt{F}\}$)* $\psi^\iota(\overline{e}[e', x]) = \mathtt{V}$.

cas $\varphi = \exists x \psi$ **:** *avec $x \in Var$ (resp. $x \in VarFct_i$, $x \in VarFct_i$)*
$\varphi^\iota(\overline{e}) = \mathtt{V}$ *ssi*
$\exists e' \in E_\iota$ *(resp. $e' \in E_\iota^i \mapsto E_\iota$, $e' \in E_\iota^i \mapsto \{\mathtt{V}, \mathtt{F}\}$)* $\psi^\iota(\overline{e}[e', x]) = \mathtt{V}$.

6.2 Résultats négatifs

Soit la formule $\varphi \in F_2$ définie par :

$$\varphi \equiv \exists R\ (\forall x\ (\neg R(x,x) \wedge (\exists y\ R(x,y)) \wedge (\forall y\ \forall z\ (R(x,y) \wedge R(y,z)) \Rightarrow R(x,z))))$$

où $x, y, z \in Var, R \in VarPred_2$.
Le support associé à cette formule ne contient ni fonction ni prédicat. Il s'agit donc d'une formule qui raisonne sur les ensembles. Elle affirme que ι est un modèle ssi il existe une relation transitive irréflexive sur E_ι t.q. tout élément est un composant gauche d'une paire de la relation.

Lemme 43 *Une interprétation ι est un modèle de φ ssi E_ι est infini.*

La proposition suivante établit le premier résultat négatif : la logique du second ordre n'est pas compacte.

Proposition 48 *Il existe un ensemble dénombrable de formules non simultanément satisfaisables t.q. tout sous-ensemble fini est satisfaisable.*

Reprenons le support de l'arithmétique étudié dans le chapitre consacré aux théorèmes de Gödel. Soit les formules :
$T1 : \forall P\ (P(\underline{0}) \wedge \forall x\ P(x) \Rightarrow P(x')) \Rightarrow \forall x\ P(x)$
$T2 : \forall x\ x' \neq 0$
$T3 : \forall x\ \forall y\ x' = y' \Rightarrow x = y$

Proposition 49 *Une interprétation ι satisfait $T1, T2, T3$ ssi ι est isomorphe à $(\mathbb{N}, 0, ')$.*

La proposition suivante établit le deuxième résultat négatif : l'incomplétude sémantique.

Proposition 50 *La logique du second ordre n'admet pas de système d'axiomes et de règles de déduction récursivement énumérables qui soit sémantiquement complet.*

6.3 Logique et langage

La logique du second ordre se caractérise par une grande expressivité et l'absence de système de déduction sémantiquement complet. Elle est donc intéressante à étudier dans le cadre de théories particulières. Nous illustrons ce point vis à vis des langages formels.

Un mot est une suite finie de lettres sur un alphabet fini disons A. Nous expliquons comment restreindre les interprétations d'une logique de second ordre aux mots finis. Nous nous dotons d'un prédicat binaire S représentant le fait que deux positions se succèdent et de $|A|$ prédicats unaire $\{Q_a\}_{a \in A}$ indiquant qu'en une position donnée la lettre a apparaît. Nous excluons le cas du mot de longueur nulle qui présente des complications techniques sans intérêt du point de vue théorique

Proposition 51 *Il existe un ensemble fini de formules de la logique du second ordre dont les seuls modèles sont les mots finis (de longueur non nulle).*

Dans la suite, les modèles considérés sont donc les mots finis et on parle du langage reconnu par une formule (close) φ. On introduit les abréviations. $first(x) \equiv \forall y\ \neg S(y, x)$ et $last(x) \equiv \forall y\ \neg S(x, y)$.

Proposition 52 *Pour tout automate fini $\mathcal{A} = (Q, A, q_0, \delta, F)$, on peut construire une formule φ t.q. l'ensemble des mots de longueur non nulle reconnus par $\mathcal{A}$ soit $L(\varphi)$.*

Lorsque les formules ont pour variables de prédicats uniquement des prédicats unaires. On appelle ce fragment MSO (« Monadic Second Order Logic »). Notre objectif est de démontrer que les langages des formules MSO sont rationnels. On aura ainsi obtenu une caractérisation logique des langages rationnels. Avant d'établir la preuve, nous transformons légèrement les formules afin de masquer les variables de premier ordre dans des abréviations. Plus précisément nous procédons comme suit.

- La première abréviation indique qu'un (ou une variable de) prédicat unaire est vrai en une unique position.
  $Sing(X) \equiv \exists x\ X(x) \wedge \forall x\ \forall y\ (X(x) \wedge X(y)) \Rightarrow x = y$
- La deuxième abréviation indique qu'un (ou une variable de) prédicat unaire désigne un sous-ensemble de positions d'un autre prédicat unaire.
  $X \subseteq Y \equiv \forall x\ X(x) \Rightarrow Y(x)$
- La troisième abréviation indique que deux prédicats unaires sont vrais en une unique position, l'une étant le successeur de l'autre.
  $Succ(X, Y) \equiv Sing(X) \wedge Sing(Y) \wedge \exists x\ \exists y\ X(x) \wedge Y(y) \wedge S(x, y)$

Toute formule MSO peut être alors transformée pour ne plus faire apparaître que des variables de second ordre. Nous en donnons un exemple qui devrait convaincre le lecteur de la simplicité de la transformation. La formule :

$$\varphi \equiv \forall x\ Q_a(x) \Rightarrow (\exists y\ S(x, y) \wedge Z(y))$$

devient :

$$\varphi' \equiv \forall X\ Sing(X) \Rightarrow (X \subseteq Q_a \Rightarrow (\exists Y\ Sing(Y) \wedge (Succ(X, Y) \wedge Y \subseteq Z)))$$

Nous appellerons ce fragment MSO_0.

Nous voulons démontrer le résultat par induction sur la taille de la formule. Cependant ceci nous conduit à nous interroger sur la signification de la satisfaction d'une formule qui comporte des variables libres. Dans le cas présent, il s'agit uniquement de variables de prédicat unaires. Autrement, la satisfaction dépend de la valeur attribuée à ces prédicats unaires. Puisqu'on raisonne sur des mots, la valuation de chaque prédicat peut être interprétée comme un booléen étiquetant chaque position. Autrement dit, soit φ une formule dont $X_1, \ldots, X_k$ sont les variables libres, soit $w = w_0 \ldots w_n$ un mot et soit $P_1, \ldots, P_k$ une valuation de $X_1, \ldots, X_k$. Alors $\varphi^w[\{X_i \leftarrow P_i\}_i] = \mathtt{V}$ ssi $\varphi^{w'} = \mathtt{V}$ où $w' = w'_0 \ldots w'_n$ est construit sur l'alphabet $A \times \{\mathtt{V}, \mathtt{F}\}^k$ par $w'_p = (w_p, P_1(p), \ldots, P_k(p))$ et où $Q_a(p)^{w'}$ est vrai si la première composante de w'_p est a et $X_i(p)$ est vrai si si la $i+1$ème composante de w'_p est $\mathtt{V}$. On vient donc d'établir une correspondance biunivoque entre les mots de A^+ dotés d'une valuation pour $X_1, \ldots, X_k$ et les mots de $(A \times \{\mathtt{V}, \mathtt{F}\}^k)^+$. On peut donc parler du langage d'une formule MSO_0 quelconque, la nouvelle notion coïncidant avec l'ancienne dans le cas des formules closes.

Proposition 53 *Pour toute formule $\varphi \in MSO_0$, on peut construire un automate fini qui reconnaisse $L(\varphi)$.*

En combinant les deux théorèmes, on en déduit que toute formule MSO sur les mots finis est équivalente à une formule $EMSO$ c'est à dire une formule où les variables du second ordre sont quantifiées existentiellement.

6.4 TD n°6

Question n°1. Démontrer le lemme 43.

Question n°2. Démontrer la proposition 48.

Question n°3. Démontrer la proposition 49.

Question n°4. Démontrer la proposition 50.

Question n°5. Démontrer la proposition 51.

Question n°6. Démontrer la proposition 52.

Question n°7. Démontrer la proposition 53.

Chapitre 7

Introduction aux classes de complexité

7.1 Machines de Turing universelles

Théorème 20 *Il existe une machine de Turing déterministe $\mathcal{U}$ qui prend en entrée la représentation d'une machine de Turing déterministe $\mathcal{M}$ et un mot x de l'alphabet de $\mathcal{M}$ t.q. si T est le temps de calcul de $\mathcal{M}$ sur x alors $\mathcal{U}$ produit le même résultat en temps $O(T\log(T))$ (avec une constante qui dépend de $\mathcal{M}$ mais pas de x).*

Preuve
Dans un premier temps, nous construisons une machine $\mathcal{U}$ à bandes bidirectionnelles. $\mathcal{U}$ a une bande d'entrée, une bande de sortie et deux bandes de travail, appelées bande de simulation et bande d'état. La bande d'état contient une représentation de l'état courant de $\mathcal{M}$ et la position de la tête de lecture de $\mathcal{M}$ sur sa bande d'entréee et la bande de simulation, la seule à être utilisée de façon bidirectionnelle contient le contenu « juxtaposé » des bandes de travail. Le codage traditionnel de plusieurs bandes par une unique bande à alphabet fixé se fait en deux étapes. Tout d'abord si l'alphabet de $\mathcal{M}$ est Σ (avec $|\Sigma| \geq 2$) et si k est le nombre de bandes de travail de $\mathcal{M}$, on considère l'alphabet Σ^k. Chaque lettre de cet alphabet est alors associé à une représentation binaire de longueur $m = \lfloor \log_2(|\Sigma^k| - 1) \rfloor + 1$ et chaque groupe de k cellules juxtaposées est codé par un bloc de m cellules.

A l'aide d'une bande bidirectionnelle, on peut s'arranger pour déplacer le contenu des bandes simulées de telle sorte qu'au début d'un pas de simulation, la tête de lecture de la bande de simulation soit toujours à la position 0 (voir la figure 7.1). Cependant ce déplacement même limité aux caractères différents du blanc conduit à une simulation d'un pas en $O(T)$ (puisqu'il peut y avoir T caractères utiles) et par conséquent à une simulation en $O(T^2)$.

Afin de limiter les déplacements des bandes, on a recours à une représentation expansée des bandes à l'aide d'un caractère supplémentaire $\oplus$ différent du blanc. Dans cette représentation la bande est partitionnée en segments $\ldots, L_i, \ldots, L_1, [0,0], R_1, \ldots, R_i, \ldots$ t.q. $L_i = [-2^{i+1} + 2, -2^i + 1]$ et

$R_i = [2^i - 1, 2^{i+1} - 2]$. Au début de tout pas de simulation, chaque (représentation de) bande vérifie les propriétés suivantes :

- Chaque segment (L_i, R_i) est soit *vide*, soit *plein* soit *à moitié-plein*. Il est vide s'il ne contient que des $\oplus$, plein s'il ne contient pas de $\oplus$, et à moitié plein si la moitié de ces caractères sont des $\oplus$.
- L_i est vide (resp. plein, à moitié plein) ssi R_i est plein (resp. vide, à moitié plein)
- Initialement, tous les segments sont à moitié-plein (il suffit de remplacer la moitié des blancs de L_i et de R_i par des $\oplus$ la première fois qu'on rencontre L_i ou R_i).
- La position 0 ne contient pas de $\oplus$.

Nous décrivons maintenant sur chaque représentation de bande l'effet d'un déplacement. Nous nous limitons à un déplacement à droite car l'autre cas est symétrique.

- $\mathcal{U}$ cherche le plus petit R_i non vide (et donc L_i n'est pas plein).
- $\mathcal{U}$ recopie le premier caractère différent de $\oplus$ de R_i à la position 0 et les $2^{i-1}-1$ autres caractères suivants différents de $\oplus$ de R_i dans $R_1, R_2, \ldots, R_{i-1}$ en remplissant à moitié ces segments.
- A gauche de la tête de lecture, il y a l'ancien caractère de la position 0 et $2(2^i - 1)$ caractères différents de 0 dans $L_{i-1}, \ldots, L_1$. $\mathcal{U}$ recopie ensuite les 2^i caractères les plus à gauche dans L_i et dispose les $2^{i-1} - 1$ caractères restants de telle sorte que $L_{i-1}, \ldots, L_1$ soient à moitié pleins.

La représentation expansée et la simulation du déplacement sont illustrées par la figure 7.2 sur le même exemple que pour la représentation compacte.

Le point clef de la simulation est que lorsqu'un caractère de la bande L_i ou R_i est mis en position 0, alors ces deux segments ne peuvent être accédés qu'après au mois 2^{i-1} déplacements de la tête dans une direction car les segments $R_1, R_2, \ldots, R_{i-1}$ et $L_{i-1}, \ldots, L_1$ sont à moitié pleins. Nous avons illustré ce phénomène à la figure 7.3 pour $i = 3$.

Par conséquent si la machine $\mathcal{M}$ comporte b bandes de travail, alors les segments L_i et R_i sont déplacés au plus $\frac{bT}{2^{i-1}}$ fois. Un déplacement de L_i ou de R_i se fait en $O(2^i)$. Par conséquent, le temps global de la simulation est :

$$O(\sum_{i=1}^{\log_2(T)+1} \frac{bT2^i}{2^{i-1}}) = O(T \log_2(T))$$

La simulation d'une machine à bande bidirectionnelle par une machine à bande unidirectionnelle se fait en repliant la bande sur elle-même et le temps d'un pas de cette simulation est en $O(1)$.

Puisque seul le choix du pas de $\mathcal{M}$ est non déterministe, $\mathcal{U}$ est déterministe si elle se borne à simuler des machines déterministes.

c.q.f.d. $\diamondsuit\diamondsuit\diamondsuit$

Grace au pouvoir de calcul du non déterminisme, on peut produire une machine universelle plus efficace.

Théorème 21 *Il existe une machine de Turing non déterministe $\mathcal{U}$ qui prend en entrée la représentation d'une machine de Turing non déterministe $\mathcal{M}$ et un*

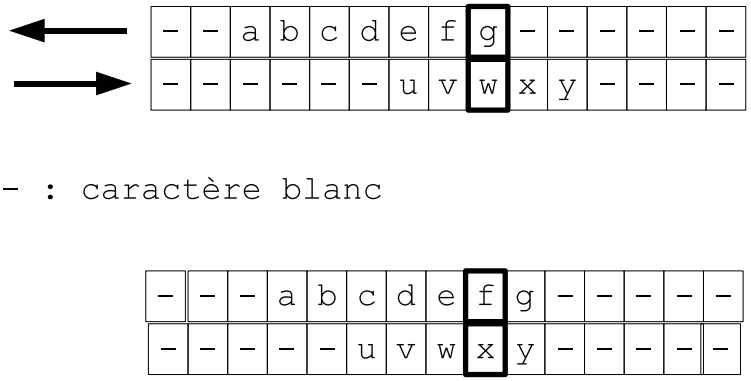

FIG. 7.1: Représentation compacte et simulation de plusieurs bandes

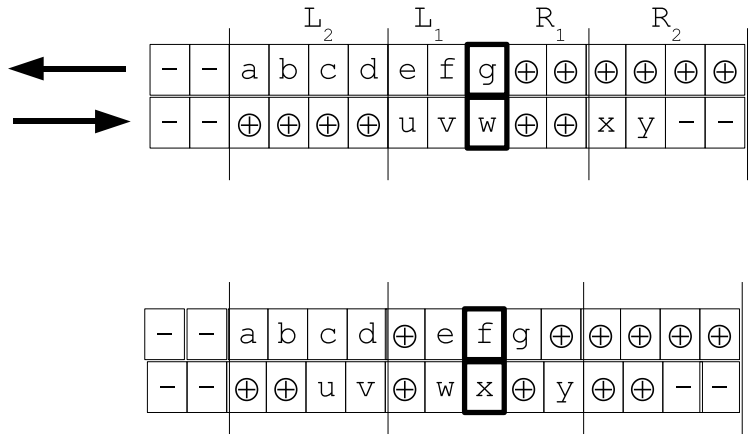

FIG. 7.2: Représentation expansée et simulation de plusieurs bandes

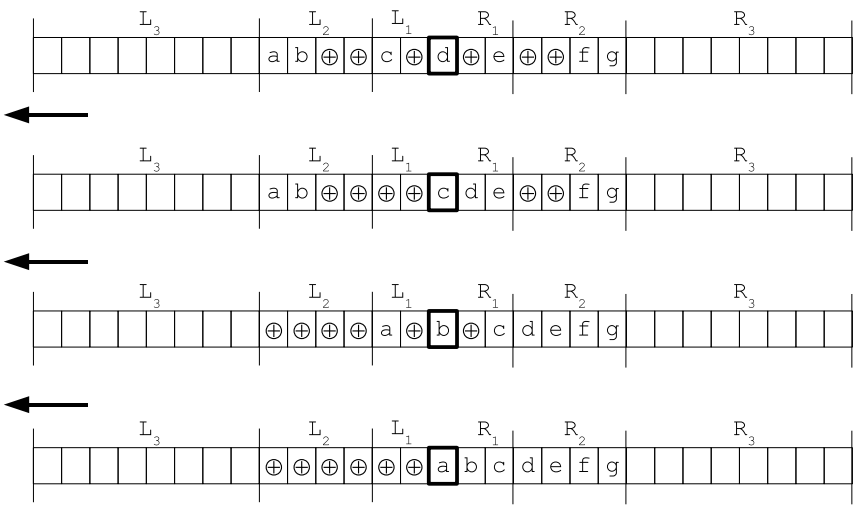

FIG. 7.3: Déplacement simulé d'une bande « équilibrée »

mot x de l'alphabet de $\mathcal{M}$ t.q. si T est le temps de calcul de $\mathcal{M}$ sur x alors $\mathcal{U}$ produit le même résultat en temps $O(T)$ (avec une constante qui dépend de $\mathcal{M}$ mais pas de x).

Preuve
D'après nos hypothèses, toute exécution de $\mathcal{M}$ se termine en un temps au plus T avec un résultat soit négatif soit positif. On suppose de plus que la machine dispose d'une transition qui ne fait rien et qui est possible uniquement dans l'état final.

La machine $\mathcal{U}$ a quelques bandes de travail : une bande de prédiction, une bande auxiliaire, une bande de compteur et une bande de sortie simulée correspondant à la bande sortie de $\mathcal{M}$. $\mathcal{U}$ itère le processus suivant à l'aide d'un compteur initialisé à 1. Sur sa bande de prédiction, il construit (ou complète la suite déjà construite) de manière non déterministe une suite de transitions de $\mathcal{M}$ de longueur égale au compteur et écrit le résultat correspondant sur la bande de sortie simulée. Deux transitions successives de cette suite sont t.q. l'état d'arrivée de la première transition est aussi l'état de départ de la deuxième transition. Dans le cas contraire, il vérifie une bande après l'autre que l'exécution prédite est réalisable sur chaque bande (à l'aide de sa bande auxiliaire). Si l'exécution n'est pas réalisale il rejette. Sinon il y a trois cas possibles :

- Le dernier état est un état final et la sortie est positive. $\mathcal{U}$ accepte.
- Le dernier état est un état final et la sortie est négative. $\mathcal{U}$ rejette.
- Le dernier état n'est pas un état final. $\mathcal{U}$ double la valeur du compteur et passe à l'itération suivante.

La correction de la simulation ne présente pas de difficulté. La simulation effectuera au plus $O(\log(T))$ tours. Le temps d'exécution de chaque tour est proportionnel à la valeur courante du compteur $1, 2, 4, \ldots$ qui ne peut excéder $2T$. D'où un temps cumulé en $O(\sum_k O(T/2^k)) = O(T)$.

Le point clef de cette simulation est la possibilité d'effectuer les simulations sur chaque bande de manière indépendante et donc d'éviter la recherche des têtes de lecture.

c.q.f.d. ◊◊◊

Théorème 22 *Il existe une machine de Turing (déterministe) $\mathcal{U}$ qui prend en entrée la représentation d'une machine de Turing (déterministe) $\mathcal{M}$ et un mot x de l'alphabet de $\mathcal{M}$ t.q. si E est l'espace nécessaire au calcul de $\mathcal{M}$ sur x alors $\mathcal{U}$ produit le même résultat en espace $O(\max(E, \log_2(|x|))$ (avec une constante qui dépend de $\mathcal{M}$ mais pas de x).*

Preuve
Il suffit d'adopter la représentation compacte du théorème 20. Le facteur $\log_2(|x|)$ provient du stockage de la tête de lecture de la bande d'entrée de $\mathcal{M}$.

c.q.f.d. ◊◊◊

7.2 Hiérarchies de complexité

Afin d'établir des théorèmes qui établissent des inclusions strictes entre classes de complexité, il est nécessaire de se retreindre à des fonctions de mesure de complexité réalistes (et réalisables).

Définition 63 *Soit $f : \mathbb{N} \mapsto \mathbb{N}^*$ une fonction, f est dite* temporellement *(resp.* spatialement*) constructible s'il existe une machine $\mathcal{M}$ qui produit pour une entrée de longueur n, une sortie constituée de la représentation binaire ou unaire de $f(n)$ (resp. d'un marqueur sur la cellule $f(n)$, e.g. le mot $0^{n-1}1$) en un temps inférieur ou égal à $f(n)$ (resp. en utilisant un espace inférieur ou égal à $f(n)$).*

La plupart des fonctions usuelles sont spatialement et temporellement constructibles : $n, n^2, 2^n, \ldots$. La fonction $\lceil \log_2(n+1) \rceil + 1$ est spatialement (mais pas temporellement) constructible.

Le deuxième ingrédient dont nous avons besoin est une représentation des machines de Turing telle qu'une machine admette une infinité de représentations et plus précisément telle que pour toute machine $\mathcal{M}$, il existe n_0 vérifiant $\forall n \geq n_0$ il existe une représentation de $\mathcal{M}$ de taille n. Cette représentation est très facile à construire. Donnons-nous une représentation quelconque des machines de Turing disons $x_{\mathcal{M}}$, la représentation recherchée est de la forme $1^n 0 x_{\mathcal{M}}$ pour n quelconque.

Nous présentons les théorèmes de hiérarchie par ordre de difficulté croissante.

Théorème 23 *Soient $f(n) \geq \log_2(n)$ et $g(n) \geq \log_2(n)$ deux fonctions spatialement constructibles vérifiant :*

$$\liminf_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Alors il existe un langage L accepté par une machine de Turing opérant sur une entrée x en espace $g(|x|)$ mais par aucune par une machine de Turing opérant sur une entrée x en espace $f(|x|)$.

Preuve
Nous construisons une machine $\mathcal{U}'$ qui est une variante de la machine universelle du théorème 22. $\mathcal{U}'$ a une bande supplémentaire dite bande de compteur pour stocker un compteur. Soit x une entrée de taille n, $\mathcal{U}'$ commence par marquer ses bandes de travail avec un marqueur en position $g(n)$. Par la suite, si sa simulation (y compris dans la phase initiale) le conduit à dépasser son marqueur il s'arrête et rejette.

Si x n'est pas la représentation d'une machine de Turing (disons $\mathcal{M}$) alors $\mathcal{U}'$ rejette. Dans le cas contraire, il initialise son compteur à $n_q f(n)^{n_t} n_a^{n_t f(n)}$ avec n_q le nombre d'états de $\mathcal{M}$, n_t le nombre de bandes de $\mathcal{M}$ et n_q le nombre de lettres de $\mathcal{M}$. Observons que d'une part ce compteur représente un majorant strict du plus grand nombre de pas que peut faire une machine qui se termine en opérant en espace $f(n)$ et d'autre part que ce compteur occupe une place en $O(f(n))$ si on fait croître n en laissant la machine $\mathcal{M}$ fixe.

Puis $\mathcal{U}'$ entreprend la simulation de $\mathcal{M}$ sur x en décrémentant son compteur et en avortant sa simulation si son compteur s'annule et en rejetant. Lorsque la simulation se termine, alors $\mathcal{U}'$ rejette ssi $\mathcal{M}$ accepte.

Soit L le langage accepté par $\mathcal{U}'$. Par construction $\mathcal{U}'$ opère en espace $g(n)$. Supposons que L soit reconnu par une machine $\mathcal{M}$ qui opère en espace $f(n) \geq \log_2(n)$. D'après le théorème 22 la simulation de $\mathcal{M}$ requiert un espace $O(f(n))$. Sachant que $\liminf_{n\to\infty} \frac{f(n)}{g(n)} = 0$ et qu'on peut choisir n quelconque suffisamment grand pour la taille d'une représentation x de $\mathcal{M}$, la simulation de $\mathcal{M}$ pour un tel x se poursuit jusqu'à son terme conduisant à un résultat différent pour $\mathcal{M}$ et $\mathcal{U}'$ d'où la contradiction.

c.q.f.d. ◊◊◊

Théorème 24 *Soient $f(n) \geq n$ et $g(n) \geq n$ deux fonctions temporellement constructibles vérifiant :*

$$\liminf_{n\to\infty} \frac{f(n)\log(f(n))}{g(n)} = 0 \wedge \lim_{n\to\infty} \frac{g(n)}{n} = \infty$$

Alors il existe un langage L accepté par une machine de Turing déterministe opérant sur une entrée x en temps $g(|x|)$ mais par aucune par une machine de Turing déterministe opérant sur une entrée x en temps $f(|x|)$.

Preuve
Nous construisons une machine $\mathcal{U}'$ qui est une variante de la machine universelle du théorème 20. $\mathcal{U}'$ a deux bandes supplémentaires dites bandes de compteur pour stocker un compteur binaire et unaire. Soit x une entrée de taille n, $\mathcal{U}'$ commence par calculer $g(n)$ en binaire puis à l'aide de ce résultat marque sa bande de compteur avec un marqueur en position $g(n)$ (le tout en temps $5g(n)$ où la constante 5 prend en compte l'initialisation puis la décrémentation du compteur binaire et le déplacement simultané du marqueur unaire). Par la suite, chaque pas d'exécution de $\mathcal{U}'$ déplace le marqueur à gauche avec arrêt et rejet si le marqueur se « déplace » à gauche de la bande.

Si x n'est pas la représentation d'une machine de Turing (disons $\mathcal{M}$) alors $\mathcal{U}'$ rejette.

Puis $\mathcal{U}'$ entreprend la simulation de $\mathcal{M}$ sur x. Si la simulation se termine sans rejet dû au compteur, alors $\mathcal{U}'$ rejette ssi $\mathcal{M}$ accepte.

Soit L le langage accepté par $\mathcal{U}'$. Par construction $\mathcal{U}'$ opère en temps $6g(n)$ dont $g(n)$ pour la simulation. Supposons que L soit reconnu par une machine $\mathcal{M}$ qui opère en espace $f(n)$. D'après le théorème 20 la simulation de $\mathcal{M}$ requiert un temps $O(f(n)\log(f(n)))$. Sachant que $\liminf_{n\to\infty} \frac{f(n)\log(f(n))}{g(n)} = 0$ et qu'on peut choisir n quelconque suffisamment grand pour la taille d'une représentation x de $\mathcal{M}$, la simulation de $\mathcal{M}$ pour un tel x se poursuit jusqu'à son terme conduisant à un résultat différent pour $\mathcal{M}$ et $\mathcal{U}'$. Donc L n'est pas reconnu par une machine qui opère en espace $f(n)$.

Puisque $\lim_{n\to\infty} \frac{g(n)}{n} = \infty$, il est possible de construire une machine $\mathcal{U}''$ qui reconnaît L et qui opère en temps $g(n)$ (ceci se fait en groupant les cellules par bloc sur les bandes de travail et en utilisant une bande de travail supplémentaire pour recopier l'entrée sous forme bloc).

c.q.f.d. ◊◊◊

Le théorème de hiérarchie pour les machines non déterministes est plus difficile à établir car l'argument de diagonalisation requiert pour la machine « universelle » le calcul de tous les chemins d'exécution de la machine à simuler afin d'accepter si tous ces chemins ne sont pas acceptants.

Théorème 25 *Soient $f(n) \geq n$ et $g(n) \geq n$ deux fonctions temporellement constructibles vérifiant :*

$$\liminf_{n\to\infty} \frac{f(n+1)}{g(n)} = 0 \wedge \lim_{n\to\infty} \frac{g(n)}{n} = \infty$$

Alors il existe un langage L accepté par une machine de Turing non déterministe opérant sur une entrée x en temps $g(|x|)$ mais par aucune par une machine de Turing non déterministe opérant sur une entrée x en temps $f(|x|)$.

Preuve

On définit d'abord une suite d'intervalles de $\mathbb{N}$ définis par $]u_k, u_{k+1}]$ (et l'intervalle $[0,0]$) avec $u_0 = 0$ et $u_{k+1} = f(u_k+1)2^{f(u_k+1)}$.

Nous construisons une machine $\mathcal{U}'$ qui est une variante de la machine universelle non déterministe du théorème 21. $\mathcal{U}'$ a deux bandes supplémentaires dites bandes de compteur pour stocker un compteur binaire et unaire. Soit x une entrée de taille n, $\mathcal{U}'$ commence par calculer $g(n)$ en binaire puis à l'aide de ce résultat marque sa bande de compteur avec un marqueur en position $g(n)$ (le tout en temps $5g(n)$ où la constante 5 prend en compte l'initialisation puis la décrémentation du compteur binaire et le déplacement simultané du marqueur unaire). Par la suite, chaque pas d'exécution de $\mathcal{U}'$ déplace le marqueur à gauche avec arrêt et rejet si le marqueur se « déplace » à gauche de la bande.

Si x n'est pas la représentation d'une machine de Turing (disons $\mathcal{M}$) alors $\mathcal{U}'$ rejette. Dans le cas contraire, $\mathcal{U}'$ détermine à quel intervalle $]u_k, u_{k+1}]$, $|x|$ appartient. Ceci se fait en un temps $O(|x|)$ en raison de la croissance exponentielle de la suite $\{u_k\}$.

Si $|x| < u_{k+1}$, $\mathcal{U}'$ entreprend la simulation de $\mathcal{M}$ sur $1x$. Si la simulation se termine sans rejet dû au compteur, alors $\mathcal{U}'$ accepte ssi $\mathcal{M}$ accepte.

Si $|x| = u_{k+1}$, $\mathcal{U}'$ entreprend une simulation *déterministe* de $\mathcal{M}$ sur le suffixe de x de taille u_k+1. Si la simulation se termine sans rejet dû au compteur, alors $\mathcal{U}'$ rejette ssi $\mathcal{M}$ accepte. Observons que cette simulation se fait en $O(T2^T)$ où T est le temps d'exécution de $\mathcal{M}$ sur ce suffixe.

Soit L le langage accepté par $\mathcal{U}'$. Par construction $\mathcal{U}'$ opère en temps $6g(n)$ dont $g(n)$ pour la simulation. Supposons que L soit reconnu par une machine $\mathcal{M}$ qui opère en espace $f(n)$. D'après le théorème 21 la simulation de $\mathcal{M}$ sur une entrée de taille différente d'un u_k requiert un temps $O(f(n+1))$. Sachant que $\liminf_{n\to\infty} \frac{f(n+1)}{g(n)} = 0$ et qu'on peut choisir n quelconque suffisamment grand pour un intervalle $]u_k, u_{k+1}]$ de taille de représentation de $\mathcal{M}$, la simulation de $\mathcal{M}$ pour les x se poursuit jusqu'à son terme. Seul le cas $|x| = u_{k+1}$ nécessite une explication. Puisque la simulation se fait sur un x' de taille u_k+1 la simulation déterministe prend un temps $O(f(u_k+1)2^{f(u_k+1)}) = O(u_{k+1}) = O(f(u_{k+1}+1))$.

Examinons maintenant les différents résultats. Dans la suite, x est la représentation de $\mathcal{M}$ de taille $|x|$. Pour tout $u_k+1 \leq |x| < u_{k+1}$, le résultat de $\mathcal{M}$ et de $\mathcal{U}'$ coïncident mais le résultat de $\mathcal{U}'$ sur x est par définition le résultat de $\mathcal{M}$ sur $1x$. Donc pour tout $u_k+1 \leq |x| \leq u_{k+1}$, le résultat de $\mathcal{M}$ est identique. Or $\mathcal{U}'$ sur l'entrée x de taille u_{k+1} a le résultat opposé à celui de $\mathcal{M}$ sur l'entrée x de taille u_k+1 donc à celui sur l'entrée x de taille u_{k+1}. D'où la contradiction.

Donc L n'est pas reconnu par une machine qui opère en espace $f(n)$.

Puisque $\lim_{n\to\infty} \frac{g(n)}{n} = \infty$, il est possible de construire une machine $\mathcal{U}''$ qui reconnaît L et qui opère en temps $g(n)$ (ceci se fait en groupant les cellules par bloc sur les bandes de travail et en utilisant une bande de travail supplémentaire pour recopier l'entrée sous forme bloc).

c.q.f.d. ◊◊◊

On introduit les classes de complexités suivantes.

Définition 64 *Un langage L appartient à :*
- *$D(f(n))$ (resp. $ND(f(n))$) s'il existe une machine de Turing déterministe (resp. non déterministe) opérant en temps $f(n)$ sur un mot de longueur n dont le langage est L.*
- *$DSPACE(f(n))$ (resp. $NDSPACE(f(n))$) s'il existe une machine de Turing déterministe (resp. non déterministe) opérant en espace $f(n)$ sur un mot de longueur n dont le langage est L.*

Le théorème qui suit correspond à des inclusions larges obtenues par deux observations élémentaires : une machine déterministe est un cas particulier de machine non déterministe et une machine ne peut occuper plus d'espace que son temps d'exécution.

Théorème 26 *On a pour toute fonction f les inclusions suivantes :*
- $D(f(n)) \subseteq ND(f(n)) \subseteq NDSPACE(f(n))$
- $D(f(n)) \subseteq DSPACE(f(n)) \subseteq NDSPACE(f(n)$

En pratique, seules certaines classes de complexité correspondent à des cas couramment rencontrés. Ainsi on définit :
- $LOGSPACE \equiv DSPACE(\log_2(n))$,
  $NLOGSPACE \equiv NDSPACE(\log_2(n))$,

- $P \equiv \bigcup_{k\in\mathbb{N}} D(n^k)$ (appelée aussi $PTIME$),
  $NP \equiv \bigcup_{k\in\mathbb{N}} ND(n^k)$ (appelée aussi $NPTIME$),
- $PSPACE \equiv \bigcup_{k\in\mathbb{N}} DSPACE(n^k)$,
  $NPSPACE \equiv \bigcup_{k\in\mathbb{N}} NDSPACE(n^k)$,
- $EXP \equiv \bigcup_{k\in\mathbb{N}} D(2^{n^k})$ (appelée aussi $EXPTIME$),
  $NEXP \equiv \bigcup_{k\in\mathbb{N}} ND(2^{n^k})$ (appelée aussi $NEXPTIME$), etc.

A l'aide du théorème précédent, on obtient les inclusions suivantes. Nous en rediscuterons à la fin de la prochaine section.

$$LOGSPACE \subseteq NLOGSPACE \subseteq PTIME \subseteq NPTIME \subseteq NPSPACE$$

7.3 Egalité de classes de complexité

7.3.1 Le théorème de Savitch

Théorème 27 (Savitch) *Le problème de l'accessibilité dans un graphe orienté à n sommets se résout par un algorithme en taille d'espace $O(log^2(n))$.*

Preuve
L'algorithme proposée est une procédure récursive, disons `Cherche` qui prend en entrée trois paramètres, un sommet source `in`, un sommet destination `out` et une longueur `lg`. Cette procédure renvoie vrai s'il existe un chemin de longueur au plus `lg` qui joint `in` à `out`. Initialement, cette procédure sera appelée avec les deux sommets pour lesquels on veut tester l'accessibilité et une longueur égale à $n-1$. La procédure procède ainsi :

- Si $\texttt{lg} \leq 1$ elle teste si $\texttt{in} = \texttt{out}$ ou s'il existe un arc $(\texttt{in}, \texttt{out})$.
- Si $\texttt{lg} > 1$ elle parcourt l'ensemble des sommets avec une variable `mid`. Dans une itération, elle teste par deux appels récursifs s'il existe un chemin d'une longueur au plus $\lceil \frac{\texttt{lg}}{2} \rceil$ qui joint `in` à `mid` et un chemin d'une longueur au plus $\lfloor \frac{\texttt{lg}}{2} \rfloor$ qui joint `mid` à `out`. Si elle les trouve elle renvoie vrai. A la fin des itérations, elle renvoie faux.

La correction de cette procédure est évidente. Analysons sa complexité en espace. Il y au plus $\lceil \log_2(n) \rceil + 1$ appels emboîtés. Les sommets sont représentés par des identifiants de taille $\lceil \log_2(n) \rceil + 1$ puisqu'il y a n sommets. Enfin la longueur maximale est $n-1$ et peut aussi être codée sur $\lceil \log_2(n) \rceil + 1$ bits. Par conséquent chaque appel consomme $O(\log_2(n))$ espace.

c.q.f.d. $\Diamond\Diamond\Diamond$

L'accessibilité dans un graphe est un problème très proche de l'acceptation d'un mot par une machine de Turing qui opère en espace borné.

Corollaire 10 *Pour toute fonction f space-calculable t.q. $f(n) \geq \log_2(n)$, $NSPACE(f(n)) \subseteq SPACE(f^2(n))$*

Preuve
Soit $\mathcal{M}$ une machine de Turing non déterministe opérant en espace $f(n)$ sur un mot x de taille n. On fait l'hypothèse non restrictive qu'il existe une unique configuration acceptante. On construit la machine $\mathcal{M}'$ une machine de Turing déterministe opérant en espace $f^2(n)$ ainsi. Elle calcule d'abord $f(n)$ pour déterminer la taille des configurations à considérer. Puis $\mathcal{M}'$ teste l'accessibilité

de la configuration acceptante à partir de la configuration initiale en implémentant l'algorithme du théorème 27. Elle ne consulte son entrée que lorsqu'elle teste l'accessibilité en au plus un pas. La contrainte sur $\log_2(n)$ est nécessaire car dans une configuration de $\mathcal{M}$, la représentation de la position de la tête de lecture de la bande d'entrée occupe $O(\log_2(n))$ bits.

c.q.f.d. $\Diamond\Diamond\Diamond$

Le corollaire suivant est certainement le plus utilisé mais on a aussi $EXPSPACE = NEXPSPACE$, etc.

Corollaire 11 $PSPACE = NPSPACE$

Notre suite d'inclusions devient :

$$LOGSPACE \subseteq NLOGSPACE \subseteq PTIME \subseteq NPTIME \subseteq PSPACE$$

De plus, on a $NLOGSPACE \subsetneq PSPACE$ car $NLOGSPACE \subseteq DSPACE(\log^2(n))$. Par contre, il est conjecturé que toutes les inclusions ci-dessus sont strictes mais ces problèmes sont ouverts depuis longtemps ...

7.3.2 Le théorème d'Immerman-Szelepscényi

Dans le lemme suivant, la sémantique d'une machine de calcul non déterministe est la suivante : lorsqu'elle accepte, le résultat est sur sa bande de calcul (ou sur une bande spécifique) et le résultat doit être le même quelque soit la configuration acceptante.

Lemme 44 *Soit $\mathcal{M}$ une machine de Turing non déterministe s'exécutant en un espace de taille $s(n) \geq \log_2(n)$ space-calculable[1] où n est la taille de l'entrée. Alors il existe une machine de Turing non déterministe $\mathcal{M}'$ s'exécutant en un espace de taille $O(s(n))$, qui calcule N, le nombre de configurations atteignables depuis la configuration initiale.*

Preuve
Avant tout, la machine $\mathcal{M}'$ calcule $s(n)$.

Notons N_d, le nombre de configurations différentes atteintes par la machine $\mathcal{M}$ depuis la configuration initiale en au plus d pas. $N_0 = 1$. La machine $\mathcal{M}'$ calcule itérativement N_d. Supposons que $\mathcal{M}'$ ait une exécution qui a calculé N_d, elle continue son calcul de la façon suivante :

- Elle initialise un compteur *cptext* à 0. Puis elle énumère les configurations occupant une place $s(n)$.
- Pour chaque configuration, disons *current*, elle initialise un compteur, disons *cptint* et énumère de nouveau les configurations occupant une place $s(n)$. Dans cette boucle interne, elle teste de manière non déterministe si la configuration courante de cette boucle, disons *local* est accessible en au plus d pas en devinant un chemin de longueur au plus d. Ce test non déterministe nécessite seulement une configuration et un compteur. Si c'est le cas, elle incrémente *cptint* puis elle teste si *current* est accessible en un

[1] Cette hypothèse n'est nécessaire ni ici ni dans la suite de cette section mais elle simplifie la preuve.

pas à partir de *local* et dans ce cas incrémente *cptext* et sort de la boucle interne. Si elle termine sa boucle interne (sans avoir incrémenté *cptext*). Elle contrôle si le compteur *cptint* est égal à N_d. Si ce n'est pas le cas elle s'arrête sans accepter.
- A la fin de l'énumération la plus externe *cptext* est égal à N_{d+1} pusique les seules erreurs possibles ont été détectées par le contrôle effectué *via cptint*.

La machine s'arrête lorsque $N_{d+1} = N_d = N$. La machine occupe $O(s(n))$ espace puisque les compteurs sont bornés par $2^{s(n)}$ les configurations par $s(n)$.

c.q.f.d. ◊◊◊

Théorème 28 (Immerman-Szelepscényi) *Soit $\mathcal{M}$ une machine de Turing non déterministe s'exécutant en un espace de taille $s(n) \geq \log_2(n)$ space-calculable où n est la taille de l'entrée. Alors il existe une machine de Turing non déterministe $\mathcal{M}'$ s'exécutant en un espace de taille $O(s(n))$, qui accepte le langage complémentaire de celui accepté par $\mathcal{M}$.*

Preuve
La machine $\mathcal{M}'$ est presque identique celle du lemme 44. La seule différence réside dans le fait que lorsqu'elle trouve une configuration acceptante de $\mathcal{M}$, elle s'arrête et rejette. Par conséquent, si elle se termine en acceptant cela signifie qu'à son dernier calcul de N_d, elle a rencontré toutes les configurations accessibles de $\mathcal{M}$ et qu'aucune n'est acceptante.

c.q.f.d. ◊◊◊

Le corollaire les plus important est relatif à $NLOGSPACE$. Pour cela, on introduit de nouvelles classes de complexité relatives à des machines non déterministes dont la sémantique est la suivante : ces machines acceptent si toutes leurs exécutions sont acceptantes. Clairement si un langage L est accpeté par une machine non déterministe « existentielle », le langage complémentaire est accepté par une machine non déterministe « universelle » et vice versa. Toutes les classes non déteministes avec sémantique universelle seront notées avec le préfixe *co*. Le résultat précédent donne donc lieu à ce corollaire important.

Corollaire 12 *co-$NLOGSPACE = NLOGSPACE$*

7.4 Problèmes P-space complets

7.4.1 Universalité des langages réguliers

Tester la vacuité d'un langage régulier $L = \emptyset$? (où L est donné par un automate ou une expression régulière) se fait en temps polynomial car ce problème se réduit à un problème d'accessibilité dans un graphe (e.g. existe-t-il un chemin de l'état initial vers un état final ?). Tester l'universalité $L = \Sigma^*$? n'est pas *a priori* un problème équivalent car le passage à l'automate complémentaire entraine un facteur d'accroissement exponentiel. Nous allons donc étudier ce problème. Dans notre étude, nous utiliserons indifféremment la spécification du langage régulier par un automate non déterministe sans ε-transition ou par une expression rationnelle car il existe des traductions en temps polynomial d'une représentation vers l'autre.

Proposition 54 *Soit $\mathcal{A}$ un automate non déterministe sur un alphabet Σ sans ε-transition et $L(\mathcal{A})$ son langage. Alors le problème de l'universalité de $L(\mathcal{A})$ est dans PSPACE.*

Preuve
Supposons que $\mathcal{A}$ ne reconnaisse pas un mot, alors $\mathcal{A}^c$ l'automate déterministe complémentaire obtenu par la construction des sous-ensembles reconnaît ce mot. Il y a donc un chemin dans $\mathcal{A}^c$ de l'état initial vers un état final. Or l'automate complémentaire a au plus 2^n états. Donc le chemin le plus court de l'état initial vers un état final a une longueur au plus égale à $2^n - 1$.

Nous recherchons ce chemin par une procédure non déterministe sans construire $\mathcal{A}^c$. Cette procédure maintient un compteur initialisé à $2^n - 1$ et un état courant de $\mathcal{A}^c$ (i.e. un sous-ensemble d'états de $\mathcal{A}$). Elle choisit de manière non déterministe un lettre de Σ et construit le successeur de l'état courant dans $\mathcal{A}^c$ et décrémente le compteur. Ceci s'effectue uniquement à l'aide de $\mathcal{A}$. Elle itère ce processus jusqu'à ce qu'elle recontre un état final de $\mathcal{A}^c$ et renvoie vrai ou que le compteur soit nul et elle renvoie faux.

Cette procédure occupe un espace polynomial (compteur et état représentés en $O(n)$). Il suffit alors de la déterminiser par la procédure de Savitch.

c.q.f.d. $\Diamond\Diamond\Diamond$

La borne supérieure est fait optimale ainsi que le démontre la proposition suivante.

Proposition 55 *Soit E une expression rationnelle sur un alphabet Σ et $L(E)$ son langage. Alors le problème de l'universalité de $L(E)$ est PSPACE-difficile.*

Preuve
Soit une machine de Turing déterministe $\mathcal{M} = (Q, T, A, \delta, \flat, q_0, q_f)$ qui s'exécute en espace polynomial vis à vis de son entrée x, disons $p(n)$ où p est un polynôme et n est la taille de x. Q est l'ensemble des états, T les symboles de la bande, $A \subseteq T \setminus \{\flat\}$ l'alphabet des entrées, $\flat \in T$ le blanc, q_0, q_f les états initial et final.

Nous associons un mot à chaque exécution de la machine. L'alphabet de ce mot est $\Sigma \equiv T \cup \{qX \mid q \in Q \wedge X \in T\} \cup \{\sharp\}$. On note $\Delta = \Sigma \setminus \{\sharp\}$ et $QT = \{qX \mid q \in Q \wedge X \in T\}$. Le mot s'écrit $w = \sharp w_1 \sharp w_2 \sharp \ldots \sharp w_n \sharp$ pour une exécution de longueur n avec w_i la représentation de la $i$ème configuration. Chaque mot w_i a une longueur $p(n)$ et représente le contenu de la bande avec un unique symbole qX signalant à la fois l'état de la machine et la position de la machine.

Nous allons construire une expression régulière E qui accepte les mots w qui ne sont pas des codages d'exécution ou ceux qui codent des exécutions qui ne rencontrent pas l'état q_f. Autrement dit, $E = \Sigma^*$ ssi x n'est pas accepté par $\mathcal{M}$. Enumérons les différents cas possibles.

1. w ne contient pas un symbole $q_f X$. Appelons A ce langage.
2. w ne contient pas la configuration initiale mais $\sharp(q_0 x_1) x_2 \ldots x_n \flat \ldots \flat$ avec $x = x_1 \ldots x_n$ et$\flat$ apparaît $p(n) - n$ fois, n'est pas un préfixe de w. Appelons B ce langage.

3. w n'est pas de la forme $w = \sharp w_1 \sharp w_2 \sharp \ldots \sharp w_n \sharp$ avec pour tout i, $|w_i| = p(n)$, une lettre de w_i est de la forme qX et toutes les autres lettres de w_i appartiennent à T. Appelons C ce langage.
4. w contient deux configurations successives qui ne correspondent pas à un pas de la machine. Appelons D ce langage.

Dans la suite, si $S = \{s_1, \ldots, s_k\}$ est un sous-ensemble de lettres, on utilise S comme abréviation de l'expression rationnelle $s_1 + \ldots + s_k$. et E^i comme abréviation de $E.E.\ldots.E$ où E apparaît i fois, E^0 étant ε. Soit l'expression $E_1 \equiv (\Sigma \setminus \{q_f X \mid X \in T\})^*$, alors $L(E_1) = A$.

Soit :
- $E_{2,1} = \Delta.\Sigma^*$
- $E_{2,2} = \Sigma.\Sigma \setminus \{q_0 x_1\}.\Sigma^*$
- $\forall 3 \leq i \leq n+1$ $E_{2,i} = \Sigma^i.\Sigma \setminus \{x_i\}.\Sigma^*$
- $\forall n+2 \leq i \leq p(n)+1$ $E_{2,i} = \Sigma^i.\Sigma \setminus \{\flat\}.\Sigma^*$

Posons $E_2 = E_{2,1} + \ldots + E_{2,p(n)+1} + \varepsilon + \Sigma + \ldots + \Sigma^{p(n)}$. Alors $L(E_2) = B$.

Soit :
- $\forall 0 \leq i \leq p(n)-1$ $E_{3,i} = \Sigma^*.\sharp.\Delta^i.\sharp.\Sigma^*$
  (mots avec des configurations trop courtes)
- $E_{3,p(n)+1} = \Sigma^*.\sharp.\Delta^{p(n)+1}.\Delta^*.\sharp.\Sigma^*$
  (mots avec des configurations trop longues)
- $F_3 = \Delta^* + \Delta^*.\sharp.\Delta^* + \Delta.\Sigma^* + \Sigma^*.\Delta$
  (mots sans ou avec un seul $\sharp$, ne commençant ou ne finissant pas par $\sharp$)
- $G_3 = \Sigma^*.\sharp.T^*.\sharp.\Sigma^* + \Sigma^*.\sharp.T^*.QT.T^*.QT.T^*.\sharp.\Sigma^*$
  (mots avec des configurations ne contenant pas d'état ou contenant deux états)

Posons $E_3 = E_{3,1} + \ldots + E_{3,p(n)-1} + E_{3,p(n)+1} + F_3 + G_3$. Alors $L(E_3) = C$.

Remarquons que si un mot code une exécution alors trois lettres consécutives $a_{i-1}a_i a_{i+1}$ de ce mot appartenant à Δ, déterminent de façon unique la lettre $a_{p(n)+i+1}$ si elle existe. Appelons $f : \Delta^3 \mapsto 2^\Sigma$, la fonction qui associe à trois lettres de Δ, le sous-ensemble de Σ privé de la lettre ainsi déterminée (où Σ lui-même dans le cas d'une suite de trois lettres contenant au moins deux lettres de QT). Posons $D_{c_1,c_2,c_3} = \Sigma^*.c_1.c_2.c_3.\Sigma^{p(n)}.f(c_1,c_2,c_3).\Sigma^*$ et E_4 la somme des D_{c_1,c_2,c_3}. Alors $L(E_4) = D$.

Nous laissons le soin au lecteur de vérifier que $E = E_1 + E_2 + E_3 + E_4$ est taille polynomiale vis à vis de n.

c.q.f.d. $\Diamond\Diamond\Diamond$

7.4.2 Satisfaisabilité d'une formule booléenne quantifiée

Une *formule booléenne quantifiée* (QBF) est une formule de la forme $\varphi \equiv Q_n x_n \ldots Q_1 x_1 \psi$ où φ est une formule propositionnelle sur l'ensemble des propositions $\{x_1, \ldots, x_m\}$ avec $m \geq n$. Soit une affectation $\overline{e} : \{x_1, \ldots, x_m\} \mapsto \{\mathtt{V}, \mathtt{F}\}$, on définit inductivement (par rapport à n) la satisfaction de φ par $\overline{e}$, notée $\overline{e} \models \varphi$
- Si $n = 0$, il s'agit de satisfaction standard en logique propositionnelle.
- Si $Q_n = \exists$, $\overline{e} \models \varphi$ ssi il existe $v \in \{\mathtt{V}, \mathtt{F}\}$ t.q. $\overline{e}[v, n] \models Q_{n-1}x_{n-1} \ldots Q_1 x_1 \psi$.
- Si $Q_n = \forall$, $\overline{e} \models \varphi$ ssi pour tout $v \in \{\mathtt{V}, \mathtt{F}\}$, $\overline{e}[v, n] \models Q_{n-1}x_{n-1} \ldots Q_1 x_1 \psi$.

Le problème de la satisfaisabilité consiste à savoir s'il existe une affectation des variables (booléennes) qui satisfasse la formule. Dans le cas d'une formule close, la satisfaction ne dépend pas de l'affectation des variables. On remarque aussi que vis à vis du problème de la satisfaisabilité on peut se ramener au cas d'une formule close en quantifiant existentiellement les variables libres. Nous proposons un algorithme pour résoudre ce problème.

Proposition 56 *Soit φ formule booléenne quantifiée close. Alors le problème de la satisfaisabilité de φ est dans $PSPACE$.*

Preuve
L'algorithme consiste en procédure récursive qui prend en entrée une formule QBF, φ dont la sous-formule propositionnelle contient les variables quantifiées, `V`, `F` et les connecteurs booléens.

Si la formule n'est pas quantifiée, alors on évalue en temps et en espace polynomial la formule qui ne contient que des constantes et on renvoie le résultat.

Si le quantificateur le plus externe est $\forall$ (resp. $\exists$) la procédure remplace dans la formule passée en paramètre la variable par `V`, supprime la quantification et s'appelle récursivement. Si le résultat de l'appel est `F` (resp. `V`) la procédure renvoie `F` (resp. `V`). Sinon la procédure remplace dans la formule passée en paramètre la variable par `F`, supprime la quantification, s'appelle récursivement et renvoie le résultat de son appel.

Le nombre d'appels simultanés est au plus $n+1$ où n est le nombre de quantificateurs. Chaque appel consomme un espace linéaire pour stocker les paramètres et les variables locales. Par conséquent, cet algorithme est dans $PSPACE$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Nous voulons obtenir un résultat de complexité en restreignant la forme des formules QBF ce qui nous conduit à introduire ce lemme.

Lemme 45 *La formule propositionnelle $\varphi \equiv \bigwedge_{i\leq n} \varphi_i$ et la formule QBF $\psi \equiv \forall z_1 \ldots \forall z_n \bigvee_{i\leq n}(\varphi_i \wedge z_i) \vee (\bigwedge_{i\leq n} \neg z_i)$ sont équivalentes.*

Preuve
Supposons que φ est vrai. Donc pour tout i, φ_i est vrai. Soit $v_1, \ldots v_n$ une valuation de $z_1, \ldots z_n$. Si tous les v_i sont faux alors la dernière clause conjonctive de ψ est vrai. Sinon supposons v_i vrai, alors la clause $\varphi_i \wedge z_i$ est vrai.

Supposons que ψ est vrai. Pour un i quelconque, choisisssons la valuation $v_1, \ldots v_n$ t.q. v_i est vrai et tous les v_j, pour $j \neq i$, sont faux. Alors φ_i doit être vrai. Donc φ est vrai.

c.q.f.d. $\Diamond\Diamond\Diamond$

Proposition 57 *Soit φ une formule booléenne quantifiée close dont la sous-formule propositionnelle est sous forme normale disjonctive. Alors le problème de la satisfaisabilité de φ est $PSPACE$-difficile donc $PSPACE$-complet.*

Preuve
Soit une machine de Turing déterministe $\mathcal{M}$ qui s'exécute en espace polynomial vis à vis de son entrée.

La configuration de la machine peut être codée sous forme de bits donc de variables booléennes disons $x_1, \ldots, x_n$ où n est une fonction polynomiale de la taille de l'entrée de la machine de Turing puisque l'espace utilisé est polynomial. Par exemple, voici un codage possible (avec renommage) :

- $state_q$ est vrai si l'état de la configuration est q.
- $head_i$ est vrai si la tête est à la position i.
- $tape_{c,i}$ est vrai si le contenu de la bande à la position i est le caractère c.

Certaines valuations ne correspondent pas à des configurations mais cela ne nous gênera pas dans la suite.

Le nombre de configurations différentes est d'au plus 2^n et une exécution acceptante exécute au plus $2^n - 1$ pas. Nous allons écrire récursivement une formule QBF φ_i avec $2n$ variables libres $x_1, \ldots, x_n, y_1, \ldots, y_n$ t.q. φ est vrai si partant de la configuration représentée par $x_1, \ldots, x_n$ la machine peut atteindre en au plus 2^i pas la configuration représentée par $y_1, \ldots, y_n$. Dans la suite de la preuve $\overline{x}$ ($\overline{y}$, etc.) désigne un vecteur de variables et $Q\overline{x}$ signifie que le quantificateur Q est répété devant chacune des variables de $\overline{x}$.

La formule φ_0 est définie par $\varphi_0 \equiv \bigwedge_{i \leq n} x_i = y_j \vee \varphi'_0$ où φ'_0 représente le fait que $\overline{x}$ atteint $\overline{y}$ par un pas de la machine. Nous transformons $\bigwedge_{i \leq n} x_i = y_j$ en :
$\forall z_1 \ldots \forall z_n \bigvee_{i \leq n} (x_i \wedge y_i \wedge z_i) \vee (\neg x_i \wedge \neg y_i \wedge z_i) \vee (\bigwedge_{i \leq n} \neg z_i)$
grace au lemme 45. La formule a une taille linéaire par rapport à n.

La formule φ'_0 est une conjonction de formules :
$\neg state^x_q \vee \neg head^x_i \vee \neg tape^x_{c,i} \vee tape^y_{newt(q,c),i}$
$\neg state^x_q \vee \neg head^x_i \vee \neg tape^x_{c,i} \vee state^y_{newq(q,c)}$
$\neg state^x_q \vee \neg head^x_i \vee \neg tape^x_{c,i} \vee head^y_{i+dep(q,c)}$
$head^x_i \vee \neg tape^x_{c,i} \vee tape^x_{c,i}$
etc.
Il y a un nombre linéaire de clauses (par rapport à n) chacune d'au plus 4 littéraux. On applique le lemme 45 puis chaque sous-formule comme :
$(\neg state^x_q \vee \neg head^x_i \vee \neg tape^x_{c,i} \vee tape^y_{newt(q,c),i}) \wedge z_j$
est transformée en :
$(\neg state^x_q \wedge z_j) \vee (\neg head^x_i \wedge z_j) \vee (\neg tape^x_{c,i} \wedge z_j) \vee (tape^y_{newt(q,c),i} \wedge z_j)$
La taille de la nouvelle formule est au plus le triple de la formule originale.

Supposons que nous ayons construit φ_i. Alors φ_{i+1} est défini en introduisant une configuration intermédiaire. De plus, à l'aide d'une construction astucieuse, on évite d'écrire deux fois φ_i ($\overline{z}$, $\overline{t}$ et $\overline{u}$ sont de nouvelles variables).

$$\varphi_{i+1} \equiv \exists \overline{z} \forall \overline{t} \forall \overline{u} ((\overline{x} = \overline{u} \wedge \overline{z} = \overline{t}) \vee (\overline{z} = \overline{u} \wedge \overline{y} = \overline{t})) \Rightarrow \varphi_i(\{\overline{x} \leftarrow \overline{u}\} \cup \{\overline{y} \leftarrow \overline{t}\})$$

Cette formule n'est pas tout à fait sous la forme recherchée. A l'aide des transformations déjà vues lors des formes prénexes puisque les variables liées de φ_i sont absentes du reste de la formule, on peut les placer les quantificateurs de φ_i à la suite des quantificateurs de tête φ_{i+1}. On remplace $A \Rightarrow B$ par $\neg A \vee B$. Enfin :
$\neg((\overline{x} = \overline{u} \wedge \overline{z} = \overline{t}) \vee (\overline{z} = \overline{u} \wedge \overline{y} = \overline{t}))$ est transformée en une disjonction de $16n^2$ clauses conjonctives traduisant les différentes façons de satisfaire la formule, comme par exemple :
$x_i \wedge \neg u_i \wedge y_j \wedge \neg t_j$
$\neg x_i \wedge u_i \wedge y_j \wedge \neg t_j$

$z_i \wedge \neg t_i \wedge y_j \wedge \neg t_j$
etc.

La taille de laformule croît ainsi d'un facteur additif polynomial en fonction de n. Pour terminer, la formule recherchée est la formule close $\varphi_n(\{\overline{x} \leftarrow \overline{init}\} \cup \{\overline{y} \leftarrow \overline{fin}\})$. $\overline{init}$ décrit la configuration initiale et $\overline{fin}$ décrit la configuration finale (sans perte de généralité, on peut supposer qu'il en existe une seule).

c.q.f.d. ◊◊◊

Le corollaire nous ramène à une variante d'un problème connu.

Corollaire 13 (QSAT) *Soit φ une formule booléenne quantifiée close dont la sous-formule propositionnelle est sous forme normale conjonctive. Alors le problème de la satisfaisabilité de φ est $PSPACE$-complet.*

Preuve
Soit un problème Pb dans $PSPACE$, le problème complémentaire Pbc est dans $PSPACE$ et se réduit à l'évaluation d'une formule QBF sous forme normale disjonctive φ. Par conséquent tout problème dans $PSPACE$ se réduit à l'évaluation de $\neg\varphi$ où φ est une formule QBF sous forme normale disjonctive mais par les transformations vues aux chapitres 2 et 3, $\neg\varphi$ est équivalente à une formule QBF sous forme normale conjonctive d'une taille égale au plus au double de la taille de φ.

c.q.f.d. ◊◊◊