Éléments de Logique
pour le cours de 2ème année Ensimag:
Fondements de Logique pour l'Informatique

Thierry Boy de la Tour Mnacho Echenim

Année 2017-2018

Table des matières

Ce polycopié est constitué des définitions et preuves des résultats qui seront étudiés dans le cours de Logique au premier semestre de la deuxième année à l'Ensimag. C'est un support de cours qui doit servir de référence aux étudiants et qui n'est pas destiné à remplacer la présence en classe. Il est divisé en trois chapitres :

Inférences et logiques. Ce chapitre permet de comprendre comment des énoncés peuvent être engendrés à partir d'autres énoncés (systèmes d'inférence), comment attribuer une valeur de vérité à ces énoncés (logiques), et les liens entre ces deux notions. Ce chapitre assez abstrait sert de base pour tout ce qui suit dans le cours de logique.

La logique propositionnelle. Ce chapitre définit dans un cadre général les notions connues de variables booléennes, de tables de vérité... Puis différents systèmes d'inférence permettant de tester efficacement la satisfaisabilité d'une formule sont présentés, ainsi que le théorème fondamental de compacité de la logique propositionnelle.

La logique du premier ordre Ce chapitre définit de façon formelle la logique du premier ordre, puis présente les théorèmes de Herbrand et de Löwenheim-Skolem avant d'introduire l'opération d'unification qui est très utilisée dans de nombreux domaines (Bases de Données, IA...), et la règle de résolution pour la logique du premier ordre.

Ceci est la deuxième version du polycopié, il est certain qu'il contient encore des coquilles. Nous sommes reconnaissants à Cyril Lorenzetto pour sa relecture de la première version et la détection de nombreuses coquilles ; nous serons reconnaissants à tout lecteur qui nous signalera les coquilles restantes.

Prolégomènes sur les ensembles

Ce cours de logique s'appuye sur certaines notions issues des mathématiques modernes qui ne sont plus systématiquement enseignées et font parfois défaut aux étudiants. S'il n'est pas question de faire de tout ingénieur un mathématicien, il faut bien constater que l'informatique fait un usage intensif de notions comme les entiers, les nombres réels, les tableaux (qui ressemblent étrangement à des fonctions), les listes, les mots, les pointeurs (qui permettent de former des graphes), etc. Ce n'est pas un hasard si ces objets sont tous de nature mathématique. Écrire un algorithme c'est décrire une infinité de calculs possibles et les mathématiques sont le seul outil intellectuel permettant d'assoir les raisonnements indispensables à une tâche aussi complexe. Une des causes de cette complexité est la nécessité de penser l'infini.

Or cette notion est historiquement l'une des plus controversée depuis l'antiquité et celle qui a fait dire le plus d'âneries teintées de mysticisme à des penseurs pourtant brillants, au point que Galilée considérait qu'il fallait l'éviter dans les raisonnements car elle menait inévitablement à des contradictions. Il est donc illusoire d'espérer faire mieux que ces antiques génies sans se donner des moyens dont ils ne disposaient pas.

Car ces moyens existent, ils ont été découverts au XIXème siècle par Georg Cantor et ils ont permis dans une large mesure de résoudre le problème de l'infini. Et l'outil qui permet de penser l'infini est précisément celui qui est au centre des mathématiques modernes, c'est la *théorie des ensembles*. S'il n'est pas question ici de développer cette théorie puisqu'on ne saurait le faire en quelques pages, il nous a cependant semblé utile pour la compréhension du cours d'en décrire certains aspects.

La notion d'ensemble fait partie de celles qui sont considérées comme suffisamment simples pour ne pas être définies bien qu'on y ait constamment recours. Tout le monde connaît l'ensemble vide $\emptyset$, l'ensemble $\mathbb{N}$ des entiers naturels et sait qu'une figure géométrique est (ou peut être vue comme) un ensemble de points. Les ensembles sont ainsi perçus comme des boîtes contenant divers objets mathématiques et notre expérience du monde physique est censée nous fournir une intuition suffisante de ce que sont les propriétés de ces boîtes.

Pourtant, s'il est évident que l'ensemble A des multiples de 2 et l'ensemble B des multiples de 3 sont distincts, ces boîtes contiennent toutes deux l'entier 6 (qui ne s'est pas dédoublé lorsqu'on a défini A et B). C'est même la boîte des multiples de 6 qui est entièrement contenue dans les boîtes A et B. On voit ainsi que les frontières de ces boîtes s'interpénètrent, ce qui nous éloigne radicalement du monde physique et laisse notre intuition un peu en déshérence [1]. Sommes-nous alors libres d'imaginer et de définir ces frontières comme bon nous semble ?

C'est ce qu'on a pu croire au XIXème siècle, mais la découverte du paradoxe de Russell au début du XXème siècle a montré que c'était une impasse. Il consiste

1. on utilise aussi les diagrammes de Venn comme support géométrique à l'intuition, mais ils deviennent vite illisibles ; on dépasse rarement le diagramme à quatre ensembles.

à considérer l'ensemble E de tous les ensembles et parmi ceux-ci l'ensemble A des ensembles X qui ne se contiennent pas ($X \notin X$). Par exemple on a $E \in E$ donc $E \notin A$. Mais avons-nous $A \in A$? Si c'est vrai alors $A \notin A$ (comme pour E), donc c'est faux. Mais si c'est faux alors par définition de A on a $A \in A$, donc c'est vrai. On voit donc que $A \in A$ ne peut être ni vrai ni faux ; c'est un paradoxe.

Ce merveilleux outil qui avait permis à Cantor d'élucider les questions que posaient l'infini devait-il être abandonné, victime de la malédiction de Galilée ? En réalité, les travaux de Cantor ne souffraient d'aucune contradiction, il ne restait donc plus qu'à développer une théorie des ensembles assez restreinte pour éviter le paradoxe de Russell (et certains autres), mais assez puissante pour y développer les résultats de Cantor. Ce travail fut accompli avec succès principalement par Ernst Zermelo et Abraham Fraenkel dans les premières décennies du XXème siècle.

Nous ne donnerons pas ici les axiomes de cette théorie ; leur complexité montre à quel point le discours mathématique, dont le but est d'éliminer toute subjectivité, ne peut pas s'appuyer seulement sur l'intuition même lorsqu'il porte sur des objets en apparence aussi simples que les ensembles. Ces axiomes permettent de construire des ensembles comme l'union, l'ensemble des parties ou le produit cartésien, à partir d'ensembles ou d'autres objets mathématiques. Un *ensemble* est alors, par définition, un objet E qui est soit l'ensemble vide $\emptyset$, soit qui contient au moins un élément x (donc tel que $x \in E$). Les éléments d'un ensemble peuvent être ou ne pas être des ensembles.

Bien entendu, si l'on utilise les ensembles avec d'autres objets mathématiques auxquels on confère des propriétés axiomatiques, celles-ci peuvent entrer en contradiction avec les axiomes de la théorie des ensembles. Une façon d'éliminer ce risque est d'utiliser des objets sans propriétés (donc sans axiomes). On peut se demander quel est l'intérêt d'utiliser des objets mathématiques sans rien supposer de leurs propriétés. Pourtant, c'est exactement ce qu'on fait en géométrie, où les points sont des objets sur lesquels ne pèsent que l'hypothèse d'existence [2]. S'agissant d'une existence imaginaire et non physique, c'est bien peu de chose. De même, en théorie des langages on se donne pour vocabulaire un ensemble de lettres qui sont des objets sans propriétés.

Une autre façon d'éliminer le risque de contradictions exogènes est d'interdire toute importation : il suffit d'admettre que tout objet est un ensemble. La théorie qui en résulte peut sembler trop pure pour être féconde, mais il apparaît que les mathématiques y sont entièrement contenues. Plus exactement, tous les objets mathématiques peuvent être construits comme des ensembles de cette théorie pure, et leurs propriétés découlent de cette construction et des axiomes sur les ensembles. Ainsi, un nombre entier peut être représenté par un ensemble, de même qu'un nombre réel, une fonction, une liste, etc. C'est cette construction qu'on a appelée mathématiques modernes ; elle établit que la correction des mathématiques (l'absence de contradic-

2. On peut aussi supposer que ces points sont 2 à 2 distincts, ce qui revient à supposer l'existence propre de chaque point, indépendamment des autres.

tion) ne repose que sur la correction de la théorie des ensembles, qui apparaît donc comme une théorie unificatrice des mathématiques. Cette construction semblait trop abstraite à certains conservateurs qui la combatirent, comme Kronecker, mais elle finit par séduire l'immense majorité des mathématiciens. « Personne ne nous chassera du paradis que Cantor a créé » disait Hilbert.

On peut donc admettre que les points en géométrie ou les lettres en théorie des langages sont des ensembles (ou n'importe quels objets mathématiques, puisque cela revient au même). En tant que tels ils ont des propriétés, mais tant qu'on ne les utilise pas rien n'est modifié dans la géométrie ou les langages. Et si on les utilise tous les résultats standards sont évidemment préservés. On peut ainsi construire un cercle dont les points sont eux-mêmes des cercles sur d'autres points (ce qui pourrait constituer un tore), on peut construire un langage dont les lettres sont des graphes, on peut construire un graphe dont les nœuds sont des langages, etc. Tant que ces constructions restent ensemblistes elles ne peuvent pas créer de contradictions.

C'est sur ce principe qu'est conçu ce cours, en particulier dans le développement (très classique) de la sémantique du premier ordre qui fait appel à des ensembles quelconques. Nous supposons bien entendu qu'ils obéissent aux axiomes de la théorie des ensembles afin d'éviter toute contradiction exogène, mais c'est la seule contrainte et on peut donc admettre qu'ils contiennent des entiers, des nombres réels, des tableaux... puisque ces objets peuvent être considérés (ou construits) comme des ensembles. En logique du premier ordre on peut parler de tout, tant que ce sont des ensembles.

Pré-requis

Nous supposons acquises des notions de base en théorie des langages et en calculabilité, ainsi bien sûr qu'une certaine aisance avec les objets mathématiques tels que les ensembles, les fonctions, les relations, les nombres entiers, etc.

De la théorie des langages nous ne retenons que les notions de *vocabulaire* (n'importe quel ensemble dont les éléments seront considérés comme des lettres [3]), de *mot* (une séquence finie d'éléments d'un vocabulaire) et de *langage* (un ensemble de mots). Nous pourrons concaténer deux mots m et m' pour former le mot mm', nous pourrons aussi substituer dans un mot m toute occurrence d'une lettre l par un mot m' pour obtenir le mot $m[m'/l]$ (m où m' remplace l). Par exemple, sur le vocabulaire $\{a, b, c\}$,

$$abbcab[cb/b] \;=\; acbcbcacb.$$

Ce dernier mot peut également être considéré comme un mot sur le vocabulaire $\{ac, bc, b\}$. Il est donc toujours important de savoir de quel vocabulaire il est question. Par exemple, la langue française écrite peut être vue comme un langage sur le

3. Il est standard en théorie des langages de ne considérer que des vocabulaires *finis*. Ici, un vocabulaire peut être infini.

vocabulaire alphabétique (plus quelques signes comme le tiret, le point, l'espace, etc.) ou comme un langage sur un dictionnaire français dont les entrées sont considérées alors comme des lettres. Autrement dit, puisque tout ensemble peut être considéré comme un vocabulaire, alors un langage peut également être utilisé comme vocabulaire.

Nous utiliserons également une notation moins standard pour remplacer un sous-mot s de m par s'. Mais cela peut être ambigu ; par exemple, remplacer aa par bb dans $aaaa$ pourrait donner $abba$ ou $bbbb$. Nous convenons alors que s fait référence à une occurrence particulière dans m qu'on peut spécifier par un entier : si $m = m_1 s m_2$ où m_1 est un mot de longueur n, alors $m[s'/s]_n = m_1 s' m_2$. On a donc $aaaa[bb/aa]_0 = bbaa$, $aaaa[bb/aa]_1 = abba$ et $aaaa[bb/aa]_2 = aabb$. Mais pour ne pas nous encombrer du paramètre n, nous écrirons $m[s' \to s]$ en convenant implicitement que dans un même contexte nous faisons toujours référence à la même occurrence de s. La flèche suggère que cette occurrence peut également être spécifiée par un pointeur.

Les notions de calculabilité sont généralement définies sur des ensembles d'entiers mais nous les utiliserons sur des langages. Cette généralisation est commune et ne change essentiellement rien à ces notions : on peut l'obtenir soit en utilisant des machines de Turing sur le vocabulaire considéré, soit via une traduction des mots vers les entiers (ce qui suppose des langages au plus dénombrables). Nous dirons donc qu'une fonction partielle[4] $f : L_1 \times \cdots \times L_n \to L$ est *calculable* s'il existe un programme qui calcule $f(m_1, \ldots, m_n)$ en un temps fini à partir des mots $m_1, \ldots, m_n$ lorsque $f(m_1, \ldots, m_n)$ existe, et qui ne termine pas lorsque $f(m_1, \ldots, m_n)$ n'est pas définie. Ce programme peut être une machine de Turing ou un élément de n'importe quel langage de programmation, cela revient exactement au même (du point de vue de l'existence). Une relation sur des langages (ou simplement un langage) sera dite *récursive* ou *décidable* si sa fonction caractéristique est calculable. Une fonction caractéristique est bien sûr totale.

De même, nous dirons qu'une relation $R \subseteq L_1 \times \cdots \times L_n$ est *semi-décidable* s'il existe une fonction partielle calculable $f : L_1 \times \cdots \times L_n \to \{\text{V}, \text{F}\}$ telle que

$$\forall \langle m_1, \ldots, m_n \rangle \in L_1 \times \cdots \times L_n, f(m_1, \ldots, m_n) = \text{V} \text{ ssi } \langle m_1, \ldots, m_n \rangle \in R.$$

Autrement dit, il existe un programme qui termine toujours si la réponse est V et qui donne toujours une réponse correcte s'il termine. Il est donc possible que ce programme ne termine pas si $\langle m_1, \ldots, m_n \rangle \notin R$. Il est évident que toute relation décidable est semi-décidable, mais l'inverse est faux : une relation indécidable peut être semi-décidable, ce qui est mieux que rien.

Enfin, la relation R sera dite *récursivement énumérable* s'il existe une fonction totale calculable $f : \mathbb{N} \to L_1 \times \cdots \times L_n$ telle que $f(\mathbb{N}) = R$. Cette notion est en fait équivalente à la précédente, nous utiliserons l'une ou l'autre indifféremment.

4. Une relation est une fonction partielle s'il existe au plus une image pour chaque antécédent. Une relation est une fonction totale s'il existe exactement une image pour chaque antécédent.

Chapitre 1

Inférences et logiques

1.1 Introduction

L'étude de la Logique a été initiée par Aristote qui cherchait une façon systématique de démasquer les faux raisonnements des sophistes. Ces derniers étaient des orateurs très cultivés et d'une grande éloquence dont le but, d'après Platon, était de convaincre l'auditoire de leurs propos, que ceux-ci soient vrais ou non. Alors que le rôle des philosophes était la recherche de la vérité, celui des sophistes aurait été la recherche de la persuasion. Aristote a donc cherché à distinguer les raisonnements corrects de ceux qui ne l'étaient pas, ce qui l'a amené à définir des *syllogismes* qui sont des exemples de raisonnements corrects classés selon leur forme.

Le but de ce court chapitre est de présenter la généralisation moderne des idées d'Aristote. Nous commençons par la notion de *système d'inférence* qui permet de composer des *règles d'inférence* données (comme les syllogismes) afin d'obtenir les raisonnements admis dans ce système. Nous aborderons ensuite la notion de *logique*, qui permet d'attribuer un sens aux mots utilisés afin de juger de la correction des raisonnements. La séparation de ces deux aspects du raisonnement et leur interaction est fondamentale dans l'approche moderne de la logique.

1.2 Systèmes d'inférence

Le syllogisme le plus connu peut s'énoncer de la façon suivante : à partir de propositions de la forme

> `tout ce qui est` A `est` B
> C `est` A

on peut en déduire la proposition

> C `est` B,

où A et B dénotent des adjectifs et C dénote un groupe nominal quelconque ; c'est le syllogisme *barbara*. Par exemple, selon ce syllogisme, à partir de

`tout ce qui est humain est mortel`
`Socrate est humain`

on peut déduire

`Socrate est mortel`.

Ce raisonnement élémentaire nous semble évidemment correct, mais le syllogisme *barbara* appliqué aux propositions suivantes

`tout ce qui est rare est cher`
`un cheval pas cher est rare`

permet de déduire que

`un cheval pas cher est cher`,

ce qui nous semble évidemment incorrect. Mais l'évidence est trompeuse et comme indiqué ci-dessus nous aborderons la difficile question de la correction ultérieurement. Nous ne nous attachons ici qu'à la forme purement syntaxique des raisonnements.

Une première remarque est que les adjectifs et groupes nominaux A, B et C ne sont limités que par le langage utilisé et donc que l'ensemble des raisonnements qui relèvent du syllogisme *barbara* est potentiellement *infini*. Malgré cela, il est aisé de savoir si un raisonnement donné correspond ou non au syllogisme *barbara*. C'est le cas des deux exemples précédents, ce n'est pas le cas du suivant [1] :

`tout ce qui est humain est mortel`
`aucun dieu n'est mortel`
donc `aucun dieu n'est humain,`

Cette propriété essentielle se traduit mathématiquement grâce aux notions de calculabilité.

Définition 1.1 Soient V un ensemble au plus dénombrable et $\mathcal{F}$ un langage récursif non vide sur le vocabulaire V, dont les éléments sont appelés *formules*. On appelle *règle d'inférence sur* $\mathcal{F}$ toute relation récursive $R \subseteq \mathcal{F}^* \times \mathcal{F}$, où $\mathcal{F}^* = \bigcup_{n\in\mathbb{N}} \mathcal{F}^n$.

On appelle *inférence par* R tout élément $\langle p_1, \ldots, p_n, c\rangle \in R$; on dit alors que les p_i sont les *prémisses* [2] et que c est la *conclusion* de cette inférence. On utilise la notation suivante pour représenter cette inférence :

$$\frac{p_1 \cdots p_n}{c}\, R. \qquad \triangle$$

On peut donc écrire le syllogisme *barbara* comme une règle d'inférence à deux prémisses :

$$\frac{\texttt{tout ce qui est } A \texttt{ est } B \quad C \texttt{ est } A}{C \texttt{ est } B}\ \textit{barbara}$$

1. qui correspond à un autre syllogisme, dit *camestres*. Les logiciens scolastiques en ont dénombré une bonne vingtaine.

2. A ne pas confondre avec les *prémices*, premières manifestations d'un phénomène.

Le fait d'imposer que $\mathcal{F}$ soit un langage récursif signifie simplement qu'un programme doit être capable de reconnaître si une suite de lettres est une formule ou non. Concrètement, l'ensemble $\mathcal{F}$ est souvent défini par induction ou bien à l'aide d'une grammaire hors-contexte, de telle sorte à ce que ses éléments soient reconnus facilement. Cela signifie aussi que nous ne pourront utiliser la langue française sinon dans des versions extrêmement simplifiées.

Une règle d'inférence étant un ensemble récursif d'inférences, il existe donc un programme qui permet de savoir si une suite finie de formules est une inférence appartenant à cette règle. En général, ces inférences partagent une forme simple à reconnaître et peuvent donc être regroupées dans une même règle au moyen d'un « schéma » utilisant des « variables » A, B, C comme nous l'avons fait pour le syllogisme *barbara* ; dans ce cas le programme de reconnaissance des inférences est évident. Un tel schéma permet donc de définir un ensemble infini d'inférences qui est récursif. Un exemple connu est le *modus ponens* MP, définit par le schéma

$$\frac{A \qquad \texttt{si } A \texttt{ alors } B}{B}\ \text{MP}.$$

Cela signifie que la règle d'inférence MP est l'ensemble des triplets $\{\langle A, \texttt{si } A \texttt{ alors } B,\ B\rangle \mid A, B \in \mathcal{F}\}$, qui contient par exemple[3] les inférences suivantes (en supposant que ces formules sont dans $\mathcal{F}$) :

$$\frac{1+1=2 \qquad \texttt{si } 1+1=2 \texttt{ alors } 2H+O \to H_2O}{2H+O \to H_2O}$$

$$\frac{0=1 \qquad \texttt{si } 0=1 \texttt{ alors je suis le pape}}{\texttt{je suis le pape}}$$

Il sera parfois nécessaire d'utiliser des notions plus subtiles pour définir des règles d'inférence, par exemple en rajoutant des *conditions* sur les variables (comme dans le syllogisme *barbara*). Bien entendu, nous n'avons aucune raison de limiter les raisonnements à une seule règle d'inférence, même si elle contient une infinité d'inférences. Nous avons en général besoin d'utiliser des raisonnements de différentes formes et donc d'admettre plusieurs règles d'inférence.

Définition 1.2 Un *système d'inférence* est un triplet $\mathcal{S} = \langle V, \mathcal{F}, \mathcal{R}\rangle$, où $\mathcal{F}$ est un langage récursif sur le vocabulaire au plus dénombrable V, et $\mathcal{R}$ est un ensemble fini de règles d'inférence sur $\mathcal{F}$.

3. Un jour que Bertrand Russell expliquait à ses interlocuteurs qu'à partir de l'hypothèse $0 = 1$ on pouvait montrer n'importe quoi, l'un d'eux le mit au défi d'en déduire qu'il était le pape. Il y parvint par le raisonnement suivant : *si* $0 = 1$ *alors* $0 + 1 = 1 + 1$, *donc* $1 = 2$, *or le pape et moi sommes 2 personnes, donc le pape et moi sommes 1 personne, donc je suis le pape.*

Toute inférence par une règle dans $\mathcal{R}$ est appelée *inférence dans* $\mathcal{S}$. On appelle *axiome de* $\mathcal{S}$ toute formule a telle que $\exists R \in \mathcal{R}$, où $a \in R$. Autrement dit, (a) est une inférence sans prémisse, qui peut donc s'écrire

$$\frac{}{a}\ R.$$

△

Mais le raisonnement ne serait pas grand chose s'il se limitait aux inférences. Ce qui en fait la richesse, c'est la possibilité de combiner plusieurs inférences afin d'obtenir, par un subtil agencement, des raisonnements qui ne paraissent pas aussi évidents que les inférences qui les composent.

L'agencement se fait par l'utilisation des conclusions comme prémisses de nouvelles inférences. Par exemple, on peut faire suivre deux inférences *barbara* :

1 - `tout ce qui est humain est mortel`
2 - `Socrate est humain`
donc par 1 et 2 :
3 - `Socrate est mortel`
4 - `tout ce qui est mortel est vivant`
donc par 4 et 3 :
5 - `Socrate est vivant`.

Comme nous ne disposons pas toujours d'axiomes, les raisonnements doivent pouvoir être construits à partir de formules données : les *hypothèses*. Dans le raisonnement précédent, les hypothèses sont les formules 1, 2 et 4. La formule 3 est la conclusion de la première inférence et une prémisse de la seconde.

Les agencements que nous pouvons faire sont libres, avec cependant une restriction très importante : nous n'autorisons pas les raisonnements circulaires. La raison en est simplement que cela permettrait de faire passer une hypothèse pour une conclusion. Par exemple :

1 - `tout ce qui est vivant est mortel`
donc par 1 et 4 :
2 - `Le buste de Socrate est mortel`
3 - `tout ce qui est mortel est vivant`
donc par 3 et 2 :
4 - `Le buste de Socrate est vivant`.

Donc en n'utilisant que les hypothèses 1 et 3, qui sont vraies, on serait capable de déduire une formule évidemment fausse. Nous devons donc admettre que nos raisonnements doivent être ordonnés ; nous les appellerons *déductions*.

Définition 1.3 Une $\mathcal{S}$*-déduction à partir de* $H \subseteq \mathcal{F}$ est une séquence *finie* non vide de formules $(f_i)_{i=1}^n$ (pour $n \in \mathbb{N}$) telle que pour tout $1 \leq i \leq n$ au moins une des conditions suivantes est vérifiée :

— $f_i \in H$,
— il existe m prémisses $f_{i_1}, \ldots, f_{i_m}$ (où $m \in \mathbb{N}$) telles que $i_1 < i, \ldots, i_m < i$ et $(f_{i_1}, \ldots, f_{i_m}, f_i)$ est une inférence dans $\mathcal{S}$.

Cette déduction est *de longueur* n, les éléments de H sont les *hypothèses* de la déduction et f_n en est la *conclusion*.

Pour $f \in \mathcal{F}$, on dit que f *est une conséquence de* H *dans* $\mathcal{S}$ (ou que f *se déduit de* H *dans* $\mathcal{S}$) et on note $H \vdash_{\mathcal{S}} f$ s'il existe une $\mathcal{S}$-déduction $(f_i)_{i=1}^n$ à partir de H qui contient f (i.e. $\exists i$ tel que $f_i = f$). Pour $C \subseteq \mathcal{F}$, on écrit $H \vdash_{\mathcal{S}} C$ si pour toute formule $f \in C$ on a $H \vdash_{\mathcal{S}} f$.

Une $\mathcal{S}$*-preuve* est une $\mathcal{S}$-déduction à partir de $\emptyset$ (sans hypothèses). On écrit $\vdash_{\mathcal{S}} f$ pour $\emptyset \vdash_{\mathcal{S}} f$; on dit alors que f est un $\mathcal{S}$*-théorème*. Donc $\vdash_{\mathcal{S}} C$ signifie que C est un ensemble de théorèmes. △

Bien entendu, il ne peut y avoir de théorèmes s'il n'y a pas d'axiomes. De plus, tout axiome est un théorème et tout théorème est une conséquence de n'importe quelles hypothèses.

EXEMPLE 1.4 On considère le vocabulaire $V = \{1, +, =\}$ et l'ensemble de formules $\mathcal{F} = \left\{1^i + 1^j = 1^k \mid i, j, k \in \mathbb{N}\right\}$, où 1^n est la dénotation standard de n concaténation du symbole '1'. L'ensemble $\mathcal{R}$ contient une règle dont les éléments sont de la forme :

$$\frac{1^i + 1^j = 1^k \qquad 1^l + 1^m = 1^n}{1^{i+l} + 1^{j+m} = 1^{k+n}}.$$

Posons $H = \{1 + 1 = 11\}$, alors la formule $1111 + 1111 = 11111111$ est une conséquence de H dans $\mathcal{S}$. En effet, la suite $(1 + 1 = 11, 11 + 11 = 1111, 1111 + 1111 = 11111111)$ est une déduction à partir de H : la première formule est l'hypothèse, la deuxième est obtenue par inférence en utilisant deux fois la première comme prémisse, et la dernière est obtenue en utilisant deux fois la deuxième.

Soit $\mathcal{S}' = \langle V, \mathcal{F}, \mathcal{R}' \rangle$, où $\mathcal{R}'$ contient les mêmes règles que $\mathcal{R}$, ainsi que l'axiome suivant :

$$\frac{}{11 + 111 = 1}.$$

Alors la formule $1111 + 111111 = 11$ est un $\mathcal{S}'$-théorème (en utilisant deux fois l'axiome). ◊

Les propriétés suivantes sont des conséquences simples de la définition d'un système d'inférence, mais elle sont fondamentales :

Théorème 1.5

1. *L'ensemble des preuves est récursif.*
2. *L'ensemble des théorèmes est récursivement énumérable.*

PREUVE. Le fait que l'ensemble des preuves soit récursif est évident : par définition d'un système d'inférence, $\mathcal{F}$ est récursif et $\mathcal{R}$ est un ensemble fini de relations récursives. Il est donc possible de construire un programme prenant en entrée une séquence de mots, qui teste si cette entrée correspond bien à une séquence valide de formules et s'assure que cette séquence est effectivement une déduction en testant que chaque formule est conclusion par une règle d'inférence de prémisses qui la précède dans la suite (il faut essayer toutes les règles et toutes les séquences de prémisses parmi un ensemble fini). Un tel programme produirait en sortie « oui » si la séquence est effectivement une preuve et « non » si ce n'est pas le cas, et il terminerait toujours.

Pour décider si une formule f est un théorème, il suffit de construire un programme qui énumère toutes les preuves valides et qui teste si f apparaît dans l'une d'elles. Comme l'ensemble des suites finies de formules est dénombrable, on peut facilement calculer ses éléments $s_0, s_1, s_2 \ldots$ les uns après les autres, et on utilise le programme précédent pour décider si s_i est une preuve. Si non, on passe à s_{i+1} ; si oui, on teste d'abord si f apparaît dans s_i et si c'est le cas on a terminé.

Si f est un théorème alors il existe $i \in \mathbb{N}$ tel que s_i est une preuve de f, donc le programme termine. Si f n'est pas un théorème, ce programme ne termine pas. •

Définition 1.6 Soit $H \subseteq \mathcal{F}$ un ensemble récursif, on note $\mathcal{S}_H$ le système d'inférence suivant : $\mathcal{S}_H = \langle V, \mathcal{F}, \mathcal{R} \cup \{H\} \rangle$ (on ajoute les éléments de H comme axiomes de $\mathcal{S}_H$). △

Si l'hypothèse « H est récursif » avait été omise, $\mathcal{S}_H$ n'aurait pas nécessairement été un système d'inférence, puisque la condition de récursivité des règles d'inférence aurait pu être violée.

Proposition 1.7 *Étant donné un ensemble H récursif, l'ensemble des déductions à partir de H dans $\mathcal{S}$ est l'ensemble des preuves dans $\mathcal{S}_H$.*

PREUVE. On prouve par induction sur n que toute déduction $(f_1, \ldots, f_n)$ à partir de H dans $\mathcal{S}$ est une preuve dans $\mathcal{S}_H$. Supposons le résultat vrai pour toute déduction de longueur au plus $n-1$, alors la déduction $(f_1, \ldots, f_{n-1})$ est une preuve dans $\mathcal{S}_H$ par hypothèse. Considérons la formule f_n, deux cas sont possibles.

1. Si $f_n \in H$, alors f_n est un axiome dans $\mathcal{S}_H$ par construction.
2. S'il existe m prémisses $f_{i_1}, \ldots, f_{i_m}$ telles que $i_1 < i, \ldots, i_m < n$ et $(f_{i_1}, \ldots, f_{i_m}, f_n)$ est une inférence dans $\mathcal{S}$, alors c'est également une inférence dans $\mathcal{S}_H$ par construction.

Ainsi, dans tous les cas, $(f_1, \ldots, f_n)$ est bien une preuve dans $\mathcal{S}_H$.

Réciproquement, toute preuve $(f_1, \ldots, f_n)$ dans $\mathcal{S}_H$ est une déduction à partir de H dans $\mathcal{S}$. La preuve procède par induction sur n, comme ci-dessus. •

Le Théorème 1.5 et la Proposition 1.7 ont pour conséquence immédiate le résultat suivant :

Corollaire 1.8 *L'ensemble des conséquences de H dans $\mathcal{S}$ est l'ensemble des théorèmes de $\mathcal{S}_H$. Donc, si H est récursif alors $\{f \in \mathcal{F} \mid H \vdash_{\mathcal{S}} f\}$ est récursivement énumérable.*

Définition 1.9 H est *$\mathcal{S}$-consistant* si $\exists f \in \mathcal{F}$ tel que $H \nvdash_{\mathcal{S}} f$. $\mathcal{S}$ est *consistant* si $\emptyset$ est $\mathcal{S}$-consistant. △

Un système d'inférence inconsistant est donc un système dans lequel on peut prouver tout et n'importe quoi, ce qui limite beaucoup l'intérêt des preuves.

1.3 Logiques

Jusqu'ici les déductions ne sont qu'un jeu de construction qui ne signifie rien, sinon que des formules peuvent être construites à partir d'autres formules. Ce jeu est encore bien éloigné de la recherche de la vérité chère à Aristote et donc de ce que nous qualifions habituellement de raisonnement correct.

Intuitivement, un raisonnement correct serait une déduction dont la conclusion est nécessairement vraie. Mais qu'entendons-nous précisément par *vraie* ? Est-il vrai que Socrate est vivant, comme nous l'avons déduit plus haut ? En réalité, cela dépend du contexte. Il ne s'agit pas seulement d'être un contemporain de Socrate ; il faut aussi savoir qui est Socrate et s'il possède la qualité d'être vivant.

Autrement dit, nous devons admettre que le sens de certains mots est indéterminé, ou au moins variable dans une certaine mesure, et que ce n'est que dans un contexte que ces sens se déterminent et que les formules deviennent vraies ou fausses. Pour simplifier, nous allons ignorer les mots et leurs sens en supposant une relation directe entre les contextes et la véracité des formules. En logique, les contextes sont appelés des *interprétations*.

Définition 1.10 Une *logique* est un quadruplet $\mathcal{L} = \langle V, \mathcal{F}, \mathcal{I}, \models_{\mathcal{L}} \rangle$, où
- $\mathcal{F}$ est un langage récursif sur le vocabulaire V, ses éléments sont appelés *formules*,
- $\mathcal{I}$ est un ensemble dont les éléments sont appelés *interprétations*,
- $\models_{\mathcal{L}}$ est une relation binaire dans $\mathcal{I} \times \mathcal{F}$, dite *relation de satisfaction*. Pour $I \in \mathcal{I}$ et $f \in \mathcal{F}$, il est standard de noter $I \models_{\mathcal{L}} f$ pour signifier que le couple (I, f) est élément de la relation $\models_{\mathcal{L}}$.

Lorsque $I \models_{\mathcal{L}} f$ pour $I \in \mathcal{I}$ et $f \in \mathcal{F}$, nous dirons que f est *vraie dans* I, que I *satisfait* f et que I est un *modèle de* f. Lorsque $I \not\models_{\mathcal{L}} f$ nous dirons que f est *fausse dans* I, que I *ne satisfait pas* f et que I est un *contre-modèle de* f.

On étend $\models_{\mathcal{L}}$ aux ensembles de formules de la façon suivante : pour $E \subseteq \mathcal{F}$ et $I \in \mathcal{I}$, on note $I \models_{\mathcal{L}} E$ si $I \models_{\mathcal{L}} f$ pour tout $f \in E$. En particulier, $I \models_{\mathcal{L}} \{f\}$ si et seulement si $I \models_{\mathcal{L}} f$.

On appelle *système d'inférence pour* $\mathcal{L}$ tout système $\mathcal{S} = \langle V, \mathcal{F}, \mathcal{R} \rangle$ sur le même langage de formules. △

Remarque. En général il n'y aura pas d'ambiguïté sur la logique considérée et on pourra noter $\models$ à la place de $\models_{\mathcal{L}}$.

Ainsi, on peut dire que dans le contexte de la Grèce antique, tel que nous le connaissons, la formule `Socrate est humain` est vraie. Il existe cependant d'autres interprétations où cette formule est fausse ; on a pu imaginer par exemple que Socrate ne serait qu'un personnage inventé par Platon.

Remarque. Il faut noter que les interprétations sont (en général) décrites mathématiquement au moyen d'ensembles qui peuvent ne pas être finis, ni même dénombrables (voir l'Exercice 13). Il n'y a donc pas de sens à supposer des propriétés de calculabilité concernant la relation de satisfaction. C'est une différence essentielle entre une logique et un système d'inférence.

Il reste à faire le lien avec les systèmes d'inférence. Pour qu'une inférence constitue un raisonnement correct, suffit-il que sa conclusion soit vraie dans une interprétation particulière ? Considérons par exemple l'inférence suivante :

`tout ce qui est humain est mortel`
`Socrate est humain`
donc `Socrate est un philosophe`.

La conclusion est vraie dans l'acception courante, de même que les prémisses, et pourtant on ne peut qualifier ce raisonnement de correct. Pourquoi ? Le problème est que la conclusion devient fausse si on interprète le nom de `Socrate` comme celui d'un footballeur brésilien, qui est bien un homme mais non un philosophe. Par contre, les deux Socrate sont bien mortels (du moins l'étaient-ils de leur vivant) ; la conclusion `Socrate est mortel` est donc correcte.

On voit ici que la notion de raisonnement correct serait bien difficile à concevoir sans variabilité des interprétations. Elle seule permet d'exiger une forme de robustesse de la conclusion vis-à-vis de la diversité des interprétations que l'on peut faire des prémisses. Nous pouvons maintenant définir précisément cette robustesse.

Définition 1.11 Soient $\mathcal{L} = \langle V, \mathcal{F}, \mathcal{I}, \models_{\mathcal{L}} \rangle$ une logique, $c \in \mathcal{F}$ et $H \subseteq \mathcal{F}$, on dit que c est une *conséquence logique* de H, et on note $H \models_{\mathcal{L}} c$, si pour toute interprétation $I \in \mathcal{I}$, $I \models_{\mathcal{L}} H$ entraîne $I \models_{\mathcal{L}} c$. Si $\mathcal{S}$ est un système d'inférence pour $\mathcal{L}$, une déduction $H \vdash_{\mathcal{S}} c$ est dite *correcte pour* $\mathcal{L}$ si $H \models_{\mathcal{L}} c$. On dit que $\mathcal{S}$ est *correct pour* $\mathcal{L}$ si toute inférence dans $\mathcal{S}$ est correcte pour $\mathcal{L}$.

Soit C un ensemble de formules de $\mathcal{F}$, on note $H \models_{\mathcal{L}} C$ si $H \models_{\mathcal{L}} c$ pour tout $c \in C$. Soit $h \in \mathcal{F}$, on note $h \models_{\mathcal{L}} c$ pour $\{h\} \models_{\mathcal{L}} c$. On note également $\models_{\mathcal{L}} c$ pour $\emptyset \models_{\mathcal{L}} c$ et on dit alors que c est *valide dans* $\mathcal{L}$. De même on note $\models_{\mathcal{L}} C$ pour $\emptyset \models_{\mathcal{L}} C$.

Lorsque $h \models_{\mathcal{L}} c$ et $c \models_{\mathcal{L}} h$, on dit que h et c sont *logiquement équivalents* et on note $h \equiv_{\mathcal{L}} c$. Cette définition s'étend aux ensembles de formules : deux ensembles de formules C et D sont *logiquement équivalents*, et on note $C \equiv_{\mathcal{L}} D$, si $C \models_{\mathcal{L}} D$ et $D \models_{\mathcal{L}} C$. △

EXERCICE 1. *Soit $\mathcal{L}$ une logique, et A, B, C, D des ensembles de formules. Les énoncés suivants sont-ils vrais ou faux ? Donnez une preuve des énoncés vrais et un contre-exemple des énoncés faux.*

1. *Si $A \models_{\mathcal{L}} C \cup D$ alors $A \models_{\mathcal{L}} C$.*
2. *Si $A \models_{\mathcal{L}} C$ alors $A \models_{\mathcal{L}} C \cup D$.*
3. *Si $A \models_{\mathcal{L}} C$ alors $A \cup B \models_{\mathcal{L}} C$.*
4. *Si $A \models_{\mathcal{L}} C$ et $B \models_{\mathcal{L}} D$ alors $A \cup B \models_{\mathcal{L}} C \cup D$.*
5. *Si $A \models_{\mathcal{L}} B$ et $B \models_{\mathcal{L}} C$ alors $A \models_{\mathcal{L}} C$.*

Par définition, une formule est donc valide si elle est vraie dans toute interprétation, ce que nous pouvons considérer comme une vérité universelle (dans le cadre de $\mathcal{L}$). Ainsi, la formule `tout ce qui est humain est mortel` est valide dans l'ensemble des interprétations raisonnables de la langue française. C'est hélas une vérité universelle.

Si $\mathcal{S}$ est correct pour $\mathcal{L}$, il est évident que tout axiome de $\mathcal{S}$ est valide. Cette propriété se généralise aux théorèmes.

Proposition 1.12 *Si $\mathcal{S}$ est correct pour $\mathcal{L}$ alors tout théorème de $\mathcal{S}$ est valide pour $\mathcal{L}$.*

PREUVE. On montre par induction sur la longueur des preuves dans $\mathcal{S}$ que si $(f_1, \ldots, f_n)$ est une preuve dans $\mathcal{S}$ alors f_n est valide dans $\mathcal{L}$. Supposons le résultat vrai pour toute preuve de longueur au plus $n-1$, et considérons la preuve $(f_1, \ldots, f_n)$. Par définition, il existe $m \geq 0$ prémisses $f_{i_1}, \ldots, f_{i_m}$ telles que $0 \leq i_1 < n, \ldots, i_m < n$ et $(f_{i_1}, \ldots, f_{i_m}, f_n)$ est une inférence dans $\mathcal{S}$. Si $m = 0$ alors f_n doit être un axiome de $\mathcal{S}$ et est valide par hypothèse. Sinon, par hypothèse d'induction, les formules $f_{i_1}, \ldots, f_{i_m}$ sont valides. Comme $\mathcal{S}$ et correct pour $\mathcal{L}$, on a $\{f_{i_1}, \ldots, f_{i_m}\} \models_{\mathcal{L}} f_n$, ce qui prouve que f_n est également valide. •

Dans ce cas, les théorèmes sont donc bien des vérités universelles. Nous pouvons cependant généraliser ce résultat.

Proposition 1.13 *Si $\mathcal{S}$ est correct pour $\mathcal{L}$ alors toute déduction dans $\mathcal{S}$ est correcte pour $\mathcal{L}$.*

EXERCICE 2. *Démontrer la Proposition 1.13.*

Il suffit donc que les règles d'inférence soient correctes pour que les raisonnements le soient. Par exemple, le syllogisme *barbara* est (intuitivement) correct et donc tous les raisonnements qui en découlent doivent également l'être. Pourtant, nous sommes parvenus à déduire une formule contradictoire : `un cheval pas cher est cher`. Plus précisément :

Définition 1.14 Une formule $f \in \mathcal{F}$ (resp. un ensemble E de formules) est *satisfaisable* s'il existe $I \in \mathcal{I}$ tel que $I \models_{\mathcal{L}} f$ (resp. $I \models_{\mathcal{L}} E$). Sinon, f (resp. E) est *insatisfaisable*. △

Une formule contradictoire est donc une formule insatisfaisable. Mais comment un raisonnement correct peut-il mener à une contradiction ? La réponse se trouve dans l'exercice suivant :

EXERCICE 3. *Montrez que si $H \models_{\mathcal{L}} c$ et c est insatisfaisable, alors H est insatisfaisable.*

Nous pouvons donc en conclure que l'ensemble

$$\{\texttt{tout ce qui est rare est cher}, \texttt{un cheval pas cher est rare}\}$$

est insatisfaisable. Ces deux hypothèses sont contradictoires, elles ne peuvent être vraies simultanément (dans la même interprétation). Nous l'avons prouvé par un procédé proche de la preuve par l'absurde [4]. Mais la preuve par l'absurde nécessite une négation.

Définition 1.15 Une logique $\mathcal{L}$ *admet une négation* s'il existe un symbole $\neg \in V$ tel que $\neg f$ est une formule si et seulement si f en est une, et $I \models_{\mathcal{L}} \neg f$ si et seulement si $I \not\models_{\mathcal{L}} f$ pour tout $I \in \mathcal{I}$. △

Proposition 1.16 *Si $\mathcal{L}$ a une négation $\neg$, alors toute formule $f \in \mathcal{F}$ est insatisfaisable si et seulement si $\neg f$ est valide.*

EXERCICE 4. *Démontrer la Proposition 1.16.*

On peut également relier cette notion à celle de conséquence logique.

Proposition 1.17 *Si $\mathcal{L}$ a une négation $\neg$, pour tout $H \subseteq \mathcal{F}$ et $c \in \mathcal{F}$, on a $H \models_{\mathcal{L}} c$ si et seulement si $H \cup \{\neg c\}$ est insatisfaisable.*

EXERCICE 5. *Démontrer la Proposition 1.17*

C'est ce procédé qu'on appelle communément *preuve par l'absurde* ; on prouve c en montrant qu'une contradiction découle de $\neg c$. On peut en conclure que :

$$\texttt{tout ce qui est rare est cher} \models \texttt{un cheval pas cher n'est pas rare},$$

mais également

$$\texttt{un cheval pas cher est rare} \models \texttt{tout ce qui est rare n'est pas cher},$$

4. Ce genre d'absurdité peut justifier l'expression anglaise qui qualifie l'économie de *dismal science*...

ce qui correspond mieux à la réalité. Il est facile de trouver des objets rares qui n'ont pas la moindre valeur.

Une autre façon de voir la correction est de dire que l'ensemble des théorèmes est inclus dans l'ensemble des formules valides, ou qu'à partir de n'importe quelles hypothèses l'ensemble des formules qui s'en déduisent en sont des conséquences logiques. Un système d'inférence correct permet donc d'explorer les vérités universelles ainsi que les conséquences logiques d'un ensemble d'hypothèses H, c'est à dire les vérités qui ne sont universelles que dans l'ensemble des modèles de H. Malheureusement, cette exploration n'est pas nécessairement exhaustive ; l'ensemble des théorèmes peut être plus petit que l'ensemble des formules valides. Dans certaines logiques $\mathcal{L}$, il se peut même que dans tout système d'inférence $\mathcal{S}$ correct pour $\mathcal{L}$, il existe une formule f valide dans $\mathcal{L}$ qui n'est pas un théorème de $\mathcal{S}$.

Définition 1.18 $\mathcal{S}$ est *complet pour* $\mathcal{L}$ si toutes les formules valides pour $\mathcal{L}$ sont des théorèmes de $\mathcal{S}$. $\mathcal{S}$ est *fortement complet pour* $\mathcal{L}$ si pour tout $H \subseteq \mathcal{F}$, toute conséquence logique de H se déduit de H dans S.

Une logique qui n'admet pas de système d'inférence correct et complet est dite *incomplète*. △

La complétude est une conséquence de la complétude forte et la réciproque est souvent facile à établir (grâce à certaines propriétés de $\mathcal{L}$). Pour une logique donnée, on veut évidemment trouver un système d'inférence correct et complet s'il en existe un, car cela permet de réduire la validité à la prouvabilité. Mais la complétude d'un système d'inférence est souvent bien plus difficile à établir que sa correction. À plus forte raison l'incomplétude d'une logique est souvent très difficile à prouver, mais on peut utiliser le résultat suivant, qui illustre à nouveau la proximité de la logique avec l'informatique.

Proposition 1.19 *Si l'ensemble des formules valides de $\mathcal{L}$ n'est pas récursivement énumérable, alors $\mathcal{L}$ est incomplète.*

Exercice 6. *Démontrer la Proposition 1.19.*

1.4 Exercices supplémentaires

Exercice 7. *Montrez que si $\mathcal{L}$ a une négation et que $\mathcal{S}$ est un système d'inférence correct pour $\mathcal{L}$, alors $\mathcal{S}$ est consistant.*

EXERCICE 8. *On se place sur le vocabulaire $V = \{\circ, \bullet\}$ et on considère le langage $\mathcal{F}$ des mots de 9 lettres sur V. Pour plus de lisibilité, on écrit ces formules en carré, i.e., le mot $a_1a_2a_3a_4a_5a_6a_7a_8a_9$ s'écrit $\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}$. On note $\mathbf{0} = \begin{smallmatrix} \circ\circ\circ \\ \circ\circ\circ \\ \circ\circ\circ \end{smallmatrix}$. On définit une complémentation sur V, par $\bar{\circ} = \bullet$ et $\bar{\bullet} = \circ$, et on utilisera les règles d'inférences suivantes :*

$$\frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} \bar{a}_1\bar{a}_2\bar{a}_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{L}_1 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} a_1\bar{a}_2a_3 \\ \bar{a}_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{L}_2 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} a_1a_2a_3 \\ \bar{a}_4\bar{a}_5\bar{a}_6 \\ \bar{a}_7\bar{a}_8\bar{a}_9 \end{smallmatrix}}\,\mathrm{L}_3 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} \bar{a}_1a_2a_3 \\ a_4\bar{a}_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{D}_1$$

$$\frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} \bar{a}_1a_2a_3 \\ \bar{a}_4a_5a_6 \\ \bar{a}_7a_8a_9 \end{smallmatrix}}\,\mathrm{C}_1 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} a_1\bar{a}_2a_3 \\ a_4\bar{a}_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{C}_2 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} a_1a_2\bar{a}_3 \\ a_4a_5\bar{a}_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{C}_3 \qquad \frac{\begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}{\begin{smallmatrix} a_1\bar{a}_2\bar{a}_3 \\ a_4\bar{a}_5a_6 \\ a_7a_8a_9 \end{smallmatrix}}\,\mathrm{D}_2.$$

On considère donc le système d'inférence $\mathcal{S} = \langle V, \mathcal{F}, \{\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3, \mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \mathrm{D}_1, \mathrm{D}_2\}\rangle$.

▷ **Question 1** *Pouvez-vous prouver $\mathbf{0} \vdash_{\mathcal{S}} \begin{smallmatrix} \circ\circ\circ \\ \circ\bullet\circ \\ \circ\circ\circ \end{smallmatrix}$? Pouvez-vous prouver $\mathbf{0} \vdash_{\mathcal{S}} \begin{smallmatrix} \bullet\circ\circ \\ \circ\circ\circ \\ \circ\circ\circ \end{smallmatrix}$?*

On associe un entier à chaque élément de V : $|\circ| = 0$ et $|\bullet| = 1$. Soient $I = \{0, 1\}$ et $\models_1$ la relation sur $I \times \mathcal{F}$ définie par

$$i \models_1 \begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix} \text{ si et seulement si } \; |a_1| + |a_3| + |a_7| + |a_9| \equiv i[2].$$

On considère la logique $\mathcal{L}_1 = \langle V, \mathcal{F}, I, \models_1\rangle$.

▷ **Question 2** *Le système $\mathcal{S}_{\{\mathbf{0}\}}$ est-il consistant ?*

Enfin, on considère la relation $\models_2$ sur $I \times \mathcal{F}$ définie par

$$i \models_2 \begin{smallmatrix} a_1a_2a_3 \\ a_4a_5a_6 \\ a_7a_8a_9 \end{smallmatrix} \text{ si et seulement si } \; |a_1| + |a_2| + |a_4| + |a_6| + |a_8| + |a_9| \equiv i[2]$$

et la logique $\mathcal{L}_2 = \langle V, \mathcal{F}, I, \models_2\rangle$.

▷ **Question 3** *Le système $\mathcal{S}$ est-il fortement complet pour $\mathcal{L}_1$?*

Remarque : dans les logiques $\mathcal{L}_1$ et $\mathcal{L}_2$ les formules n'ont qu'une seule interprétation possible ; il n'y a donc pas de formules valides. Ce sont des exemples très particuliers de logiques, qu'on pourrait qualifier de cas dégénérés.

EXERCICE 9. *On souhaite prouver qu'il existe deux nombres irrationnels a, b tels que a^b est rationnel. On admet que $\sqrt{2}$ est irrationnel.*

▷ **Question 1** *Démontrer le résultat en considérant le cas où le nombre $\sqrt{2}^{\sqrt{2}}$ est rationnel et le cas où il ne l'est pas.*

La preuve donnée ci-dessus n'est pas valide pour les intuitionnistes*, pour lesquels on ne peut prouver l'existence d'un objet qu'en le* construisant*.*

▷ **Question 2** *Démontrer que $\log_2(9)$ n'est pas rationnel, et en déduire une preuve constructive de la propriété.*

EXERCICE 10. *Étant donnés deux langages L, M sur un vocabulaire V, on définit le langage quotient à droite de L par M, noté L/M, de la façon suivante :*

$$L/M \;=\; \{w \in V^* \mid \exists w' \in M : ww' \in L\}\,.$$

▷ **Question 1** *Supposons que L est un langage régulier, et soit $A = (Q, V, \delta, I, F)$ un automate reconnaissant L. Comment définir F' pour que (Q, V, δ, I, F') reconnaisse L/M ?*

▷ **Question 2** *En déduire que si L est régulier, alors L/M est régulier.*

▷ **Question 3** *Étant donné un mot $w \in \{0,1\}^*$ et un nombre r, on définit $N(w, r)$ comme le nombre d'occurrences de la chaîne w dans le développement binaire de r. Par exemple, si $w = 01$ et $r = \frac{45}{8}$, alors $N(w, r) = 2$ (car le développement binaire de $\frac{45}{8}$ est $101,101$). On définit les langages suivants :*

$$\begin{aligned} L &= \{w \in \{0,1\}^* \mid w \text{ contient au moins une occurrence de la chaîne } 00\}\,, \\ M &= \{w \in \{0,1\}^* \mid N(w\pi) \text{ est fini}\}\,. \end{aligned}$$

Construire un automate reconnaissant L et en déduire un automate reconnaissant L/M. Quel est le problème ?

Chapitre 2

Logique propositionnelle

La logique propositionnelle peut être considérée comme la plus simple des logiques, ce qui ne la rend pas moins importante. Elle provient d'une analyse systématique de la forme des propositions plutôt que de la forme des raisonnements sur ces propositions, analyse qui ayant été négligée par Aristote fut initiée quelques décennies plus tard par les stoïciens. Elle repose sur une décomposition des propositions selon des *connecteurs* (la conjonction `et`, la disjonction `ou`, la conditionnelle `si alors`), jusqu'à obtenir des propositions atomiques. On peut alors définir des raisonnements tel le *modus ponens*, typique de l'école stoïcienne.

Cette analyse très superficielle ne permet pas *a priori* de faire de distinctions entre des notions grammaticales de base, comme les substantifs et les adjectifs. Que ce soit par mépris de cette simplicité ou par vénération du génie d'Aristote, cette approche fut délaissée au profit de la syllogistique jusqu'au XIXème siècle et en particulier jusqu'aux travaux de George Boole. C'est alors qu'on réalisa, grâce à l'analyse géométrique des syllogismes par les *diagrammes d'Euler* (ultérieurement perfectionnés par Venn), que les connecteurs de la logique stoïcienne étaient suffisants pour rendre compte des syllogismes.

La simplicité de la logique propositionnelle (ou booléenne) n'est donc aucunement un défaut ; elle provient d'abord de sa très grande généralité qui lui permettra non seulement de mettre un terme à deux millénaires de syllogistique, mais en fera au XXème siècle l'un des outils fondamentaux de l'informatique.

2.1 Définitions

Définition 2.1 (Logique propositionnelle) Étant donné un ensemble au plus dénombrable $\mathcal{S}$ de *symboles propositionnels*, on considère le vocabulaire

$$\mathrm{V_p}(\mathcal{S}) \;=\; \mathcal{S} \uplus \{\blacksquare, \square, \vee, \wedge, \neg, \Rightarrow, \Leftrightarrow, \text{“(”}, \text{“)”}\}\,.$$

L'ensemble $\mathrm{P}(\mathcal{S})$ de formules considéré est construit de façon inductive : $\mathrm{P}(\mathcal{S})$ est le plus petit ensemble tel que

— $\mathcal{S} \subseteq \mathrm{P}(\mathcal{S})$,
— $\blacksquare \in \mathrm{P}(\mathcal{S})$ et $\square \in \mathrm{P}(\mathcal{S})$,
— pour tout $f, f' \in \mathrm{P}(\mathcal{S})$, $\{\neg f,\ (f \vee f'),\ (f \wedge f'),\ (f \Rightarrow f'),\ (f \Leftrightarrow f')\} \subseteq \mathrm{P}(\mathcal{S})$.

On définit $\mathrm{I_p}(\mathcal{S})$ comme l'ensemble des fonctions I de $\mathcal{S}$ dans l'ensemble $\{\mathrm{V}, \mathrm{F}\}$ des *valeurs de vérité*. La relation de satisfaction $\models_0$ est définie pour tout $I \in \mathrm{I_p}(\mathcal{S})$ et tout $f \in \mathrm{P}(\mathcal{S})$ de façon inductive :

— $I \models_0 \blacksquare$ et $I \not\models_0 \square$,
— $I \models_0 p$ si et seulement si $I(p) = \mathrm{V}$, pour tout $p \in \mathcal{S}$,
— $I \models_0 \neg f$ si et seulement si $I \not\models_0 f$,
— $I \models_0 f \vee f'$ si et seulement si $I \models_0 f$ ou $I \models_0 f'$,
— $I \models_0 f \wedge f'$ si et seulement si $I \models_0 f$ et $I \models_0 f'$,
— $I \models_0 f \Rightarrow f'$ si et seulement si $I \models_0 f$ implique $I \models_0 f'$,
— $I \models_0 f \Leftrightarrow f'$ si et seulement si on a à la fois $I \models_0 f \Rightarrow f'$ et $I \models_0 f' \Rightarrow f$.

On appelle *logique propositionnelle* la logique $\mathrm{L_p}(\mathcal{S}) = \langle \mathrm{V_p}(\mathcal{S}), \mathrm{P}(\mathcal{S}), \mathrm{I_p}(\mathcal{S}), \models_0 \rangle$. La relation d'équivalence logique relative à $\models_0$ sera notée $\equiv_0$. △

Remarque. Dans la suite nous nous autorisons pour des raisons de lisibilité à omettre certaines parenthèses, voire à les remplacer par des crochets, accolades ou tout autre délimiteur agréé par l'Académie. S'il n'y a pas d'ambiguïté sur l'ensemble de symboles propositionnels $\mathcal{S}$ considéré, nous pourrons l'omettre des notations, et par exemple considérer l'ensemble d'interprétations $\mathrm{I_p}$ ou l'ensemble de formules P.

Les symboles propositionnels sont les « mots » dont l'interprétation est variable. Leur interprétation ne peut varier que dans l'ensemble $\{\mathrm{V}, \mathrm{F}\}$ puisque ces symboles représentent des propositions atomiques ; on peut donc dire que la variabilité est minimale. Par exemple, on peut choisir

$$\mathcal{S} = \{\texttt{Socrate est humain}, \texttt{Socrate est mortel}\}$$

ce qui permet de construire la formule

$$\texttt{Socrate est humain} \Rightarrow \texttt{Socrate est mortel}$$

dont la véracité dépend de celle des symboles propositionnels qui la composent.

La logique propositionnelle étant indicée par l'ensemble $\mathcal{S}$, on doit considérer qu'on définit en réalité une *famille* de logiques. Les autres symboles de ces langages (négation, implication, etc.) ont une interprétation qui est fixée par la définition de la relation de satisfaction et ne dépend pas de $\mathcal{S}$; elle est donc *commune* à toutes les logiques de cette famille. C'est pourquoi on considère les membres de cette famille de logiques comme des variantes d'une même « logique propositionnelle ».

On peut cependant être dubitatif sur la définition donnée du sens des connecteurs. Ainsi, le connecteur $\vee$ est interprété par la conjonction de coordination *ou*. Cela peut sembler redondant, mais revient simplement à supposer le lecteur capable de

comprendre le sens de ce mot. Il serait difficile de définir mathématiquement une logique (ou n'importe quoi d'autre) à un lecteur dépourvu de tout sens logique.

Il est cependant possible d'être plus précis, ce qui serait nécessaire si on avait utilisé des connecteurs sans équivalents directs dans la langue française. Il suffit pour cela de définir l'une après l'autre les valeurs de vérité des formules en fonction des valeurs prises par les formules connectées dans une interprétation ; on dit que la *valeur* de f dans I est V (resp. F) si $I \models_0 f$ (resp. si $I \not\models_0 f$).

f	f'	$\Box$	$\blacksquare$	$\neg f$	$f \vee f'$	$f \wedge f'$	$f \Rightarrow f'$	$f \Leftrightarrow f'$
V	V	F	V	F	V	V	V	V
V	F	F	V	F	V	F	F	F
F	V	F	V	V	V	F	V	F
F	F	F	V	V	F	F	V	V

TABLE 2.1 – Sémantique des connecteurs

On voit que, comme attendu, la valeur de vérité de $\Box$ et $\blacksquare$ ne dépend pas de celles de f et f', et la valeur de vérité de $\neg f$ ne dépend pas de celle de f'. Ainsi, on a par exemple $I \models_0 f \wedge f'$ si et seulement si la valeur de f et de f' dans I est V, autrement dit, si $I(f) = I(f') = \text{V}$, en notant $I(f)$ la valeur de vérité de f dans I. Cette notation est déjà utilisée pour les symboles propositionnels, mais elle est acceptable puisque $I(p)$ est bien la valeur de vérité de la formule p pour tout $p \in \mathcal{S}$.

En utilisant le Tableau 2.1, on peut facilement évaluer la valeur de vérité d'une formule dans une interprétation. Par exemple, si $I(p) = I(q) = \text{V}$, alors $I(\neg p) = \text{F}$ et $I(p \Rightarrow q) = \text{V}$, donc $I(\neg p \vee q) = \text{V}$, donc $I((\neg p \vee q) \Leftrightarrow (p \Rightarrow q)) = \text{V}$ et ainsi de suite. Et comme $I((\neg p \vee q) \Leftrightarrow (p \Rightarrow q))$ ne dépend que de $I(p)$ et $I(q)$, on peut même en calculer toutes les valeurs possibles.

p	q	$\neg p$	$\neg p \vee q$	$p \Rightarrow q$	$(\neg p \vee q) \Leftrightarrow (p \Rightarrow q)$
V	V	F	V	V	V
V	F	F	F	F	V
F	V	V	V	V	V
F	F	V	V	V	V

TABLE 2.2 – Table de vérité de $(\neg p \vee q) \Leftrightarrow (p \Rightarrow q)$

Le Tableau 2.2 est la *table de vérité* de la formule $(\neg p \vee q) \Leftrightarrow (p \Rightarrow q)$. On y voit que sa valeur de vérité est V pour tout I, et donc que $\models_0 (\neg p \vee q) \Leftrightarrow (p \Rightarrow q)$;

c'est une formule valide. On peut de même construire la table de vérité de n'importe quelle formule f, mais il est évident que le nombre de lignes est 2^n si f contient n symboles propositionnels distincts.

EXERCICE 11. *Donner les tables de vérité des formules suivantes :*

1. $(p \Rightarrow \Box) \Leftrightarrow \neg p$
2. $(p \Rightarrow q) \wedge (q \Rightarrow p)$
3. $((p \Rightarrow q) \wedge (q \Rightarrow r)) \Rightarrow (p \Rightarrow r)$

On remarque que $\models_0 f \Leftrightarrow f'$ si et seulement si $I(f) = I(f')$ pour toute interprétation I, ce qui est équivalent à $f \equiv_0 f'$. Les tables de vérité permettent donc de prouver que deux formules sont logiquement équivalentes. L'équivalence logique admet une propriété fondamentale.

Théorème 2.2 *Soient $g \in \mathrm{P}(\mathcal{S})$, f une sous-formule de g et $f' \in \mathrm{P}(\mathcal{S})$ telle que $f \equiv_0 f'$, si g' est obtenue de g en remplaçant f par f' alors $g \equiv_0 g'$.*

PREUVE. Si $f = g$ alors $g' = f'$ et le résultat est évident. Sinon on procède par induction sur la construction de g.

Si $g = \neg h$ alors f est une sous-formule de h, $g' = \neg h'$ et f' est une sous-formule de h'. Par hypothèse d'induction on a $h \equiv_0 h'$, donc $g \equiv_0 g'$.

De même si $g = h_1 \ \omega \ h_2$ avec $\omega \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$, alors f est une sous-formule de h_i ($i = 1$ ou $i = 2$), donc $g' = h'_1 \ \omega \ h'_2$ (où $h'_j = h_j$ pour $j \neq i$) et f' est une sous-formule de h'_i. Par hypothèse d'induction on a $h_i \equiv_0 h'_i$, donc $g \equiv_0 g'$. •

Cela signifie qu'on peut voir l'équivalence logique comme une relation d'égalité algébrique, qui permet donc de remplacer des égaux par des égaux. Cette approche algébrique de la logique a été initiée par George Boole qui espérait ainsi remplacer la pensée par le calcul. On peut en effet utiliser un certain nombre d'identités afin de simplifier une formule, comme l'idempotence $f \wedge f \equiv_0 f$, les éléments neutres $f \wedge \blacksquare \equiv_0 f$, les éléments absorbants $f \wedge \Box \equiv_0 \Box$ et bien d'autres.

Mais ces identités sont-elles suffisantes pour réduire toute formule valide à $\blacksquare$? Nous avons vu dans le Tableau 2.2 que $\neg p \vee q \equiv_0 p \Rightarrow q$, mais faut-il remplacer systématiquement $\neg f \vee g$ par $f \Rightarrow g$, ou l'inverse ? Dans ce cas on pourrait remplacer les implications par des disjonctions, en rajoutant des négations, mais est-ce une simplification de la formule ?

Tout dépend de la stratégie adoptée pour arriver au résultat ; il ne peut y avoir une méthode unique car la conjonction et la disjonction jouent des rôles parfaitement symétriques. Une première approche serait effectivement d'éliminer les connecteurs que l'on peut exprimer à l'aide d'autres connecteurs grâce à des identités adéquates.

Exercice 12. *Montrer que toute formule de* $\mathrm{L_p}(\mathcal{S})$ *est équivalente à une formule construite à l'aide des ensembles restreints de connecteurs suivants :*

— $\neg$ *et* $\vee$,
— $\neg$ *et* $\wedge$,
— $\neg$ *et* $\Rightarrow$,
— $\square$ *et* $\Rightarrow$,
— $\mid$ *(l'opérateur NAND*[1]*).*

Les identités utilisées peuvent être établies grâce à des tables de vérité ou à partir d'autres identités. On n'oubliera pas que $\neg$, $\blacksquare$ *et* $\square$ *sont aussi des connecteurs. On supposera que* $\mathcal{S} \neq \emptyset$.

Cependant, les formules obtenues par ces éliminations de connecteurs ne sont pas nécessairement plus simples que celles d'où elles proviennent ; c'est en particulier vrai si on n'utilise que l'opérateur NAND. Mais ces transformations peuvent être utiles pour la réalisation de circuits électriques au moyen d'une technologie ne permettant de réaliser que certains connecteurs (ou « portes ») logiques.

L'utilisation de l'algèbre booléenne pour la conception de circuits électriques a été initiée par Claude Shannon ; elle mènera avec les travaux d'Alan Turing à une révolution technologique majeure, la réalisation d'ordinateurs qui marquent l'entrée dans l'ère numérique. On voit ainsi que des considérations très abstraites portant sur la pensée et le calcul peuvent être dévoyées en applications concrètes, voire même en profits.

Exemple 2.3 (Lien avec les circuits électriques) Il est nécessaire en électronique numérique de concevoir des circuits qui calculent des *fonctions logiques*. Ces fonctions prennent en entrée et produisent en sortie des paramètres qui ne peuvent avoir que deux valeurs possibles (marche/arrêt, vrai/faux, ouvert/fermé...). En composant des *portes logiques* qui représentent des opérateurs logiques, il est possible de concevoir des circuits calculant n'importe quelle fonction logique. Parmi les portes logiques les plus standard, on retrouve celles de la Figure 2.1.

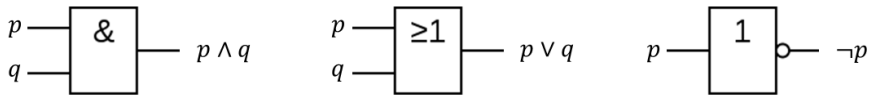

Figure 2.1 – Les portes logiques 'et', 'ou' et 'non'

Pour se fixer les idées, on peut supposer que les valeurs possibles des paramètres sont 'le courant passe' et 'le courant ne passe pas'. Ainsi, pour la porte 'et', il y a du courant en sortie si et seulement s'il y a du courant dans les deux entrées. La

1. Où la notation $p \mid q$ est équivalente à $\neg(p \wedge q)$.

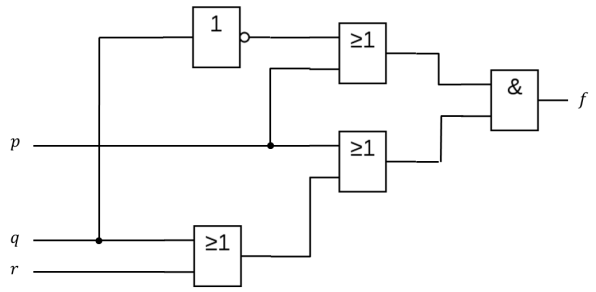

FIGURE 2.2 – Un circuit avec trois entrées

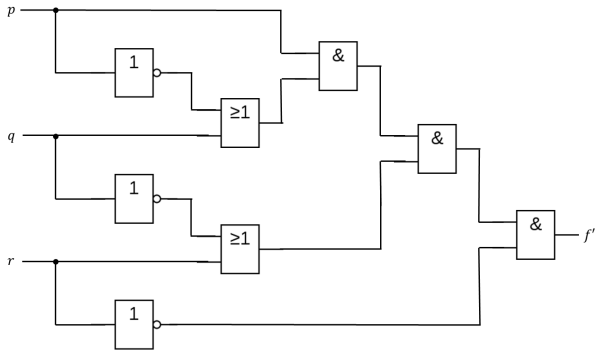

FIGURE 2.3 – Un second circuit avec trois entrées

traduction logique des différentes portes est immédiate : à chaque entrée correspond un symbole propositionnel, et en sortie de porte, on obtient une formule correspondant à la combinaison de ces symboles par l'opérateur logique de cette porte. Une interprétation I désigne les entrées par lesquelles passe un courant, et si la sortie d'un circuit donné correspond à une formule logique f, alors $I \models_0 f$ si et seulement s'il y a du courant à la sortie du circuit.

Considérons le circuit de la Figure 2.2, dont les entrées sont dénotées par p, q, r. On lui associe la logique $\mathrm{L_p}(\mathcal{S})$, où $\mathcal{S} = \{p, q, r\}$; l'ensemble $\mathrm{I_p}(\mathcal{S})$ contient donc 8 interprétations. La présence de courant en sortie est représentée par la formule logique $f = (p \vee q \vee r) \wedge (p \vee \neg q)$.

Notons I_1 l'interprétation qui interprète p, q comme étant vraies et r comme étant fausse : $I_1(p) = I_1(q) = \mathrm{V}$ et $I_1(r) = \mathrm{F}$. Selon notre convention, cette interprétation correspond au cas où un courant passe dans les entrées p et q, mais pas dans l'entrée r. On peut dire dans ce cas qu'un courant passe également à la sortie, puisque $I_1 \models_0 f$. Par contre, notons I_2 l'interprétation qui interprète q comme étant vraie et p, r comme étant fausses. On a alors $I_2 \not\models_0 f$; le courant ne passe pas.

Considérons maintenant le circuit de la Figure 2.3, dont la sortie correspond à la formule

$$f' \;=\; p \wedge (\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r).$$

La formule f' est insatisfaisable, ce qui signifie que quelles que soient les entrées par lesquelles passe un courant, il ne pourra jamais passer de courant par la sortie. $\Diamond$

EXERCICE 13. *Supposons que l'ensemble de symboles propositionnels $\mathcal{S}$ soit infini dénombrable. Montrer que $\mathrm{I_p}(\mathcal{S})$ est alors indénombrable.*

2.2 Un système d'inférence

Nous avons vu dans l'Exercice 12 qu'on peut simplifier toute formule en une formule ne contenant que les connecteurs $\Rightarrow$ et $\neg$. Nous pouvons donc proposer un système d'inférence spécialisé pour ces formules.

Pour des raisons de lisibilité, nous supposons maintenant fixé un ensemble infini de symboles propositionnels $\mathcal{S}$. Nous noterons donc P_1 pour $\mathrm{P}(\mathcal{S})_1$, $\mathrm{L_{p1}}$ pour $\mathrm{L_p}(\mathcal{S})_1$, et ainsi de suite.

Définition 2.4 On définit le système d'inférence $\mathcal{S}_1 = \langle \mathrm{V_p}, \mathrm{P}_1, \mathcal{R}_1 \rangle$, où $\mathcal{R}_1 = \{\mathrm{MP}, \mathrm{K}, \mathrm{S}, \mathrm{N}\}$ est constitué des règles suivantes [2] :

$$\frac{A \quad A \Rightarrow B}{B}\mathrm{MP} \qquad \frac{}{(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))}\mathrm{S}$$

$$\frac{}{A \Rightarrow (B \Rightarrow A)}\mathrm{K} \qquad \frac{}{(\neg B \Rightarrow \neg A) \Rightarrow ((\neg B \Rightarrow A) \Rightarrow B)}\mathrm{N}$$

On reconnaît dans la règle MP le *modus ponens* reformulé en logique propositionnelle. $\triangle$

Proposition 2.5 *$\mathcal{S}_1$ est correct pour $\mathrm{L_{p1}}$.*

EXERCICE 14. *Démontrer cette proposition.*

Théorème 2.6 (Admis) *$\mathcal{S}_1$ est complet pour $\mathrm{L_{p1}}$.*

Ceci signifie que toute formule valide de $\mathrm{L_{p1}}$ est un théorème de $\mathcal{S}_1$ et peut donc être engendré à partir des axiomes et du *modus ponens*.

2. les symboles A, B et C dénotent des formules propositionnelles simplifiées.

EXERCICE 15. *Soit $p \in \mathcal{S}$, montrer que $p \Rightarrow p$ est un théorème de $\mathcal{S}_1$.*

L'Exercice 15 illustre bien la difficulté qu'il peut y avoir à prouver des résultats pourtant évidents. Ainsi, même si le Théorème 2.6 garantit que $\mathcal{S}_1$ peut être utilisé pour tester si une formule donnée est valide ou non, l'intérêt de ce système d'inférence peut sembler limité [3], notamment quand on compare la difficulté de trouver la preuve de $p \Rightarrow p$ à la simplicité de vérifier la validité de cette formule par sa table de vérité.

2.3 Forme clausale

Exprimer une formule en n'utilisant qu'un nombre minimal de connecteurs n'est donc pas un gage de simplicité (voir l'Exercice 12). Nous devons tenir compte de la difficulté qu'il y a à *prouver* une formule, et nous devons reconnaître que tous les systèmes formels, même corrects et complets, ne se valent pas sous cet angle. Dans cette section nous allons transformer les formules propositionnelles en *ensembles* de formules extrêmement simples ; ces formules sont appelées des *clauses*.

Dans une première étape, on élimine les connecteurs $\Leftrightarrow$ et $\Rightarrow$ de la façon déjà décrite en exercice, c'est à dire en utilisant les identités suivantes :

$$f \Rightarrow g \equiv_0 \neg f \vee g \qquad f \Leftrightarrow g \equiv_0 (\neg f \vee g) \wedge (\neg g \vee f)$$

de la gauche vers la droite. Afin d'éviter des répétitions inutiles, nous notons l'orientation adoptée au moyen de flèches :

$$f \Rightarrow g \to \neg f \vee g \qquad f \Leftrightarrow g \to (\neg f \vee g) \wedge (\neg g \vee f).$$

On voit que l'élimination de $\Leftrightarrow$ nécessite une duplication des sous-formules f et g.

EXERCICE 16. *Appliquer cette transformation aux formules*

$$p_1 \Leftrightarrow (p_2 \Leftrightarrow \cdots (p_{n-1} \Leftrightarrow p_n) \cdots),$$

pour tout $n \in \mathbb{N}^*$. Que peut on en conclure ?

Il ne reste donc que les connecteurs $\vee$, $\wedge$, $\neg$, $\blacksquare$ et $\square$. Leur avantage sur $\Rightarrow$ et $\Leftrightarrow$ est qu'ils possèdent des propriétés particulièrement utiles. Les premières que nous utiliserons sont les identités de De Morgan, ainsi que des propriétés simples de la négation :

$$\neg(f \vee g) \to \neg f \wedge \neg g \quad \neg(f \wedge g) \to \neg f \vee \neg g \quad \neg(\neg f) \to f \quad \neg\blacksquare \to \square \quad \neg\square \to \blacksquare.$$

3. Ce système d'inférence semble limité si le but est seulement de prouver des formules. Il a en fait d'autres applications.

EXERCICE 17. *Appliquer cette transformation à la formule suivante :*

$$\neg[(p \vee \neg q) \wedge (\neg(q \wedge r) \vee (r \wedge \blacksquare))].$$

Il est clair que cette transformation nous permet de restreindre les négations aux symboles propositionnels.

Définition 2.7 On appelle *littéral* toute formule de la forme p ou $\neg p$, où $p \in \mathcal{S}$. Les littéraux p et $\neg p$ sont dit *complémentaires* ; p est un littéral *positif*, et $\neg p$ est un littéral *négatif*. Le littéral complémentaire de p est $\overline{p} \stackrel{\text{def}}{=} \neg p$, et le littéral complémentaire de $\neg p$ est $\overline{\neg p} \stackrel{\text{def}}{=} p$. △

Nous obtenons donc une formule composée de $\vee$, $\wedge$, $\blacksquare$, $\square$ et de littéraux. Les connecteurs $\vee$ et $\wedge$ peuvent cependant alterner sans aucune limitation. Nous allons maintenant « aplatir » la formule afin de ne conserver qu'un seul niveau d'alternance. Pour cela, nous utilisons la distributivité de $\vee$ sur $\wedge$:

$$(f \wedge g) \vee h \to (f \vee h) \wedge (g \vee h) \qquad h \vee (f \wedge g) \to (h \vee f) \wedge (h \vee g).$$

On voit ainsi que toutes les conjonctions peuvent être « remontées » au-dessus des disjonctions, ou que les disjonctions peuvent être « descendues » sous les conjonctions, ce qui revient au même. Mais comme avec $\Leftrightarrow$, cela nécessite de dupliquer la sous-formule h.

Dans la suite, afin de simplifier les notations nous utiliserons l'associativité de $\vee$ et $\wedge$:

$$(f \vee g) \vee h \equiv_0 f \vee (g \vee h) \qquad (f \wedge g) \wedge h \equiv_0 f \wedge (g \wedge h),$$

pour omettre des parenthèses. Ainsi nous écrirons $p \vee q \vee r \vee s$ pour les formules $(p \vee (q \vee r)) \vee s$, $((p \vee q) \vee r) \vee s$, $p \vee (q \vee (r \vee s))$, $p \vee ((q \vee r) \vee s)$ et $(p \vee q) \vee (r \vee s)$.

EXERCICE 18. *Appliquer la distributivité aux formules* $(p_1 \wedge q_1) \vee \cdots \vee (p_n \wedge q_n)$ *pour* $n \in \mathbb{N}^*$. *Que peut on en conclure ?*

Enfin, nous allons éliminer presque entièrement le connecteur $\blacksquare$ grâce aux règles suivantes :

$$f \vee \blacksquare \to \blacksquare \qquad \blacksquare \vee f \to \blacksquare \qquad f \wedge \blacksquare \to f \qquad \blacksquare \wedge f \to f.$$

EXERCICE 19. *Appliquer l'élimination de* $\blacksquare$ *à la formule* $(p \vee \blacksquare) \wedge (\blacksquare \vee q \vee \square)$.

Après cette transformation il ne peut rester au plus qu'un seul connecteur $\blacksquare$, et dans ce cas c'est tout ce qui reste de la formule, qui est donc valide. Il faut remarquer que la formule $p \vee \neg p$ est également valide, donc toutes les formules valides ne sont pas transformées en $\blacksquare$.

Définition 2.8 On appelle *clause* toute formule ne contenant que des littéraux et les connecteurs $\vee$ et $\Box$. On appelle *forme normale conjonctive* (ou *fnc*) toute conjonction d'un nombre fini de clauses, ainsi que la formule $\blacksquare$. $\triangle$

EXEMPLE 2.9 Les formules $\Box$, $\neg p$, $\Box \vee \Box$, $p \vee \neg q \vee \Box$ et $\Box \vee p \vee q \vee \neg p \vee q$ sont des clauses, donc également des fnc. Les formules $\blacksquare$, $p \wedge q$, $\neg p \wedge \Box \wedge \neg p$, $(p \vee q) \wedge (p \vee q) \wedge p$ sont des fnc mais pas des clauses. $\Diamond$

Nous avons donc vu que toute formule peut être transformée en une fnc équivalente. On aurait pu, en appliquant la distributivité de $\wedge$ sur $\vee$ et des règles d'élimination de $\Box$, obtenir de même une forme normale disjonctive équivalente. Si nous avons choisi la fnc, c'est pour une raison simple : une conjonction de formules est logiquement équivalente à l'ensemble de ces formules.

EXERCICE 20. *Montrer que pour toute fnc $f = C_1 \wedge \cdots \wedge C_n$, où les C_i sont des clauses et $n \in \mathbb{N}$, on a $f \equiv_0 \{C_1, \ldots, C_n\}$. Remarque : si $n = 0$ la notation $C_1 \wedge \cdots \wedge C_n$ (ou $\bigwedge_{i=1}^{n} C_i$) s'interprète par l'élément neutre de l'opérateur $\wedge$, qui est bien sûr $\blacksquare$. De même que la notation $\{C_1, \ldots, C_n\}$ (ou $\bigcup_{i=1}^{n} \{C_i\}$) s'interprète alors par l'ensemble vide $\emptyset$, élément neutre de $\cup$.*

Définition 2.10 Si $C_1 \wedge \cdots \wedge C_n$ est une fnc obtenue par transformation d'une formule f, alors l'ensemble $\{C_1, \ldots, C_n\}$ est une *forme clausale* de f. $\triangle$

Grâce à cette définition, nous pourrons nous contenter de règles d'inférence sur les clauses. Pour cela, nous utiliserons les notions suivantes :

Définition 2.11 On définit l'ensemble Lit des littéraux apparaissant dans une clause de la façon suivante :

$$\begin{aligned} \mathrm{Lit}(\Box) &= \emptyset, \\ \mathrm{Lit}(l) &= \{l\} \text{ pour tout littéral } l, \\ \mathrm{Lit}(C \vee C') &= \mathrm{Lit}(C) \cup \mathrm{Lit}(C') \text{ pour toutes clauses } C \text{ et } C'. \end{aligned}$$

La *longueur* d'une clause C, notée $|C|$, est le cardinal de l'ensemble $\mathrm{Lit}(C)$. Une clause C est *vide* si $|C| = 0$, et C est *unitaire* si $|C| = 1$. Si D est une clause telle que $\mathrm{Lit}(C) \subseteq \mathrm{Lit}(D)$, on dit que C *subsume* D.

On étend la définition de Lit à des ensembles de clauses : si E est un ensemble de clauses alors $\mathrm{Lit}(E) = \bigcup_{C \in E} \mathrm{Lit}(C)$. $\triangle$

Remarque. (Une source fréquente d'erreurs) Les *clauses* vides ($\Box$, $\Box \vee \Box$, $\Box \vee \Box \vee \Box \ldots$) sont insatisfaisables, alors que la *forme clausale* $\emptyset$ est valide.

Il est facile de voir que seules les clauses vides sont insatisfaisables. En effet, on a $I \models_0 C$ si et seulement si $\exists l \in \mathrm{Lit}(C)$ tel que $I \models_0 l$, par définition de la disjonction, et on peut toujours trouver I telle que $I \models_0 l$ en choisissant $I(p) = \mathrm{V}$ si $l = p \in \mathcal{S}$ et $I(p) = \mathrm{F}$ si $l = \neg p$. On peut également caractériser les clauses valides.

Théorème 2.12 *Une clause C est valide si et seulement si* $\mathrm{Lit}(C)$ *contient deux littéraux complémentaires.*

PREUVE. S'il existe $p \in \mathcal{S}$ tel que $\{p, \neg p\} \subseteq \mathrm{Lit}(C)$, alors pour toute interprétation I on a soit $I(p) = \text{V}$, auquel cas on a $I \models_0 C$, soit $I(p) = \text{F}$ et on a également $I \models_0 C$; donc $\models_0 C$.

S'il n'existe pas de littéraux complémentaires dans $\mathrm{Lit}(C)$, alors il existe une interprétation I telle que pour tout $l \in \mathrm{Lit}(C)$, on a $I \not\models_0 l$. Donc $I \not\models_0 C$, et donc $\not\models_0 C$. •

Il est également simple de caractériser la conséquence logique entre clauses.

Théorème 2.13 *Soient C une clause et D une clause non valide, on a $C \models_0 D$ si et seulement si C subsume D.*

PREUVE. Si $\mathrm{Lit}(C) \subseteq \mathrm{Lit}(D)$ et $I \models_0 C$, alors $\exists l \in \mathrm{Lit}(C)$ tel que $I \models_0 l$, mais $l \in \mathrm{Lit}(D)$, donc $I \models_0 D$, ce qui prouve que $C \models_0 D$.

Si $\mathrm{Lit}(C) \not\subseteq \mathrm{Lit}(D)$ alors $\exists l \in \mathrm{Lit}(C)$ tel que $l \notin \mathrm{Lit}(D)$. Comme D n'est pas valide, il existe I telle que $I \not\models_0 D$. Si $I \models_0 l$ alors $I \models_0 C$ donc $C \not\models_0 D$. Sinon on a $I \models_0 \bar{l}$, donc $\bar{l} \notin \mathrm{Lit}(D)$. Soit $p \in \mathcal{S}$ le symbole propositionnel apparaissant dans l, on construit l'interprétation J de la façon suivante :

— pour tout $q \in \mathcal{S} \setminus \{p\}$, $J(q) = I(q)$,

— $J(q) = \text{F}$ si $I(q) = \text{V}$, et $J(q) = \text{V}$ sinon.

On a alors $J \models_0 l$, donc $J \models_0 C$. De plus, on a $J(D) = I(D)$ puisque p n'apparaît pas dans D, donc $J \not\models_0 D$, donc $C \not\models_0 D$. •

Cela implique évidemment que si C subsume une clause quelconque D, alors $C \models_0 D$ puisque cela est vrai également lorsque D est valide.

EXERCICE 21. *Les énoncés suivants sont-ils vrais ou faux ? S'ils sont vrais, le démontrer, et s'ils sont faux, donner un contre-exemple.*

1. *Si $C \models_0 D$ alors $C \models_0 D \vee D'$.*
2. *Si $C \models_0 D$ alors $C \vee C' \models_0 D$.*
3. *Si $C \models_0 D$ et $C' \models_0 D'$ alors $C \vee C' \models_0 D \vee D'$.*

Le raisonnement sur les clauses est donc particulièrement simple grâce aux propriétés démontrées ci-dessus. De fait, nous pouvons conclure de ce qui précède que le calcul de la fnc permet toujours de prouver qu'une formule est valide. Il est en effet évident qu'une fnc est valide si et seulement si toutes ses clauses le sont, par définition de la conjonction. Il suffit donc de rajouter

$$C \vee l \vee C' \vee \bar{l} \vee C'' \rightarrow \blacksquare$$

aux règles d'élimination de $\blacksquare$ pour que toute formule valide soit ainsi transformée en $\blacksquare$. Cela réalise donc le souhait de George Boole et permet de remplacer le raisonnement propositionnel (et syllogistique) par de simples règles de calcul.

EXERCICE 22. *Prouver par le calcul que les axiomes K, S et N du système d'inférence $\mathcal{S}_1$ sont valides.*

2.4 Renommage

Comme nous l'avons vu dans les exercices, la transformation qui permet de réduire toute formule valide à $\blacksquare$ ou de calculer une forme clausale est exponentielle dans le pire des cas, à cause de l'élimination de $\Leftrightarrow$ et surtout à cause de la distributivité qui peut être très coûteuse (et difficile) à utiliser.

Est-il possible d'éviter le pire des cas et de ne transformer que des formules dont la forme clausale reste bornée polynomialement par la longueur de la formule initiale ? La réponse est positive à condition d'abandonner la stricte utilisation des identités algébriques. Le procédé utilisé s'apparente au changement de variables.

Définition 2.14 Soient $\mathcal{S} \subset \mathcal{S}'$, $f \in \mathrm{P}(\mathcal{S})$, g une sous-formule de f et $p \in \mathcal{S}' \setminus \mathcal{S}$, on appelle *renommée de f en g par p* toute formule de la forme $f' \wedge (p \Leftrightarrow g)$ où $f' \in \mathrm{P}(\mathcal{S}')$ telle que $f'[g/p] = f$. En absence de précision sur l'ensemble $\mathcal{S}'$, il sera noté $\mathcal{S} + p$, ce qui indique que $p \notin \mathcal{S}$. △

EXEMPLE 2.15 Soient $f = (p \wedge q) \vee (\neg p \wedge \neg q)$ et $g = p \wedge q$, alors la formule

$$(r \vee (\neg p \wedge \neg q)) \wedge (r \Leftrightarrow (p \wedge q))$$

est une renommée de f en g par r. ◊

On utilise donc un nouveau symbole propositionnel qu'on affirme équivalent à une sous-formule, ce qui permet de la remplacer par ce symbole. La formule ainsi obtenue étant une conjonction, sa forme clausale est l'union de celle de f' et de celle de $p \Leftrightarrow g$ qui peuvent donc être calculées séparément. Mais elle ne peut être strictement équivalente à f puisqu'elle dépend de ce nouveau symbole. Nous utiliserons la notion suivante.

Définition 2.16 Soient $\mathcal{S} \subset \mathcal{S}'$, $E \subseteq \mathrm{P}(\mathcal{S})$ et $E' \subseteq \mathrm{P}(\mathcal{S}')$, on dit que *les modèles de E sont extensibles à E'* et on note $E \models_0^{\mathcal{S}} E'$ si pour tout $I \in \mathrm{I_p}(\mathcal{S})$ tel que $I \models_0 E$, il existe $I' \in \mathrm{I_p}(\mathcal{S}')$ tel que $I' \models_0 E'$ et $I'|_{\mathcal{S}} = I$. △

EXEMPLE 2.17 Si $\mathcal{S} = \{p\}$, alors $p \models_0^{\mathcal{S}} \neg p \vee q$ et $p \models_0^{\mathcal{S}} p \wedge \neg q$, mais $p \not\models_0^{\mathcal{S}} \neg p \wedge q$. ◊

Cette notion permet d'établir un lien entre deux formules qui est proche de l'équivalence logique.

Lemme 2.18 *Soient $f \in \mathrm{P}(\mathcal{S})$, $p \notin \mathcal{S}$, g une sous-formule de f et h une renommée de f en g par p, alors $h \models_0 f$ et $f \models_0^{\mathcal{S}} h$.*

PREUVE. Soit $f' \in \mathrm{P}(\mathcal{S}+p)$ telle que $h = f' \wedge (p \Leftrightarrow g)$ et $f'[g/p] = f$.

Pour tout $I' \in \mathrm{I_p}(\mathcal{S}+p)$ tel que $I' \models_0 h$, on a $I' \models_0 p \Leftrightarrow g$, donc $I'(p) = I'(g)$, et $I' \models_0 f'$, donc $I' \models_0 f'[g/p]$ (par une preuve similaire à celle du Théorème 2.2). On a donc $h \models_0 f$.

Pour tout $I \in \mathrm{P}(\mathcal{S})$ tel que $I \models_0 f$, soit $I' \in \mathrm{P}(\mathcal{S}+f)$ tel que $I'|_{\mathcal{S}} = I$ et $I'(p) = I(g)$, on a donc $I' \models_0 p \Leftrightarrow g$ et $I' \models_0 f'[g/p]$, mais $I'(g) = I(g)$ puisque $g \in \mathrm{P}(\mathcal{S})$, donc $I'(g) = I'(p)$ et donc $I' \models_0 f'$, ce qui prouve que $f \models_0^{\mathcal{S}} h$. •

Corollaire 2.19 *h est satisfaisable si et seulement si f est satisfaisable.*

Afin de garantir une forme normale de taille polynomiale, il nous faut renommer toutes les sous-formules susceptibles d'être dupliquées par la transformation en forme clausale. On remarque que g est une sous-formule de $f' \wedge (p \Leftrightarrow g)$; il en est donc de même de toutes les sous-formules de g.

Définition 2.20 Une formule de la forme $f \ \omega \ f'$ où $\omega \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$ est dite *composée*.

Pour toute formule $f \in \mathrm{P}(\mathcal{S})$, soit $\{g_1, \ldots, g_n\}$ l'ensemble de ses sous-formules composées, ordonné de sorte que si g_i est une sous-formule de g_j alors $i \geq j$. Soit $\mathcal{S}' = \mathcal{S} + p_1 + \cdots + p_n$, on définit la suite $(f_i)_{i=1}^{n+1}$ par :

— $f_1 = f$,

— pour $1 \leq i \leq n$, f_{i+1} est la renommée de f_i en g_i par p_i,

alors f_{n+1} est une *forme renommée* de f. △

On remarque que g_i est bien une sous-formule de f_i puisque grâce à l'ordre utilisé, aucune des sous-formules de g_i n'a été renommée avant f_i ; la suite $(f_i)_{i=1}^{n+1}$ est donc bien définie. Cette succession de renommage préserve l'extensibilité des modèles grâce à la propriété suivante.

Lemme 2.21 *Soient $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{S}''$, $f \in \mathrm{P}(\mathcal{S})$, $f' \in \mathrm{P}(\mathcal{S}')$ et $f'' \in \mathrm{P}(\mathcal{S}'')$, si $f \models_0^{\mathcal{S}} f'$ et $f' \models_0^{\mathcal{S}'} f''$ alors $f \models_0^{\mathcal{S}} f''$.*

PREUVE. Pour tout $I \in \mathrm{I_p}(\mathcal{S})$ tel que $I \models_0 f$, comme $f \models_0^{\mathcal{S}} f'$ il existe $I' \in \mathrm{I_p}(\mathcal{S}')$ tel que $I' \models_0 f'$ et $I'|_{\mathcal{S}} = I$, et comme $f' \models_0^{\mathcal{S}'} f''$ il existe $I'' \in \mathrm{I_p}(\mathcal{S}'')$ tel que $I'' \models_0 f'$ et $I''|_{\mathcal{S}'} = I'$. On a $I''|_{\mathcal{S}} = (I''|_{\mathcal{S}'})|_{\mathcal{S}} = I'|_{\mathcal{S}} = I$, donc $f \models_0^{\mathcal{S}} f''$. •

On en déduit :

Théorème 2.22 *Soit $f \in \mathrm{P}(\mathcal{S})$, si f' est une forme renommée de f, alors $f \models_0^{\mathcal{S}} f'$ et $f' \models_0 f$.*

PREUVE. On reprend les notations de la Définition 2.20, avec $f' = f_{n+1}$. D'après le Lemme 2.18, on a $f_{n+1} \models_0 f_n \models_0 \cdots \models_0 f_1$, donc $f' \models_0 f$. On a également

$$f_i \models_0^{\mathcal{S}+p_1+\cdots+p_{i-1}} f_{i+1} \text{ pour tout } 1 \leq i \leq n.$$

Montrons par induction sur $1 \leq i \leq n+1$ que $f \models_0^{\mathcal{S}} f_i$. La propriété est triviale pour $i = 1$. Si elle est vraie pour $i \leq n$, comme $\mathcal{S} \subseteq \mathcal{S} + p_1 + \cdots + p_{i-1} \subseteq \mathcal{S}'$ par le Lemme 2.21 on obtient $f \models_0^{\mathcal{S}} f_{i+1}$. La propriété est donc vraie pour $i = n+1$, ce qui donne $f \models_0^{\mathcal{S}} f'$. •

S'il n'y a pas de sous-formule composée dans f alors $n = 0$ et $f' = f$. Dans ce cas f ne peut contenir que des $\neg$ et une occurrence de $\blacksquare$, $\square$ ou d'un symbole propositionnel. Une telle formule se réduit par élimination de la négation à $\blacksquare$, $\square$ ou à un littéral.

Si $n \geq 1$ alors la forme renommée est une conjonction d'une formule qui se réduit à p_1 ou à $\neg p_1$, et de n formules de la forme $p_i \Leftrightarrow (g\ \omega\ g')$ où $\omega \in \{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$ et g, g' n'ont pas de sous-formule composée.

EXERCICE 23. *Montrer que toute fnc de $p_i \Leftrightarrow (g\ \omega\ g')$ contient au plus 6 clauses, chacune contenant au plus 3 littéraux.*

La forme clausale de f' contient donc au plus $6n+1$ clauses. La transformation en fnc des formes renommées est donc linéaire. D'après le Théorème 2.22, si une forme renommée f' de f est valide alors f est valide. Malheureusement, la réciproque est fausse.

EXERCICE 24. *Donner une formule valide dont une forme renommée n'est pas valide.*

On ne peut donc pas utiliser le renommage pour réduire polynomialement les formules valides à $\blacksquare$. Par contre, le renommage et la transformation sous forme clausale préservent la satisfaisabilité ; nous pouvons donc l'utiliser pour montrer qu'une formule est insatisfaisable. Plus précisément, nous avons réduit polynomialement le problème de la satisfaisabilité d'une formule quelconque au problème de la satisfaisabilité d'un ensemble fini de clauses.

2.5 Résolution

Dans la suite nous allons définir deux systèmes permettant de prouver l'insatisfaisabilité d'une forme clausale : la Résolution et la procédure DPLL. Ces deux systèmes d'inférence sont donc basés sur le principe du raisonnement par l'absurde, ou *reductio ad absurdum*, qui nous est évidemment autorisé puisque la logique propositionnelle admet une négation.

Définition 2.23 Une déduction dans un système d'inférence de la formule $\square$ est appelée une *réfutation*. △

La méthode de résolution a été introduite par Robinson en 1965. En logique propositionnelle, elle permet de prouver l'insatisfaisabilité d'une forme clausale au

moyen d'une *unique* règle d'inférence sur les clauses. Cette règle très simple est une généralisation du *modus ponens*. Nous introduisons d'abord une notation permettant d'enlever les occurrences d'un littéral dans une clause.

Définition 2.24 On note $C = C_1 \dot\vee C_2$ ssi $\mathrm{Lit}(C) = \mathrm{Lit}(C_1) \cup \mathrm{Lit}(C_2)$ et $\mathrm{Lit}(C_1) \cap \mathrm{Lit}(C_2) = \emptyset$. △

EXEMPLE 2.25 Soit $C = p \vee \neg q \vee r \vee p \vee r \vee q$, alors on peut écrire $C = (p \vee \neg q \vee r) \dot\vee q$.◊

Les clauses C_1 ou C_2 peuvent être vides, et il est évident que pour toute clause C on peut écrire $C = C \dot\vee \Box$.

Définition 2.26 (Résolution propositionnelle) On définit le système d'inférence suivant : $\mathrm{S_r} = \langle \mathrm{V_p}, \mathcal{C}_\mathcal{S}, \{\mathrm{Res}_0\}\rangle$, où Res_0 est la *règle de résolution* (propositionnelle) : $\langle C_1, C_2, R\rangle \in \mathrm{Res}_0$ si et seulement s'il existe un littéral l et deux clauses D_1, D_2 telles que $C_1 = l \dot\vee D_1$, $C_2 = \bar l \dot\vee D_2$ et $R = D_1 \vee D_2$. On peut donc noter :

$$\frac{l \dot\vee D_1 \qquad \bar l \dot\vee D_2}{D_1 \vee D_2}\mathrm{Res}_0$$

La clause R est la *résolvante* de C_1 et C_2. On notera la relation de déduction $\vdash_\mathrm{r}$ au lieu de $\vdash_{\mathrm{S_r}}$. △

On remarque que ni l ni $\bar l$ ne peuvent apparaître dans la conclusion.

EXEMPLE 2.27 On a par résolution

$$\frac{p \vee q \vee r \qquad \neg p \vee q \vee \neg p \vee \neg r}{q \vee r \vee q \vee \neg r}$$

avec $D_1 = q \vee r$, $D_2 = q \vee \neg r$ et $l = p$. ◊

La première question à se poser est celle de la correction de ce système d'inférence.

Théorème 2.28 $\mathrm{S_r}$ *est correct pour* $\mathrm{L_p}$.

PREUVE. Soient C_1, C_2 des clauses telles que $C_1 = l \dot\vee D_1$ et $C_2 = \bar l \dot\vee D_2$, et soit $I \in \mathrm{I_p}$ telle que $I \models_0 C_1$ et $I \models_0 C_2$. Il s'agit de prouver que $I \models_0 D_1 \vee D_2$.

Supposons que $I \models_0 l$, le cas où $I \models_0 \bar l$ est symétrique. Comme $I \models_0 l$, nécessairement $I \not\models_0 \bar l$, ce qui signifie qu'il existe un littéral m dans D_2 tel que $I \models_0 m$, car $I \models_0 C_2$. Comme m apparaît également dans $D_1 \vee D_2$, on a $I \models_0 D_1 \vee D_2$. •

EXERCICE 25. *L'inférence suivante est-elle correcte ?*

$$\frac{p \vee q \vee r \qquad \neg p \vee \neg q}{r}$$

Comme il n'y a aucun axiome dans $\mathrm{S_r}$, il ne contient aucun théorème et il est donc utilisé pour vérifier qu'une forme clausale F est insatisfaisable. Il suffit pour cela d'en déduire une clause vide ; on a ainsi une réfutation $F \vdash_{\mathrm{r}} \square$, ce qui entraîne $F \models_0 \square$ par la correction de $\mathrm{S_r}$ et donc prouve l'insatisfaisabilité de F.

EXEMPLE 2.29 On considère la forme clausale

$$F = \{p \vee \neg q,\ q \vee r,\ p \vee \neg r,\ \neg p \vee r,\ \neg p \vee \neg r\}\,,$$

on peut en construire la réfutation suivante :

$$\dfrac{\dfrac{p \vee \neg q \qquad \dfrac{q \vee r \qquad p \vee \neg r}{p \vee q}}{p \vee p} \qquad \dfrac{\neg p \vee r \qquad \neg p \vee \neg r}{\neg p \vee \neg p}}{\square}$$

et on peut donc en déduire par le Théorème 2.28 que F est insatisfaisable.

Cette réfutation peut également être donnée sous la forme d'une séquence de clauses, conformément à la définition d'une déduction. Par souci de lisibilité on indique les prémisses menant à une résolvante.

1. $p \vee \neg q$
2. $q \vee r$
3. $p \vee \neg r$
4. $\neg p \vee r$
5. $\neg p \vee \neg r$
6. $p \vee q$ 2, 3
7. $p \vee p$ 1, 6
8. $\neg p \vee \neg p$ 4, 5
9. $\square$ 7, 8

Cette présentation est préférable lorsqu'une clause déduite est utilisée plusieurs fois ou que le nombre de clauses est trop grand. ◊

Il reste à garantir que $\mathrm{S_r}$ permet bien de déduire la clause vide de tout ensemble insatisfaisable de clauses. Un système d'inférence vérifiant cette propriété satisfait une forme de complétude particulière appelée *complétude réfutationnelle.*

Théorème 2.30 *Si une forme clausale F est insatisfaisable, alors $F \vdash_{\mathrm{r}} \square$.*

PREUVE. S'il existe dans F une clause C telle que $|C| = 0$, alors $C \equiv_0 \square$ et trivialement, $F \vdash_{\mathrm{r}} \square$. Nous supposons donc que F contient uniquement des clauses C telles que $|C| \geq 1$. Pour tout $F = \{C_1, \ldots, C_n\}$, nous définissons la mesure $\mathrm{d}(F) = \sum_{i=1}^{n}(|C_i| - 1)$. Nous démontrons le résultat par induction sur $\mathrm{d}(F)$.

Si $\mathrm{d}(F) = 0$, alors comme F ne contient que des clauses de longueur au moins 1, nécessairement, pour tout $C \in F$, $|C| = 1$. Comme F est insatisfaisable, il doit exister deux clauses $C_i \equiv_0 p$ et $C_j \equiv_0 \neg p$, et donc $C_i, C_j \vdash_{\mathrm{r}} \square$. On en déduit qu'on a bien $F \vdash_{\mathrm{r}} \square$.

Soit $n' > 0$, supposons la propriété vraie pour tout ensemble de clauses F' tel que $\mathrm{d}(F') < n'$, et supposons que $\mathrm{d}(F) = n'$. Alors il existe dans F une clause C telle que $|C| > 1$, et on a $C = l \dot{\vee} D$, où l est un littéral et D est une clause non vide telle que $|D| < |C|$.

Considérons les ensembles suivants et évaluons leur image par la mesure d :

$$\begin{array}{rclcrclcrcl}
F' & = & F \setminus \{C\} & & F_1 & = & F' \cup \{D\} & & F_2 & = & F' \cup \{l\} \\
\mathrm{d}(F') & \leq & \mathrm{d}(F) - |C| + 1 & & \mathrm{d}(F_1) & \leq & \mathrm{d}(F') + |D| - 1 & & \mathrm{d}(F_2) & \leq & \mathrm{d}(F') \\
 & < & \mathrm{d}(F) & & & < & \mathrm{d}(F') + |C| - 1 & & & < & \mathrm{d}(F). \\
 & & & & & < & \mathrm{d}(F) & & & &
\end{array}$$

Comme F_1 et F_2 contiennent F', on a $F_1 \models_0 F'$ et $F_2 \models_0 F'$. De plus, d'après le Théorème 2.13, on a $D \models_0 C$ et $l \models_0 C$, donc $F_1 \models_0 F$ et $F_2 \models_0 F$. Mais F est insatisfaisable, donc F_1 et F_2 doivent également être insatisfaisables. Par hypothèse d'induction on a donc $F_1 \vdash_{\mathrm{r}} \square$ et $F_2 \vdash_{\mathrm{r}} \square$.

Nous montrons comment utiliser la réfutation $F_1 \vdash_{\mathrm{r}} \square$ pour construire une déduction à partir de F qui est de la forme $F \vdash_{\mathrm{r}} l$ ou $F \vdash_{\mathrm{r}} \square$. Soit $(C_1, \ldots, C_k)$ une déduction de $C_k = \square$ à partir de $F' \cup \{D\}$, on construit inductivement une déduction $(C'_1, \ldots, C'_k)$ à partir de F telle que pour tout $1 \leq m \leq k$,

$$F \vdash_{\mathrm{r}} C'_m \quad \text{et} \quad \mathrm{Lit}(C_m) \subseteq \mathrm{Lit}(C'_m) \subseteq \mathrm{Lit}(C_m \vee l). \tag{2.1}$$

Supposons la propriété (2.1) vraie avant m, on distingue 3 cas.

- Si $C_m \in F'$, alors on pose $C'_m = C_m$. Il est clair que $C'_m \in F$, donc cette clause vérifie (2.1).
- Si $C_m = D$, alors on pose $C'_m = C$. Comme $C \in F$ et $C = l \dot{\vee} D$ la propriété (2.1) est bien vérifiée.
- Sinon, il existe $i, j < m$ tels que $(C_i, C_j, C_m) \in \mathrm{Res}_0$, donc par définition, il existe l', D_i, D_j tels que $C_i = l' \dot{\vee} D_i, C_j = \overline{l'} \dot{\vee} D_j$ et $C_m = D_i \vee D_j$. Par hypothèse d'induction il existe C'_i et C'_j telles que

$$\begin{array}{c}
F \vdash_{\mathrm{r}} C'_i \ \text{ et } \ \mathrm{Lit}(l' \dot{\vee} D_i) \subseteq \mathrm{Lit}(C'_i) \subseteq \mathrm{Lit}((l' \dot{\vee} D_i) \vee l), \\
F \vdash_{\mathrm{r}} C'_j \ \text{ et } \ \mathrm{Lit}(\overline{l'} \dot{\vee} D_j) \subseteq \mathrm{Lit}(C'_j) \subseteq \mathrm{Lit}((\overline{l'} \dot{\vee} D_j) \vee l).
\end{array}$$

Le littéral l' doit donc apparaître dans C'_i, et le littéral $\overline{l'}$ dans C'_j. Considérons les clause D'_i et D'_j telles que $C'_i = l' \dot{\vee} D'_i$ et $C'_j = \overline{l'} \dot{\vee} D'_j$. Alors, même dans les cas $l = l'$ ou $l = \overline{l'}$ on a :

$$\begin{array}{rcccl} \mathrm{Lit}(D_i) & \subseteq & \mathrm{Lit}(D'_i) & \subseteq & \mathrm{Lit}(D_i \vee l), \\ \mathrm{Lit}(D_j) & \subseteq & \mathrm{Lit}(D'_j) & \subseteq & \mathrm{Lit}(D_j \vee l). \end{array}$$

On pose alors $C'_m = D'_i \vee D'_j$, de sorte que $(C'_i, C'_j, C'_m) \in \mathrm{Res}_0$. Comme $\mathrm{Lit}(C_m) = \mathrm{Lit}(D_i) \cup \mathrm{Lit}(D_j)$ et $\mathrm{Lit}(C'_m) = \mathrm{Lit}(D'_i) \cup \mathrm{Lit}(D'_j)$, on a

$$F \vdash_{\mathrm{r}} C'_m \ \text{ et } \ \mathrm{Lit}(C_m) \subseteq \mathrm{Lit}(C'_m) \subseteq \mathrm{Lit}(C_m \vee l),$$

donc (2.1) est bien vérifiée.

L'induction étant vérifiée, on en déduit que $F \vdash_{\mathrm{r}} C'_k$, où $\mathrm{Lit}(C'_k) \subseteq \{l\}$. Si $\mathrm{Lit}(C'_k) = \emptyset$ alors $F \vdash_{\mathrm{r}} \square$. Sinon $F \vdash_{\mathrm{r}} l$, et comme $F \cup \{l\} \vdash_{\mathrm{r}} \square$ (puisque $F_2 \subset F \cup \{l\}$ et $F_2 \vdash_{\mathrm{r}} \square$), on a également $F \vdash_{\mathrm{r}} \square$. •

La règle de résolution est non déterministe : il existe en général plusieurs clauses candidates sur lesquelles cette règle peut être appliquée. Ceci peut entraîner une perte d'efficacité, s'il n'est pas garanti que le choix des clauses auxquelles appliquer la règle est optimal. Mais il est évidemment très difficile –voire impossible– de garantir ce choix optimal. Il existe néanmoins des *stratégies* permettant de réduire les clauses candidates à considérer pour appliquer la règle de résolution, tout en préservant la complétude réfutationnelle du calcul.

Exercice 26. *Trouver une autre réfutation de l'Exemple 2.29.*

Remarque. Il existe dans la littérature d'autres formalismes pour dénoter des clauses. Les définitions suivantes sont fréquemment utilisées :

1. Une clause est un ensemble de littéraux. Cette définition revient à identifier une clause C avec l'ensemble $\mathrm{Lit}(C)$ ci-dessus, et donc de considérer les clauses modulo l'idempotence, l'associativité et la commutativité de la disjonction $\vee$. Par exemple, la clause $p \vee q \vee r \vee p$ serait représentée par l'ensemble $\{p, q, r\}$.
2. Une clause est un *multiensemble*[4] de littéraux. Par exemple, la clause $p \vee q \vee r \vee p$ serait représentée par le multiensemble $\{p, p, q, r\}$. Cette représentation tient compte de l'associativité et de la commutativité de $\vee$, mais pas de l'idempotence.

Ces formalismes permettent de s'approcher de la structure de données qui sera utilisée dans un programme pour représenter les clauses. Ces choix ne sont pas fondamentalement différents, et il n'est pas nécessaire d'en connaître les particularités algorithmiques pour comprendre la méthode de résolution. C'est pourquoi nous avons préféré une présentation plus pure où les clauses sont des formules, c'est-à-dire des mots dans un langage conformément au Chapitre 1.

4. Un multiensemble est un ensemble permettant de multiples occurrences d'un même élément.

2.6 DPLL

En 1960, Martin Davis et Hilary Putnam ont introduit une procédure permettant de tester la validité d'une formule. Cette procédure, basée sur les mêmes idées que la résolution, pouvait nécessiter exponentiellement plus d'espace que la formule d'origine avant d'aboutir à une solution. En 1962, les deux auteurs, associés à George Logemann et Donald Loveland ont proposé un nouvel algorithme, communément appelé algorithme DPLL, qui permet de tester la validité d'une formule. Contrairement à la Résolution, cet algorithme est difficilement adaptable à la logique du premier ordre en général. Cependant, il est très efficace en logique propositionnelle, et est encore à la base de presque tous les outils permettant de tester la satisfaisabilité d'une formule propositionnelle.

Les raffinements de cet algorithme sont l'objet de nombreux travaux de recherche encore aujourd'hui, car tester la satisfaisabilité d'une formule propositionnelle est une façon fréquemment utilisée dans l'industrie pour s'assurer par exemple qu'un circuit électronique répond bien à une spécification donnée et ne contient pas d'erreurs. Tous les ans, une compétition est organisée pour déterminer le meilleur *solveur SAT*, c'est-à-dire l'outil capable de tester la satisfaisabilité d'une formule propositionnelle le plus rapidement possible. Les solveurs les plus efficaces sont aujourd'hui capables de résoudre des problèmes constitués de centaines de milliers de symboles propositionnels et de dizaines de millions de clauses en moins d'une minute.

L'algorithme est basé sur la construction incrémentale d'un modèle à l'ensemble de clauses considéré, en cherchant successivement quels littéraux de l'ensemble doivent être interprétés à vrai et lesquels doivent être interprétés à faux. L'idée sous-jacente à l'algorithme est très simple : soit E un ensemble de clauses et l est un littéral apparaissant dans E. Si, lors de la recherche d'un modèle de E, l est évalué à vrai, alors toutes les clauses contenant l sont évaluées à vrai, et toutes les clauses contenant son complémentaire doivent être évaluées à vrai grâce à d'autres littéraux. Il n'est donc pas nécessaire de considérer les clauses contenant l, il n'est pas nécessaire de considérer $\bar{l}$ dans les clauses contenant ce complémentaire. Ceci mène à la définition suivante :

Définition 2.31 Si E est un ensemble de clauses et l est un littéral apparaissant dans E, alors on note $E[l]$ l'ensemble de clauses obtenu à partir de E en :
— retirant de E toutes les clauses qui contiennent l,
— retirant $\bar{l}$ de toutes les clauses de E dans lesquelles il apparaît. △

Exemple 2.32 Considérons les clauses suivantes :

$$\begin{aligned} C_1 &= p_1 \vee \neg p_2 \vee p_3, \\ C_2 &= p_1 \vee \neg p_4, \\ C_3 &= \neg p_1 \vee \neg p_3, \end{aligned}$$

et soit $E = \{C_1, C_2, C_3\}$. Alors :

$$\begin{aligned} E[p_1] &= \{\neg p_3\}, \\ E[\neg p_3] &= \{p_1 \vee \neg p_2, p_1 \vee \neg p_4\}. \end{aligned}$$ ◊

Nous présentons tout d'abord une version simplifiée de l'algorithme DPLL, afin d'en montrer le fonctionnement global. Nous définissons un système d'inférence contenant deux règles d'inférence ; la traduction de ce système en algorithme peut être considéré comme une version primaire de l'algorithme DPLL. Nous ajouterons ensuite de nouvelles règles au système d'inférence, et la traduction du système résultant correspondra à l'algorithme DPLL original.

Définition 2.33 Soit $\mathcal{S}'_{\text{DPLL}} = \langle \mathrm{V_p}, \mathcal{C}_\mathcal{S}, \mathcal{R}'_{\text{DPLL}} \rangle$, où $\mathcal{R}'_{\text{DPLL}}$ contient les règles suivantes :

$$\frac{E \cup \{l \dot\vee C\}}{E[l]} Lit \qquad \frac{E \cup \{l \dot\vee C\}}{E[\bar{l}] \cup \{C\}} Comp$$ △

Proposition 2.34 *$\mathcal{S}'_{\text{DPLL}}$ termine.*

Preuve. Ce résultat est évident : si E est un ensemble de clauses construit sur n symboles propositionnels et si (E, E') est une dérivation valide dans $\mathcal{S}'_{\text{DPLL}}$, alors E' est un ensemble de clauses construit sur $n-1$ symboles propositionnels. Donc, toute dérivation à partir de E est de longueur au plus n, et il est aisé de vérifier que la conclusion de toute dérivation de longueur maximale est soit $\emptyset$, soit $\{\Box\}$. •

Le fait que $\mathcal{S}'_{\text{DPLL}}$ fournit un algorithme permettant de tester si un ensemble de clauses est satisfaisable ou non est une conséquence de la propriété suivante :

Proposition 2.35 *Soient l un littéral, C une clause et E un ensemble de clauses. Alors $E \cup \{l \dot\vee C\}$ est satisfaisable si et seulement si l'un des ensembles $E[l]$ ou $E[\bar{l}] \cup \{C\}$ est satisfaisable.*

Preuve. Supposons que $E \cup \{l \dot\vee C\}$ est satisfaisable, et soit I une interprétation telle que $I \models_0 E \cup \{l \dot\vee C\}$. Supposons sans perte de généralité que $I \models_0 l$, alors I interprète toutes les clauses contenant l à V et interprète un littéral autre que $\bar{l}$ à V dans les clauses contenant ce complémentaire. Ainsi, $I \models_0 E[l]$.

Réciproquement, supposons que l'un des ensembles $E[l]$ ou $E[\bar{l}] \cup \{C\}$ est satisfaisable. Sans perte de généralité, supposons qu'il existe une interprétation I telle que $I \models_0 E[l]$. On considère alors l'interprétation J qui interprète l à V, et coïncide avec I sur $\mathrm{Lit}(E[l])$. Il est clair que $J \models_0 E[l]$, et que J interprète toute clause contenant l à V. Donc, $J \models_0 E \cup \{l \dot\vee C\}$, et cet ensemble est satisfaisable. •

Cette proposition permet de déduire le résultat suivant :

Corollaire 2.36 *$\mathcal{S}'_{\text{DPLL}}$ est correct et complet pour la réfutation.*

EXERCICE 27. *Donner un algorithme en pseudo-code qui simule les règles d'inférence de $\mathcal{S}'_{\text{DPLL}}$. On supposera que la fonction écrite retourne le booléen 'vrai' si l'ensemble de clauses fourni en entrée est satisfaisable, et le booléen 'faux' s'il ne l'est pas.*

EXERCICE 28. *Appliquer l'algorithme DPLL à l'ensemble de clauses contenant les clauses suivantes :*

$$\begin{aligned} C_1 &= p \vee q \vee r \\ C_2 &= p \vee \neg q \vee r \\ C_3 &= \neg p \vee r \\ C_4 &= \neg r \end{aligned}$$

Tracer l'arbre d'appels de la fonction. Que constate-t-on ?

L'algorithme DPLL tel qu'il est présenté énumère toutes les interprétations susceptibles de satisfaire l'ensemble de clauses considéré. Si l'ensemble de clauses en question est construit sur un ensemble de symboles propositionnels de cardinalité n, alors l'algorithme teste 2^n interprétations. Ceci est évidemment très inefficace, c'est pourquoi de nombreux raffinements ont été proposés pour optimiser cet algorithme. Nous modifierons le système d'inférence $\mathcal{S}'_{\text{DPLL}}$ après avoir donné une définition intermédiaire :

Définition 2.37 Soit E un ensemble de clauses et l un littéral apparaissant dans E. Le littéral l est *pur dans* E si $\bar{l}$ n'apparaît pas dans E. △

Définition 2.38 On définit le système d'inférence $\mathcal{S}_{\text{DPLL}} = \langle \text{V}_\text{p}, \mathcal{C}_\mathcal{S}, \mathcal{R}_{\text{DPLL}} \rangle$, où $\mathcal{R}_{\text{DPLL}}$ contient les règles de $\mathcal{R}'_{\text{DPLL}}$ ainsi que les règles suivantes :

— La règle de *propagation unitaire*, qui force l'interprétation de l'unique littéral apparaissant dans une clause unitaire à V :

$$\frac{E \cup \{l \dot{\vee} \square\}}{E[l]} UnitProp$$

— La règle du *littéral pur*, qui permet de supprimer certaines clauses de E :

$$\frac{E \cup \{l \dot{\vee} C\}}{E'} PureLit,$$

où l est pur dans E et E' est obtenu en supprimant de E toute clause qui contient l.

— La règle de *subsomption*, qui permet également de supprimer certaines clauses de E :

$$\frac{E \cup \{C, C'\}}{E \cup \{C\}} Subs,$$

où C subsume C'. △

Les règles du littéral pur et de subsomption sont appelées des *règles de simplification* car l'unique transformation qu'elles effectuent sur un ensemble de clauses et d'en supprimer certains éléments.

Théorème 2.39 *$\mathcal{S}_{\text{DPLL}}$ est correct et complet pour la réfutation.*

PREUVE. Il s'agit de montrer que chacune des règles préserve la satisfaisabilité.

Règle du littéral pur. Soit l un littéral qui est pur dans E, et soit E' l'ensemble obtenu en supprimant de E toutes les clauses contenant l. Montrons que E est satisfaisable si et seulement si E' est satisfaisable. Le fait que E' est satisfaisable si E l'est est évident, puisque $E' \subseteq E$. Supposons maintenant que E' est satisfaisable, soit I un modèle de E'. On définit J comme l'interprétation qui coïncide avec I sur $\text{Lit}(E')$, et telle que J interprète l à V. Alors $J \models_0 E'$, et pour toute clause C dans E qui contient l, on a aussi $J \models_0 C$. Comme toutes les clauses dans E qui ne contiennent pas l sont dans E', on en déduit que $J \models_0 E$.

Règle de la subsomption. Soient C et C' des clauses telles que C subsume C', et soit E un ensemble de clauses. Il est clair que si $E \cup \{C, C'\}$ est satisfaisable alors $E \cup \{C\}$ l'est également. Montrons maintenant la contraposée. Supposons que $E \cup \{C\}$ est satisfaisable, et soit I un modèle de cet ensemble. Nous montrons que $I \models_0 E \cup \{C, C'\}$. Ceci est évident : comme $I \models_0 C$, il existe un littéral l dans C tel que I interprète l à V. Mais comme C subsume C', par définition, $\text{Lit}(C) \subseteq \text{Lit}(C')$, ce qui signifie que l est également un littéral dans C'. Donc, nécessairement, $I \models_0 C'$, d'où le résultat. •

EXERCICE 29. *Prouver que la règle de propagation unitaire préserve la satisfaisabilité.*

En général, il est possible que plusieurs règles de $\mathcal{S}_{\text{DPLL}}$ soient applicables en même temps à un ensemble de clause, d'où la nécessité de mettre en place une stratégie pour fixer une priorité dans l'application des règles. La priorité fixée est standard : elle consiste à accorder la priorité maximale aux règles de simplification, puis à la règle de propagation unitaire, et enfin aux règles *Lit* et *Comp*.

EXERCICE 30. *Modifier l'algorithme DPLL de l'Exercice 27 pour prendre en compte les nouvelles règles et la stratégie associée.*

2.7 Compacité

Les propriétés de complétude réfutationnelle que nous avons établies s'appliquent à des formes clausales, donc à des ensembles finis de clauses. Mais un ensemble d'hypothèses peut, tout comme un ensemble d'axiomes, être infini, même si une réfutation ne peut en utiliser qu'un nombre fini. On peut donc se demander si nous serions capables de trouver une réfutation d'un ensemble de clauses infini et insatisfaisable. Un tel résultat généraliserait la complétude réfutationnelle aux ensembles infinis.

Nous verrons dans la suite qu'il ne s'agit pas d'une spéculation abstraite : le raisonnement logique doit se confronter à l'infini, car c'est là qu'il devient indispensable. Considérer une infinité de clauses ou plus généralement de formules propositionnelles en constitue une première étape. Heureusement, la logique propositionnelle rend la chose aisée grâce à une propriété de *compacité* qui permet de toujours se ramener à un ensemble fini, même si nous ne le connaissons pas *a priori*.

Théorème 2.40 (de compacité) *Pour tout ensemble $E \subseteq \mathrm{P}$ de formules propositionnelles, E est insatisfaisable si et seulement si il existe un sous-ensemble fini de E qui est insatisfaisable.*

Preuve. La condition suffisante est évidente : si un sous-ensemble fini de E est insatisfaisable, alors E l'est aussi.

Nous montrons la contraposée de la condition nécessaire : si tout sous-ensemble fini de E est satisfaisable, alors E est satisfaisable. Ce résultat est évident lorsque E est fini puisque E est un sous-ensemble fini de lui-même ; nous supposons donc E infini. On ordonne l'ensemble des symboles propositionnels : $\mathcal{S} = \{p_1, \ldots, p_{|\mathcal{S}|}\}$ si $\mathcal{S}$ est fini et $\mathcal{S} = \{p_n \mid n \in \mathbb{N}^*\}$ sinon. On construit par induction sur n les images $I(p_1), \ldots, I(p_n)$ d'une interprétation I en maintenant la propriété suivante :

$$\forall F \text{ fini} \subset E,\ \exists J \models_0 F \text{ tel que } J =_{|p_1 \cdots p_n} I, \tag{2.2}$$

où la notation $J =_{|p_1 \cdots p_n} I$ signifie que I et J interprètent les symboles propositionnels $p_1, \ldots, p_n$ de la même manière. Si $J =_{|p_1 \cdots p_n} I$, on dira que J *correspond à I jusqu'à p_n*. Nous allons donc prouver qu'il existe un modèle I *commun* à tous les sous-ensembles finis de E (il est facile de vérifier que s'il existe des sous-ensembles finis $F, F' \subset E$ qui n'ont aucun modèle commun, alors il n'existe pas de modèle pour E). Nous montrerons ensuite que cette interprétation I est un modèle de E.

Pour $n = 0$ la propriété (2.2) est vraie puisque la condition $J =_{|p_1 \cdots p_n} I$ est trivialement vraie pour tout J si $n = 0$. Supposons maintenant qu'on a défini $I(p_1), \ldots, I(p_n)$ vérifiant (2.2) pour une valeur $n \geq 1$ (et où $n < |\mathcal{S}|$ si $\mathcal{S}$ est fini), on distingue deux cas complémentaires :

(ι) tout sous-ensemble fini de E admet un modèle qui correspond à I jusqu'au symbole p_n, et qui interprète p_{n+1} à vrai, i.e., $\forall F$ fini $\subset E$, $\exists J \models_0 F$ tel que $J =_{|p_1 \cdots p_n} I$ et $J(p_{n+1}) = \mathrm{V}$. On pose alors $I(p_{n+1}) = \mathrm{V}$,

($\iota\iota$) il existe un sous-ensemble fini de E dont tous les modèles correspondant à I jusqu'à p_n interprètent p_{n+1} à faux. On pose alors $I(p_{n+1}) = \text{F}$. Soit K un tel ensemble, c'est-à-dire tel que $\forall J \models_0 K$, si $J =_{|p_1 \cdots p_n} I$ alors $J(p_{n+1}) = \text{F}$.

Pour montrer que (2.2) est vraie en $n+1$, on considère un sous-ensemble $F \subset E$ fini quelconque. Dans le cas (ι), il existe une interprétation $J \models_0 F$ telle que $J =_{|p_1 \cdots p_n} I$ et $J(p_{n+1}) = \text{V}$. Comme on a également $I(p_{n+1}) = \text{V}$, on en déduit que $J =_{|p_1 \cdots p_{n+1}} I$, et la propriété (2.2) est vérifiée en $n+1$. Dans le cas ($\iota\iota$), par construction, $I(p_{n+1}) = \text{F}$. Comme $F \cup K \subset E$ est fini, par hypothèse d'induction, il existe une interprétation J telle que $J \models_0 F \cup K$ et $J =_{|p_1 \cdots p_n} I$. Donc, en particulier [5], $J \models_0 F$ et $J \models_0 K$, et comme $J \models_0 K$, nécessairement $J(p_{n+1}) = \text{F}$. On a donc $J =_{|p_1 \cdots p_{n+1}} I$ et $J \models_0 F$, la propriété (2.2) est donc vérifiée en $n+1$.

La propriété (2.2) est donc vraie pour tout $n \in \mathbb{N}$ (tel que $n \leq |S|$ si S est fini). Il reste à vérifier que $I \models_0 E$. Soit $f \in E$, il existe $n \in \mathbb{N}$ tel que tous les symboles propositionnels apparaissant dans f sont dans $\{p_1, \ldots, p_n\}$. Comme $\{f\} \subset E$ est fini, par (2.2) il existe $J \models_0 f$ tel que $J =_{|p_1 \cdots p_n} I$, on a donc également $I \models_0 f$. •

EXERCICE 31. *Prouver que pour tout ensemble insatisfaisable de clauses E on a $E \vdash_{\text{r}} \square$.*

5. Voir l'Exercice 1 (1).

Chapitre 3

La logique du premier ordre

La logique propositionnelle permet bien de rendre compte des syllogismes aristotéliciens qui furent étudiés et enseignés jusqu'au XIXème siècle. Pourtant, la science qui se développait depuis la Renaissance n'en faisait guère usage alors qu'à l'instar de la philosophie de l'antiquité elle recherchait la vérité et devait donc se préoccuper de la correction des raisonnements qu'elle inventait. Mais les syllogismes étaient en fait incapables de rendre compte de la plupart de ces raisonnements, ni même de beaucoup de raisonnements courants. Par exemple, le raisonnement ci-dessous :

`Jean est vigneron`
`Le frère de Jean est Paul`
donc `Le frère de Paul est vigneron`

est correct mais ne correspond à aucun syllogisme. Ce raisonnement simple n'est pas propositionnel, il porte sur des personnes et les relations qui existent entre elles. Il nécessite une analyse de la structure des propositions qui va au delà de la simple reconnaissance des connecteurs de la logique propositionnelle. On comprend facilement que le raisonnement scientifique, qui portait essentiellement sur des nombres, nécessitait déjà des moyens plus élaborés.

La science se développait donc en latin, mais comme toute langue naturelle il était peu commode d'y développer des raisonnements précis, en particulier numériques. La Renaissance bénéficiait heureusement de la numération décimale, introduite en Occident vers la fin du premier millénaire depuis l'Orient *via* le califat omeyyade de Cordoue, ainsi que d'une algèbre [1] calculatoire de même origine dans laquelle on étudiait les équations de premier et second degrés.

Cet embryon de langage mathématique connut une avancée décisive à la fin du XVIème siècle grâce à l'invention [2] de la notion de *variable* par François Viète. En notant les paramètres connus par des consonnes et les paramètres inconnus (les

1. On reconnaît facilement l'origine arabe des mots algèbre et algorithme.

2. Il est bien sûr possible de reconnaître des variables dans les travaux des mathématiciens de l'antiquité, mais il s'agit là d'une lecture moderne. La *notion* de variable n'était pas connue et leur emploi ne pouvait donc être théorisé.

variables) par des voyelles, il développe une « Algèbre Nouvelle » débarrassée des aspects calculatoires de l'ancienne algèbre. Pour des raisons de lisibilité, René Descartes utilisera ensuite les premières lettres de l'alphabet pour les paramètres et les dernières pour les inconnues ; cette convention est toujours en usage.

Le calcul devenu de plus en plus symbolique [3] (par opposition à numérique) appelait toujours plus d'invention et d'audace. Ainsi le développement du calcul infinitésimal par Leibniz et Newton au XVIIème siècle étendait le calcul à des entités qui restèrent longtemps mystérieuses. Si l'on pouvait ajouter une « quantité infinitésimale » de x, notée dx par Leibniz, à un nombre et obtenir un nombre, pourquoi ne pouvait-on considérer dx comme un nombre à part entière ? Ceci posait une question ontologique : si dx n'est pas un nombre, qu'est-il réellement ? Et si les infinitésimaux n'existent pas, comment les raisonnements qui en font usage pourraient-ils être corrects ?

C'est Cauchy qui au XIXème siècle trouvera une solution à ce problème en systématisant l'emploi des *quantificateurs*, ce qui permettra de remplacer les infinitésimaux par la notion de limite et évacuera ainsi toute question ontologique (que les mathématiques ne sauraient résoudre [4]). De plus, ces éléments du langage mathématique commençaient à se répandre dans l'ensemble des sciences ; son universalisme apparent fut donc l'objet de spéculations qui menèrent au développement de la logique moderne, également appelée *logique symbolique*.

La logique propositionnelle en fait bien sûr partie, mais la logique symbolique ne dépasse la logique aristotélicienne que lorsqu'on y introduit les variables et les quantificateurs, ces outils indispensables au raisonnement mathématique. La logique la plus simple qui en fait usage est la logique du premier ordre, que nous introduisons sur un exemple.

EXEMPLE 3.1 La logique du premier ordre permet d'énoncer la notion de continuité d'une fonction : f est continue en un point a si

$$\forall \epsilon > 0, \exists \eta > 0, \forall x, |x - a| < \eta \Rightarrow |f(x) - f(a)| < \epsilon.$$

Ici on quantifie implicitement dans l'ensemble des réels ou seulement dans les réels strictement positifs. On peut traduire cette définition en utilisant uniquement la quantification dans les réels :

$$\forall \epsilon \left[\epsilon > 0 \Rightarrow \exists \eta \left[\eta > 0 \wedge \forall x \left(|x - a| < \eta \Rightarrow |f(x) - f(a)| < \epsilon\right)\right]\right]. \qquad \lozenge$$

3. Il semble évident que l'emploi d'un alphabet, voire la confrontation avec des alphabets distincts, grec et arabe, fut essentiel dans le développement du langage mathématique, de par sa nature symbolique. On peut y trouver une explication à la stagnation de la science chinoise, qui était pourtant très en avance jusqu'à la Renaissance.

4. Ce point est tellement peu controversé aujourd'hui qu'on a pu dépenser 4 milliards d'euros pour construire le LHC afin de résoudre une simple question ontologique : le boson de Higgs existe-t-il ? Même les « décideurs » n'ont osé penser que les mathématiques seules pourraient y suffire.

3.1 Termes, atomes et formules du premier ordre

On commence par définir le langage formel qui sera utilisé. Ce langage dépendra évidemment des symboles autorisés.

La quantification se fait dans un ensemble (réels, entiers, etc.). On voit dans l'exemple précédent qu'on a besoin de fonctions dans cet ensemble (soustraction, valeur absolue). On se donne donc un ensemble Σ^{F} dont les éléments sont appelés *symboles de fonction*, et à chacun de ces symboles on associe un entier appelé son *arité* ; c'est le nombre d'arguments des fonctions. Pour tout $n \in \mathbb{N}$ on note Σ^{F}_n l'ensemble des $f \in \Sigma^{\mathrm{F}}$ tels que l'arité de f est n. Les éléments de Σ^{F}_0 sont appelés *constantes*. On a clairement

$$\Sigma^{\mathrm{F}} = \biguplus_{n\in\mathbb{N}} \Sigma^{\mathrm{F}}_n.$$

On a également besoin d'un ensemble de *variables*, qu'on note $\mathcal{V}$; on le suppose disjoint de Σ^{F} et infini dénombrable. Comme on a besoin de composer les fonctions (valeur absolue de la différence de etc.), on définit inductivement un langage $\mathrm{T}_{\Sigma^{\mathrm{F}}}$ dont les éléments sont appelés *termes du premier ordre*, ou simplement *termes* :

Définition 3.2 $\mathrm{T}_{\Sigma^{\mathrm{F}}}$ est le plus petit langage sur le vocabulaire $\mathcal{V} \uplus \Sigma^{\mathrm{F}} \uplus \{\text{“(”}, \text{“)”}, \text{“,”}\}$ [5] tel que

- $\mathcal{V} \subseteq \mathrm{T}_{\Sigma^{\mathrm{F}}}$,
- $\Sigma^{\mathrm{F}}_0 \subseteq \mathrm{T}_{\Sigma^{\mathrm{F}}}$ et
- pour tous $n > 0 \in \mathbb{N}$, $f \in \Sigma^{\mathrm{F}}_n$ et $t_1, \ldots, t_n \in \mathrm{T}_{\Sigma^{\mathrm{F}}}$, le mot $f(t_1, \ldots, t_n)$ est dans $\mathrm{T}_{\Sigma^{\mathrm{F}}}$.

Pour tout $t \in \mathrm{T}_{\Sigma^{\mathrm{F}}}$ on note $\mathrm{Var}(t)$ l'ensemble des variables de t ; la fonction Var est définie inductivement par :

- pour tout $x \in \mathcal{V}$, $\mathrm{Var}(x) = \{x\}$,
- pour tout $a \in \Sigma^{\mathrm{F}}_0$, $\mathrm{Var}(a) = \emptyset$,
- pour tous $n > 0$, $f \in \Sigma^{\mathrm{F}}_n$ et $t_1, \ldots, t_n \in \mathrm{T}_{\Sigma^{\mathrm{F}}}$, $\mathrm{Var}(f(t_1, \ldots, t_n)) = \bigcup_{i=1}^{n} \mathrm{Var}(t_i)$.

On note $\overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}} = \{t \in \mathrm{T}_{\Sigma^{\mathrm{F}}} \mid \mathrm{Var}(t) = \emptyset\}$, les éléments de cet ensemble sont appelés *termes fermés*. △

Exemple 3.3 Posons $\Sigma^{\mathrm{F}}_0 = \{a, b, c\}$, $\Sigma^{\mathrm{F}}_1 = \{g\}$ et $\Sigma^{\mathrm{F}}_2 = \{f\}$. Les symboles a, b, c sont donc des constantes, g un symbole de fonction d'arité 1 et f un symbole de fonction d'arité 2. Soit $\{x, y, z\} \subset \mathcal{V}$, alors :

- a, $g(b)$, $f(b, g(c))$ et $g(f(g(a), g(b)), c)$ sont des termes fermés (qui sont donc dans $\overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}}$),
- $f(x, y)$, $g(f(x, g(y)))$ et $f(a, g(g(z)))$ sont des termes non fermés,
- $\mathrm{Var}(f(g(x), f(z, x))) = \{x, z\}$. ◊

5. Les parenthèses et la virgule ne sont pas indispensables mais améliorent la lisibilité.

EXEMPLE 3.4 Soient $\{\epsilon, \eta, x\} \subset \mathcal{V}$, $\Sigma_0^{\mathrm{F}} = \{a, 0\}$, $\Sigma_1^{\mathrm{F}} = \{f, abs\}$, $\Sigma_2^{\mathrm{F}} = \{diff\}$, alors

$$f(a) \in \overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}} \quad \text{et} \quad abs(diff(f(x), f(a))) \in \mathrm{T}_{\Sigma^{\mathrm{F}}}. \qquad \Diamond$$

EXERCICE 32. *Définir les notions de profondeur et de taille d'un terme.*

Le résultat ci-dessous est une conséquence directe de la définition inductive des termes, et caractérise les ensembles de termes fermés non vides.

Théorème 3.5 $\overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}} = \emptyset$ *si et seulement si* $\Sigma_0^{\mathrm{F}} = \emptyset$.

PREUVE. Comme $\Sigma_0^{\mathrm{F}} \subseteq \overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}}$, si $\Sigma_0^{\mathrm{F}} \neq \emptyset$ alors $\overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}} \neq \emptyset$. Réciproquement, si $\Sigma_0^{\mathrm{F}} = \emptyset$ alors on montre facilement que $\forall t \in \mathrm{T}_{\Sigma^{\mathrm{F}}}$, $\mathrm{Var}(t) \neq \emptyset$ par induction sur t (c'est vrai pour $t \in \mathcal{V}$ et donc pour $t = f(t_1, \ldots, t_n)$), et on a donc $\overline{\mathrm{T}}_{\Sigma^{\mathrm{F}}} = \emptyset$. •

Pour exprimer la continuité d'une fonction en logique du premier ordre, il nous manque encore une notation pour la relation '$<$'. On se donne donc un ensemble Σ^{P} disjoint de Σ^{F} et $\mathcal{V}$ dont les éléments seront utilisés pour représenter des relations. Ces éléments sont appelés *symboles de prédicats*, et à chacun de ces symboles on associe un entier appelé son *arité* ; c'est le nombre d'arguments des relations (ou prédicats). Pour tout $n \in \mathbb{N}$ on note Σ_n^{P} l'ensemble des $p \in \Sigma^{\mathrm{P}}$ tels que l'arité de p est n. On a donc

$$\Sigma^{\mathrm{P}} = \biguplus_{n \in \mathbb{N}} \Sigma_n^{\mathrm{P}}.$$

Définition 3.6 On appelle *signature du premier ordre* l'ensemble $\Sigma = \Sigma^{\mathrm{F}} \uplus \Sigma^{\mathrm{P}}$. $\triangle$

Remarque. Par commodité, dans la suite, nous noterons simplement T_Σ au lieu de $\mathrm{T}_{\Sigma^{\mathrm{F}}}$. De plus, nous supposerons que l'ensemble Σ est au plus dénombrable.

Comme expliqué précédemment, le but des symboles de prédicats est de représenter des relations. Plus précisément, ces symboles permettront de représenter des relations entre les objets représentés par les termes. Nous définissons maintenant la façon de relier des termes par des symboles de prédicats.

Définition 3.7 Soit A_Σ le langage sur le vocabulaire $\mathcal{V} \uplus \Sigma \uplus \{\text{“(”}, \text{“)”}, \text{“,”}\}$ dont les mots sont

— les éléments de Σ_0^{P}, et

— les mots $p(t_1, \ldots, t_n)$, pour tout $n > 0 \in \mathbb{N}$, $p \in \Sigma_n^{\mathrm{P}}$ et $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$.

Les éléments de A_Σ sont appelés *atomes* ou *formules atomiques*. Pour tout atome $\alpha \in \mathrm{A}_\Sigma$ on note $\mathrm{Var}(\alpha)$ l'ensemble des variables de α ; cet ensemble est défini inductivement par :

— si $\alpha \in \Sigma_0^{\mathrm{P}}$ alors $\mathrm{Var}(\alpha) = \emptyset$,

— sinon $\alpha = p(t_1, \ldots, t_n)$ et alors $\mathrm{Var}(\alpha) = \bigcup_{i=1}^{n} \mathrm{Var}(t_i)$.

On note $\overline{\mathrm{A}}_\Sigma = \{\alpha \in \mathrm{A}_\Sigma \mid \mathrm{Var}(\alpha) = \emptyset\}$; ce sont les *atomes fermés*. $\triangle$

EXERCICE 33. *Le Théorème 3.5 donne une caractérisation des ensembles de termes fermés qui sont vides. Qu'en est-il des ensembles d'atomes fermés ? En procédant comme dans la preuve du Théorème 3.5, montrer que si* $\overline{\mathrm{T}}_\Sigma = \emptyset$ *alors* $\overline{\mathrm{A}}_\Sigma = \Sigma_0^{\mathrm{P}}$.

EXEMPLE 3.8 Posons $\Sigma_2^{\mathrm{P}} = \{inf\}$, inf est donc un prédicat d'arité 2 ; on a alors $inf(abs(diff(f(x), f(a))), \epsilon) \in \mathrm{A}_\Sigma$. ◊

Les définitions précédentes permettent de modéliser des objets, ainsi que des fonctions et des relations entre ces objets. On peut maintenant définir le langage des formules.

Définition 3.9 F_Σ est le plus petit langage sur le vocabulaire $\mathrm{V}_\Sigma = \mathcal{V} \uplus \Sigma \uplus \{\text{“(”}, \text{“)”}, \text{“,”}\} \cup \{\blacksquare, \square, \vee, \wedge, \neg, \Rightarrow, \Leftrightarrow, \forall, \exists\}$ tel que :
- $\mathrm{A}_\Sigma \cup \{\blacksquare, \square\} \subseteq \mathrm{F}_\Sigma$,
- pour tous $\varphi, \psi \in \mathrm{F}_\Sigma$ et $x \in \mathcal{V}$ on a

$$\{\neg\varphi,\ (\varphi \vee \psi),\ (\varphi \wedge \psi),\ (\varphi \Rightarrow \psi),\ (\varphi \Leftrightarrow \psi),\ \forall x\, \varphi,\ \exists x\, \varphi\} \subseteq \mathrm{F}_\Sigma.$$

Les mots de ce langage sont appelés *formules du premier ordre*. Le symbole $\forall$ est le *quantificateur universel* ; le symbole $\exists$ est le *quantificateur existentiel*.

On appelle *matrice* toute formule sans quantificateur, et on note M_Σ l'ensemble des matrices de F_Σ. △

EXEMPLE 3.10 En suivant les exemples précédents, le mot suivant est une formule de F_Σ :

$$\forall\epsilon\,[inf(0,\epsilon) \Rightarrow \exists\eta\,[inf(0,\eta) \wedge \forall x\,(inf(abs(diff(x,a)),\eta) \Rightarrow inf(abs(diff(f(x),f(a))),\epsilon))]].$$

Cette formule du premier ordre est à comparer à la formule mathématique exprimant la continuité d'une fonction f dans $\mathbb{R}$ en un point a :

$$\forall\epsilon\,[\epsilon > 0 \Rightarrow \exists\eta\,[\eta > 0 \wedge \forall x\,(|x-a| < \eta \Rightarrow |f(x)-f(a)| < \epsilon)]].$$ ◊

EXERCICE 34. *Montrer que les ensembles* T_Σ, F_Σ *sont infinis dénombrables et que l'ensemble* A_Σ *est au plus dénombrable. Donner une condition nécessaire et suffisante pour que* A_Σ *soit fini.*

Nous n'avons pas défini l'ensemble des variables d'une formule comme nous l'avions fait pour les termes. La raison est que nous devons distinguer les variables qui sont quantifiées de celles qui ne le sont pas, ce qui nécessite une définition à part.

Définition 3.11 Soient $\varphi, \psi \in \mathrm{F}_\Sigma$ et $x \in \mathcal{V}$ tels que $\forall x\, \varphi$ ou $\exists x\, \varphi$ soit une sous-formule de ψ (c'est-à-dire un sous-mot de ψ qui est une formule), alors toute occurrence de x dans φ est dite *liée dans* ψ. Toute occurrence d'une variable dans ψ qui n'est pas liée dans ψ est dite *libre* dans ψ. Toute variable qui a au moins une occurrence libre dans ψ est une *variable libre* de ψ. On note $\mathrm{VL}(\psi)$ l'ensemble des variables libres de ψ.

On note $\overline{\mathrm{F}}_\Sigma = \{\varphi \in \mathrm{F}_\Sigma \mid \mathrm{VL}(\varphi) = \emptyset\}$; ces formules sont dites *fermées*. De même on note $\overline{\mathrm{M}}_\Sigma = \mathrm{M}_\Sigma \cap \overline{\mathrm{F}}_\Sigma$. △

Exercice 35. *Donner une définition inductive de l'ensemble des variables libres d'une formule de* F_Σ.

Le lecteur a sûrement remarqué que mis à part les quantificateurs, les formules du premier ordre sont construites avec les mêmes connecteurs que les formules propositionnelles. Cela nous permet de considérer certaines formules de F_Σ comme des formules propositionnelles, ce qui nous sera indispensable afin d'utiliser les résultats du Chapitre 2.

Il est clair que toute formule propositionnelle dans $\mathrm{P}(\Sigma_0^{\mathrm{P}})$ (c'est-à-dire l'ensemble des formules propositionnelles construites sur l'ensemble de symboles propositionnels Σ_0^{P}, voir la Définition 2.1) est une formule du premier ordre : $\mathrm{P}(\Sigma_0^{\mathrm{P}}) \subseteq \mathrm{F}_\Sigma$. Si $\overline{\mathrm{T}}_\Sigma = \emptyset$, on a même l'égalité $\mathrm{P}(\Sigma_0^{\mathrm{P}}) = \overline{\mathrm{M}}_\Sigma$, puisque dans ce cas $\overline{\mathrm{A}}_\Sigma = \Sigma_0^{\mathrm{P}}$.

Mais il est plus intéressant de construire les formules propositionnelles sur un ensemble de symboles plus large que Σ_0^{P}. Si on n'utilise pas de quantificateurs, alors on montre facilement que les matrices sont des formules propositionnelles construites sur les formules atomiques considérées comme des symboles propositionnels.

Théorème 3.12 $\mathrm{M}_\Sigma = \mathrm{P}(\mathrm{A}_\Sigma)$.

Preuve. On montre par induction sur φ que $\forall \varphi \in \mathrm{M}_\Sigma$ on a $\varphi \in \mathrm{P}(\mathrm{A}_\Sigma)$, en suivant la Définition 3.9.

— Si $\varphi \in \mathrm{A}_\Sigma$, ceci est évident puisque $\mathrm{A}_\Sigma \subseteq \mathrm{P}(\mathrm{A}_\Sigma)$ d'après la Définition 2.1.
— On a $\blacksquare, \square \in \mathrm{P}(\mathrm{A}_\Sigma)$.
— Si $\varphi, \psi \in \mathrm{M}_\Sigma$ alors $\varphi, \psi \in \mathrm{P}(\mathrm{A}_\Sigma)$ par hypothèse d'induction, et on en déduit que $\neg\varphi, \varphi \vee \psi, \ldots$ sont toutes dans $\mathrm{P}(\mathrm{A}_\Sigma)$.

L'induction est terminée et prouve que $\mathrm{M}_\Sigma \subseteq \mathrm{P}(\mathrm{A}_\Sigma)$. L'inclusion réciproque s'obtient de même en prouvant par induction sur f que $\forall f \in \mathrm{P}(\mathrm{A}_\Sigma)$ on a $f \in \mathrm{M}_\Sigma$. •

Corollaire 3.13 $\overline{\mathrm{M}}_\Sigma = \mathrm{P}(\overline{\mathrm{A}}_\Sigma)$.

Preuve. On a donc $\overline{\mathrm{M}}_\Sigma = \mathrm{P}(\mathrm{A}_\Sigma) \cap \overline{\mathrm{F}}_\Sigma = \{\varphi \in \mathrm{P}(\mathrm{A}_\Sigma) \mid \mathrm{VL}(\varphi) = \emptyset\}$. Il est évident que pour tout $\varphi \in \mathrm{P}(\overline{\mathrm{A}}_\Sigma)$ on a $\mathrm{VL}(\varphi) = \emptyset$, donc $\mathrm{P}(\overline{\mathrm{A}}_\Sigma) \subseteq \overline{\mathrm{M}}_\Sigma$. De plus, dans tout $\varphi \in \mathrm{P}(\mathrm{A}_\Sigma) \setminus \mathrm{P}(\overline{\mathrm{A}}_\Sigma)$ apparaît au moins un atome $\alpha \in \mathrm{A}_\Sigma \setminus \overline{\mathrm{A}}_\Sigma$, donc tel que $\mathrm{Var}(\alpha) \neq \emptyset$, et on a donc $\mathrm{VL}(\varphi) \neq \emptyset$ ce qui prouve que $\overline{\mathrm{M}}_\Sigma \subseteq \mathrm{P}(\overline{\mathrm{A}}_\Sigma)$. •

On a donc évidemment $\mathrm{P}(\overline{\mathrm{A}}_\Sigma) \subseteq \mathrm{F}_\Sigma$.

3.2 Interprétations, valuations, substitutions

La formule de l'Exemple 3.10 n'est qu'un mot dans un langage, et même s'il y a de fortes similitudes entre cette formule et la formule mathématique exprimant la continuité de f dans $\mathbb{R}$ en a, on ne peut pas dire que la formule logique définit précisément cette notion. Pour cela il faut donner un sens (une interprétation) aux symboles f, abs, $diff$ etc. et bien sûr préciser qu'on se place dans $\mathbb{R}$; il faut donc définir un contexte.

Définition 3.14 Soit I_Σ la classe [6] des doublets $\mathcal{I} = \langle D, I\rangle$ tels que :
- D est un ensemble non vide, appelé *domaine* (ou domaine de discours) de $\mathcal{I}$,
- I est une fonction qui :
  - à tout $a \in \Sigma_0^{\mathrm{F}}$ associe un élément de D, qu'on note $a^{\mathcal{I}}$,
  - à tout $f \in \Sigma_n^{\mathrm{F}}$ pour $n > 0$ associe une fonction de D^n dans D, notée $f^{\mathcal{I}}$,
  - à tout $p \in \Sigma_0^{\mathrm{P}}$ associe un élément de $\{\mathrm{V}, \mathrm{F}\}$, noté $p^{\mathcal{I}}$,
  - à tout $p \in \Sigma_n^{\mathrm{P}}$ pour $n > 0$ associe une fonction de D^n dans $\{\mathrm{V}, \mathrm{F}\}$, notée $p^{\mathcal{I}}$.

Les éléments de I_Σ sont appelés *interprétations* ou *modèles du premier ordre*. △

Dans la suite, étant donnée une interprétation $\mathcal{I}$, on notera parfois également $\mathcal{I}$ le domaine de $\mathcal{I}$. On pourra donc écrire $v \in \mathcal{I}$ ou bien $n = |\mathcal{I}|$ sans ambiguïté possible.

La Définition 3.14 montre que nous autorisons un ensemble d'interprétations possibles bien plus riche que dans la logique propositionnelle. Par exemple, une constante peut être interprétée comme étant n'importe quel élément d'un domaine qui peut lui-même être n'importe quel ensemble. Les formules valides sous une telle variabilité mériteront le titre de vérités universelles !

En toute rigueur on peut cependant s'inquiéter d'une limitation imposée dans la Définition 3.14 ; le fait de ne pouvoir choisir l'ensemble vide comme domaine. Or, la pratique des mathématiques impose parfois de quantifier une variable dans un ensemble qui peut être vide. Dans ce cas la quantification existentielle est forcément fausse, et dualement la quantification universelle est forcément vraie. Par exemple, la proposition $\exists x \in E,\ (p(x) \vee \neg p(x))$ est fausse lorsque $E = \emptyset$. Le problème est qu'elle est vraie lorsque $E \neq \emptyset$. L'ensemble vide apparaît donc comme un point singulier pour de nombreuses propriétés [7], et on le considère donc comme un cas dégénéré qu'il faut exclure. Cela n'est pas réellement restreignant dans la mesure où on peut représenter l'ensemble vide par une propriété toujours fausse ; la formule ci-dessus peut être remplacée par $\exists x\,(E(x) \wedge (p(x) \vee \neg p(x)))$ (avec $E \in \Sigma_1^{\mathrm{P}}$), dont la valeur de

6. Une *classe* peut contenir des éléments mais ne peut pas être contenue dans un ensemble ou une classe, on ne peut donc pas construire une interprétation dont le domaine serait I_Σ puisque ce n'est pas un ensemble ; cela permet d'éviter des contradictions comme le paradoxe de Russell tout en utilisant les relations $\in$, $\subset$, etc. en les adaptant implicitement à la notion de classe.

7. Un autre exemple très gênant est que $\forall x\, p$ ne serait pas équivalent à p (où $p \in \Sigma_0^{\mathrm{P}}$) si on pouvait quantifier dans un domaine vide.

vérité dépend de l'interprétation de E, qui peut être la propriété toujours fausse sur D.

Mais avant d'attribuer une valeur de vérité à une formule dans une interprétation, nous devons déjà y définir la valeur des termes. Or ceux-ci peuvent contenir des variables, comme $abs(diff(x, a))$ ci-dessus. Il est donc nécessaire d'attribuer une valeur aux variables.

Définition 3.15 Soit $\mathcal{I} = \langle D, I\rangle \in \mathrm{I}_\Sigma$, on appelle *valuation dans* $\mathcal{I}$ toute fonction (totale) de $\mathcal{V}$ dans D (donc tout élément de $D^{\mathcal{V}}$). Par abus de notation, on notera $\mathcal{I}^{\mathcal{V}}$ l'ensemble des valuations dans $\mathcal{I}$. Soit $\theta \in \mathcal{I}^{\mathcal{V}}$ une valuation dans $\mathcal{I}$, on définit inductivement la *valeur* d'un terme t dans $\mathcal{I}, \theta$, qu'on note $[\![t]\!]^{\mathcal{I}}_\theta$, par :

— $\forall x \in \mathcal{V}$, $[\![x]\!]^{\mathcal{I}}_\theta = \theta(x)$,
— $\forall a \in \Sigma^{\mathrm{F}}_0$, $[\![a]\!]^{\mathcal{I}}_\theta = a^{\mathcal{I}}$ et
— $\forall n > 0$, $\forall f \in \Sigma^{\mathrm{F}}_n$, $\forall t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, $[\![f(t_1, \ldots, t_n)]\!]^{\mathcal{I}}_\theta = f^{\mathcal{I}}([\![t_1]\!]^{\mathcal{I}}_\theta, \ldots, [\![t_n]\!]^{\mathcal{I}}_\theta)$. △

Exemple 3.16 Soit $\mathcal{I}$ de domaine $\mathbb{R}$ telle que $a^{\mathcal{I}} = 1$, $abs^{\mathcal{I}}$ est la valeur absolue dans $\mathbb{R}$ et $diff^{\mathcal{I}}$ est la différence de deux réels. Soit θ une valuation dans $\mathbb{R}$ telle que $\theta(x) = 0$, alors

$$\begin{aligned}
[\![abs(diff(x, a))]\!]^{\mathcal{I}}_\theta &= |[\![diff(x, a)]\!]^{\mathcal{I}}_\theta| \\
&= |[\![x]\!]^{\mathcal{I}}_\theta - [\![a]\!]^{\mathcal{I}}_\theta| \\
&= |\theta(x) - a^{\mathcal{I}}| \\
&= |0 - 1| = |-1| = 1 \quad \text{(phase d'évaluation).}
\end{aligned}$$ ◊

Exemple 3.17 Soit $t = f(a, g(x))$ et considérons l'interprétation $\mathcal{I} = \langle D, I\rangle$, où $D = \mathbb{N}$ et $a^{\mathcal{I}} = 2$, $f^{\mathcal{I}} : x, y \mapsto 5x + 3y$ et $g^{\mathcal{I}} : x \mapsto x - 2$. Si θ_1 est une valuation telle que $\theta_1(x) = 1$ et θ_2 est une valuation telle que $\theta_2(x) = -2$, alors $[\![t]\!]^{\mathcal{I}}_{\theta_1} = 7$ et $[\![t]\!]^{\mathcal{I}}_{\theta_2} = -2$. ◊

Il est évident que la valeur d'un terme fermé ne dépend pas des valeurs attribuées aux variables. Plus généralement, la valeur d'un terme ne dépend que des valeurs attribuées aux variables qui y apparaissent, ce qui est énoncé dans l'Exercice 36.

Exercice 36. *Prouver que pour tous* $\mathcal{I} \in \mathrm{I}_\Sigma$, $\theta, \theta' \in \mathcal{I}^{\mathcal{V}}$ *et* $t \in \mathrm{T}_\Sigma$, *si* $\theta =_{|\mathrm{Var}(t)} \theta'$ *alors* $[\![t]\!]^{\mathcal{I}}_\theta = [\![t]\!]^{\mathcal{I}}_{\theta'}$.

On utilisera la notation suivante dans le cas où le terme t ne contient pas de variables (autrement dit, dans le cas où $t \in \overline{\mathrm{T}}_\Sigma$) :

Définition 3.18 Soient $\mathcal{I} \in \mathrm{I}_\Sigma$ et $t \in \overline{\mathrm{T}}_\Sigma$, on note $[\![t]\!]^{\mathcal{I}}$ pour $[\![t]\!]^{\mathcal{I}}_\theta$, où θ est une valuation quelconque [8] dans $\mathcal{I}$. △

8. Il en existe au moins une puisque le domaine de $\mathcal{I}$ est non vide, et n'importe quelle valuation peut être choisie.

Nous présentons maintenant un exemple de Σ^{F}-interprétation qui peut sembler simple, mais qui est très important et que nous réutiliserons ; nous en faisons donc une définition.

Définition 3.19 On note $\mathcal{T}_\Sigma$ la Σ^{F}-interprétation dont le domaine est le langage T_Σ, et telle que :
- $\forall a \in \Sigma^{\mathrm{F}}_0$, $a^{\mathcal{T}_\Sigma} = a$,
- $\forall n > 0$, $\forall f \in \Sigma^{\mathrm{F}}_n$, $f^{\mathcal{T}_\Sigma}$ est la fonction qui à $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$ associe le terme $f(t_1, \ldots, t_n)$.

$\mathcal{T}_\Sigma$ est appelé l'*algèbre des termes*.

Comme pour tout $f \in \Sigma^{\mathrm{F}}_n$ on a $f^{\mathcal{T}_\Sigma}(\overline{\mathrm{T}}^n_\Sigma) \subseteq \overline{\mathrm{T}}_\Sigma$, la restriction de $\mathcal{T}_\Sigma$ à $\overline{\mathrm{T}}_\Sigma$ est une sous-algèbre de $\mathcal{T}_\Sigma$ appelée *algèbre des termes fermés* et notée $\overline{\mathcal{T}}_\Sigma$. △

Les valuations dans l'algèbre des termes joueront un rôle important dans la suite, ce qui explique l'existence d'une terminologie particulière les concernant.

Définition 3.20 Une valuation $\sigma \in \mathrm{T}^{\mathcal{V}}_\Sigma$ dans l'algèbre des termes est appelée *substitution*, et pour tout terme $t \in \mathrm{T}_\Sigma$, on note $t\sigma$ le terme $[\![t]\!]^{\mathcal{T}_\Sigma}_\sigma$; on dit que $t\sigma$ est une *instance* de t. Le *domaine* de σ est l'ensemble $\mathrm{Dom}(\sigma) = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$. Si $\mathrm{Dom}(\sigma)$ est un ensemble fini $\{x_1, \ldots, x_n\}$ la substitution pourra être notée $\sigma = \{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}$.

Si $\sigma \in \overline{\mathrm{T}}^{\mathcal{V}}_\Sigma$ on dit que σ est une substitution *fermée* et $t\sigma \in \overline{\mathrm{T}}_\Sigma$ une instance *fermée* de t. △

Il est donc clair que si σ est fermée alors $t\sigma$ ne contient aucune variable, quelque soit le terme t.

EXEMPLE 3.21 Soit σ une substitution fermée telle que $\sigma(x) = a$, alors

$$\begin{aligned}
[\![abs(diff(x,a))]\!]^{\overline{\mathcal{T}}_\Sigma}_\sigma &= abs^{\overline{\mathcal{T}}_\Sigma}([\![diff(x,a)]\!]^{\overline{\mathcal{T}}_\Sigma}_\sigma) \\
&= abs^{\overline{\mathcal{T}}_\Sigma}(diff^{\overline{\mathcal{T}}_\Sigma}([\![x]\!]^{\overline{\mathcal{T}}_\Sigma}_\sigma, [\![a]\!]^{\overline{\mathcal{T}}_\Sigma}_\sigma)) \\
&= abs^{\overline{\mathcal{T}}_\Sigma}(diff^{\overline{\mathcal{T}}_\Sigma}(\sigma(x), a^{\overline{\mathcal{T}}_\Sigma})) \\
&= abs^{\overline{\mathcal{T}}_\Sigma}(diff^{\overline{\mathcal{T}}_\Sigma}(a,a)) = abs^{\overline{\mathcal{T}}_\Sigma}(diff(a,a)) = abs(diff(a,a)). \ \Diamond
\end{aligned}$$

EXERCICE 37. *Soit $t = f(x, g(a, h(x,y), b), h(z,a))$ et considérons la substitution $\sigma = \{x \mapsto b, y \mapsto h(b,a), z \mapsto a\}$. Calculer $t\sigma$. Trouver une substitution σ' telle que $t\sigma' = f(g(a,b), g(a, h(g(a,b),b), b), h(h(x,a),a))$.*

Il est évident que pour tout terme fermé $t \in \overline{\mathrm{T}}_\Sigma$ on a $[\![t]\!]^{\overline{\mathcal{T}}_\Sigma} = t$. Pour un terme t avec variables, $t\sigma$ est le terme obtenu en remplaçant dans t chaque occurrence de variable x par le terme $\sigma(x)$ de façon *simultanée*.

EXEMPLE 3.22 Si $t = g(x,y)$, $\sigma = \{x \mapsto f(y), y \mapsto f(x)\}$, alors $t\sigma = g(f(y), f(x))$. Par contre, on a $t[f(y)/x][f(x)/y] = g(f(f(x)), f(x)) \neq t\sigma$. ♢

La notion de valuation permet donc de définir précisément l'interprétation de n'importe quel terme, y compris celle des termes qui contiennent des variables. Il est maintenant possible de définir l'évaluation des atomes.

Définition 3.23 Soient $\mathcal{I} \in \mathrm{I}_\Sigma$ et $\theta \in \mathcal{I}^\mathcal{V}$, on définit la *valeur* d'un atome $\alpha \in \mathrm{A}_\Sigma$ dans $\mathcal{I}, \theta$, qu'on note $[\![\alpha]\!]^\mathcal{I}_\theta$, par :
— $\forall p \in \Sigma^\mathrm{P}_0$, $[\![p]\!]^\mathcal{I}_\theta = p^\mathcal{I}$,
— $\forall n > 0$, $\forall p \in \Sigma^\mathrm{P}_n$, $\forall t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, $[\![p(t_1, \ldots, t_n)]\!]^\mathcal{I}_\theta = p^\mathcal{I}([\![t_1]\!]^\mathcal{I}_\theta, \ldots, [\![t_n]\!]^\mathcal{I}_\theta)$. △

Comme ci-dessus, si $\alpha \in \overline{\mathrm{A}}_\Sigma$ est fermé alors sa valeur est indépendante d'une valuation, et on note donc $[\![\alpha]\!]^\mathcal{I}$ pour $[\![\alpha]\!]^\mathcal{I}_\theta$ où θ est une valuation quelconque.

Définition 3.24 On note $[\![.]\!]^\mathcal{I}$ la fonction qui à $\alpha \in \overline{\mathrm{A}}_\Sigma$ associe $[\![\alpha]\!]^\mathcal{I} \in \{\mathrm{V}, \mathrm{F}\}$. △

On remarque que $[\![.]\!]^\mathcal{I} \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$; c'est donc une interprétation propositionnelle pour les formules de $\mathrm{P}(\overline{\mathrm{A}}_\Sigma) = \overline{\mathrm{M}}_\Sigma$.

La notion de substitution s'étend également aux atomes dans A_Σ :

Définition 3.25 Pour toute substitution $\sigma \in \mathrm{T}^\mathcal{V}_\Sigma$ et $\alpha \in \mathrm{A}_\Sigma$, on note $\alpha\sigma$ la formule atomique définie par :
— $\forall p \in \Sigma^\mathrm{P}_0$, $p\sigma = p$,
— $\forall n > 0$, $\forall p \in \Sigma^\mathrm{P}_n$, $\forall t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, $p(t_1, \ldots, t_n)\sigma = p(t_1\sigma, \ldots, t_n\sigma)$.
La formule $\alpha\sigma \in \mathrm{A}_\Sigma$ est une *instance* de α ; c'est une *instance fermée* si σ est fermée. △

Exemple 3.26 Si $a, b \in \Sigma^\mathrm{F}_0$ et $p \in \Sigma^\mathrm{P}_1$, alors $p(a), p(b) \in \overline{\mathrm{A}}_\Sigma$. Soit $\mathcal{I}$ une interprétation sur le domaine $\{0, 1\}$ telle que $p^\mathcal{I}(0) = \mathrm{F}$, $p^\mathcal{I}(1) = \mathrm{V}$ et $a^\mathcal{I} = b^\mathcal{I} = 1$; on a donc $\mathcal{I} \models_1 p(a) \wedge p(b)$. Soit $J = [\![.]\!]^\mathcal{I}$, on a $J(p(a)) = [\![p(a)]\!]^\mathcal{I} = p^\mathcal{I}(a^\mathcal{I}) = p^\mathcal{I}(1) = \mathrm{V}$ et de même $J(p(b)) = \mathrm{V}$. On a donc $J \models_0 p(a) \wedge p(b)$ (qui est aussi une formule propositionnelle). ♢

Exercice 38. *La fonction de* I_Σ *dans* $\mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ *qui à* $\mathcal{I}$ *associe* $[\![.]\!]^\mathcal{I}$ *est-elle injective ?*

3.3 La relation de satisfaction

Nous allons maintenant définir la relation $\models_1$ entre I_Σ et F_Σ, mais comme les formules contiennent des variables nous devons encore tenir compte de leurs valuations, et nous allons donc définir $\models_1$ dans un premier temps comme une relation ternaire.

Définition 3.27 Soit $\mathcal{I} \in \mathrm{I}_\Sigma$, pour toute valuation $\theta \in \mathcal{I}^\mathcal{V}$, toute variable $x \in \mathcal{V}$ et toute valeur $u \in \mathcal{I}$, on note $\theta[x \mapsto u]$ la valuation θ' définie par : $\theta'(x) = u$ et $\forall y \in \mathcal{V}$, si $y \neq x$ alors $\theta'(y) = \theta(y)$ (autrement dit, θ' est identique à θ sur $\mathcal{V} \setminus \{x\}$).

La relation $\models_1$ est la plus petite relation telle que, pour tout $\mathcal{I} \in \mathrm{I}_\Sigma$ et $\theta \in \mathcal{I}^\mathcal{V}$,

— $\mathcal{I}, \theta \models_1 \blacksquare$ et $\mathcal{I}, \theta \not\models_1 \square$,
— $\forall \alpha \in \mathrm{A}_\Sigma$, on a $\mathcal{I}, \theta \models_1 \alpha$ si et seulement si $[\![\alpha]\!]^{\mathcal{I}}_\theta = \mathrm{V}$,
— $\forall \varphi \in \mathrm{F}_\Sigma$, on a $\mathcal{I}, \theta \models_1 \neg\varphi$ si et seulement si $\mathcal{I}, \theta \not\models_1 \varphi$,
— $\forall \varphi, \psi \in \mathrm{F}_\Sigma$, on a $\mathcal{I}, \theta \models_1 \varphi \vee \psi$ si et seulement si $\mathcal{I}, \theta \models_1 \varphi$ ou $\mathcal{I}, \theta \models_1 \psi$,
— de même pour $\wedge, \Rightarrow, \Leftrightarrow$,
— $\forall \varphi \in \mathrm{F}_\Sigma$, $\forall x \in \mathcal{V}$, on a $\mathcal{I}, \theta \models_1 \forall x\, \varphi$ si et seulement si pour tout $u \in \mathcal{I}$ on a $\mathcal{I}, \theta[x \mapsto u] \models_1 \varphi$,
— $\forall \varphi \in \mathrm{F}_\Sigma$, $\forall x \in \mathcal{V}$, on a $\mathcal{I}, \theta \models_1 \exists x\, \varphi$ si et seulement s'il existe $u \in \mathcal{I}$ tel que $\mathcal{I}, \theta[x \mapsto u] \models_1 \varphi$.

On note $\equiv_1$ l'équivalence logique correspondant à $\models_1$ (cf. Définition 1.11). △

Afin de définir une logique il nous faut une relation de satisfaction binaire. Nous devons donc dégager une condition générale sous laquelle la relation ternaire $\models_1$ que nous avons définie ne dépend pas de son deuxième argument (la valuation). Or nous voyons dans la première partie de la Définition 3.27 que la valuation $\theta[x \mapsto u]$ ne dépend pas de $\theta(x)$, cette valeur étant « masquée » par u. Par conséquent, si dans une formule φ toute variable x n'est utilisée que dans le champ d'un quantificateur $\forall x$ ou $\exists x$, alors la relation $\mathcal{I}, \theta \models_1 \varphi$ ne dépend pas de θ. Plus généralement, cette relation ne dépend que des valeurs attribuées par θ aux variables libres de φ, ce que nous prouvons maintenant.

Lemme 3.28 *Soient $\mathcal{I} \in \mathrm{I}_\Sigma$ une Σ-interprétation, et $\theta, \theta' \in \mathcal{I}^{\mathcal{V}}$ deux valuations. Pour toute formule $\varphi \in \mathrm{F}_\Sigma$, si $\theta =_{|\mathrm{VL}(\varphi)} \theta'$ alors $\mathcal{I}, \theta \models_1 \varphi$ si et seulement si $\mathcal{I}, \theta' \models_1 \varphi$.*

Preuve. Par induction sur φ.
— Si $\varphi \in \mathrm{A}_\Sigma$, alors $\mathrm{VL}(\varphi) = \mathrm{Var}(\varphi)$ et on voit facilement d'après l'Exercice 36 que $[\![\varphi]\!]^{\mathcal{I}}_\theta = [\![\varphi]\!]^{\mathcal{I}}_{\theta'}$, on a donc $\mathcal{I}, \theta \models_1 \varphi$ si et seulement si $\mathcal{I}, \theta' \models_1 \varphi$.
— On a $\mathcal{I}, \theta \models_1 \blacksquare$ et $\mathcal{I}, \theta' \models_1 \blacksquare$. De même, on a $\mathcal{I}, \theta \not\models_1 \square$ et $\mathcal{I}, \theta' \not\models_1 \square$.
— Si $\theta =_{|\mathrm{VL}(\neg\varphi)} \theta'$ alors $\theta =_{|\mathrm{VL}(\varphi)} \theta'$ (puisque $\mathrm{VL}(\neg\varphi) = \mathrm{VL}(\varphi)$), donc $\mathcal{I}, \theta \models_1 \neg\varphi$ ssi $\mathcal{I}, \theta \not\models_1 \varphi$ ssi $\mathcal{I}, \theta' \not\models_1 \varphi$ (par hypothèse d'induction) ssi $\mathcal{I}, \theta \models_1 \neg\varphi$.
— Si $\theta =_{|\mathrm{VL}(\varphi\wedge\psi)} \theta'$ alors $\theta =_{|\mathrm{VL}(\varphi)} \theta'$ et $\theta =_{|\mathrm{VL}(\psi)} \theta'$ (puisque $\mathrm{VL}(\varphi \wedge \psi) = \mathrm{VL}(\varphi) \cup \mathrm{VL}(\psi)$), donc $\mathcal{I}, \theta \models_1 \varphi \wedge \psi$ ssi $\mathcal{I}, \theta \models_1 \varphi$ et $\mathcal{I}, \theta \models_1 \psi$ ssi $\mathcal{I}, \theta' \models_1 \varphi$ et $\mathcal{I}, \theta' \models_1 \psi$ (par h. i.) ssi $\mathcal{I}, \theta' \models_1 \varphi \wedge \psi$.
— Si $\theta =_{|\mathrm{VL}(\forall x\, \varphi)} \theta'$ alors $\forall u \in \mathcal{I}$, $\theta[x \mapsto u] =_{|\mathrm{VL}(\varphi)} \theta'[x \mapsto u]$ (puisque $\mathrm{VL}(\forall x\, \varphi) = \mathrm{VL}(\varphi) \setminus \{x\}$), donc $\mathcal{I}, \theta \models_1 \forall x\, \varphi$ ssi $\forall u \in \mathcal{I}$ on a $\mathcal{I}, \theta[x \mapsto u] \models_1 \varphi$ ssi $\forall u \in \mathcal{I}$ on a $\mathcal{I}, \theta'[x \mapsto u] \models_1 \varphi$ (par h. i.) ssi $\mathcal{I}, \theta' \models_1 \forall x\, \varphi$.
— On procède de même pour $\vee$, $\Rightarrow$, $\Leftrightarrow$ et $\exists$. •

Corollaire 3.29 *Si $\varphi \in \overline{\mathrm{F}}_\Sigma$ alors pour toute interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$ et pour toutes valuations $\theta, \theta' \in \mathcal{I}^{\mathcal{V}}$, on a $\mathcal{I}, \theta \models_1 \varphi$ si et seulement si $\mathcal{I}, \theta' \models_1 \varphi$.*

On arrive donc à la définition suivante :

Définition 3.30 Soient $\mathcal{I} \in \mathrm{I}_\Sigma$ et $\varphi \in \mathrm{F}_\Sigma$, on dit que $\mathcal{I}$ est un *modèle de* φ et on note $\mathcal{I} \models_1 \varphi$ si et seulement si $\varphi \in \overline{\mathrm{F}}_\Sigma$ et $\mathcal{I}, \theta \models_1 \varphi$ pour une valuation $\theta \in \mathcal{I}^\mathcal{V}$ quelconque.

On appelle *logique du premier ordre* la logique $\mathrm{L}^\mathcal{V}_\Sigma = \langle \mathrm{V}_\Sigma, \mathrm{F}_\Sigma, \mathrm{I}_\Sigma, \models_1 \rangle$. △

En comparant les Définitions 2.1 et 3.27, on voit que les connecteurs $\blacksquare, \Box, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ ont la même signification qu'en logique propositionnelle, et on peut donc facilement prouver une correspondance évidente entre ces deux logiques.

Théorème 3.31 *Pour toute matrice fermée $\varphi \in \overline{\mathrm{M}}_\Sigma$ et tout $\mathcal{I} \in \mathrm{I}_\Sigma$, on a $\mathcal{I} \models_1 \varphi$ si et seulement si $[\![.]\!]^\mathcal{I} \models_0 \varphi$.*

Exercice 39. *Prouver le Théorème 3.31.*

Exercice 40. *(R. Caferra).*

1. *Donner un modèle de $\forall x\, p(g(x, f(x)), a)$.*
2. *Donner un modèle et un contre-modèle de $\forall x \forall y (p(f(x, y), a) \Rightarrow p(x, y))$.*
3. *Donner un modèle de $\forall x \forall y \forall z\, (p(x, y) \Rightarrow p(f(x, z), f(y, z)))$.*
4. *Donner un modèle de $\forall x \exists y\, p(x, f(f(x, y), y))$.*
5. *Donner un modèle de $\forall x \forall y\, (p(f(x, a), y) \Rightarrow p(f(y, a), x))$.*
6. *Donner un contre-modèle de $\forall x \exists y\, p(x, y) \Rightarrow \exists y \forall x\, p(x, y)$.*
7. *Donner un modèle et un contre-modèle de : $\forall x \forall y\, [(p(x, y) \wedge \neg q(x, y)) \Rightarrow (\exists z\, (p(x, z) \wedge \neg q(z, x) \wedge p(z, y) \wedge \neg q(z, y)))]$.*
8. *La formule $\forall x\, (p(x) \vee q(x)) \Rightarrow \forall x\, p(x) \vee \forall x\, q(x)$ est-elle valide ?*

Exercice 41. *On considère une constante a et un prédicat p d'arité 1. Prouver que $\forall x\, p(x) \models_1 p(a)$, $p(a) \models_1 \exists x\, p(x)$ et $\models_1 (\forall x\, p(x)) \Rightarrow (\exists y\, p(y))$*

Le Lemme 3.28 exprime le fait que la valeur de vérité d'une formule ne dépend que de l'interprétation et de la valeur des variables libres dans cette formule. Or, si la valeur de vérité de la formule ne dépend pas de la valeur des variables quantifiées, alors elle ne dépend pas non plus des variables qui y sont quantifiées. Deux formules fermées qui ne diffèrent que par les variables utilisées sont donc équivalentes.

On peut donc dans une formule remplacer les variables quantifiées par d'autres, mais à condition de respecter les différences entre variables : on a $\forall x\, \varphi \equiv_1 \forall y\, \varphi[y/x]$ seulement si y n'apparaît pas dans φ.

EXERCICE 42. *Donner un exemple de formule φ telle que $\forall x\,\varphi$ et $\forall y\,\varphi[y/x]$ ne sont pas équivalentes.*

Il faut donc prendre quelques précautions afin de généraliser aux formules la notion d'instance que nous connaissons déjà pour les termes et les formules atomiques.

Définition 3.32 Soit $\sigma \in \mathrm{T}_\Sigma^{\mathcal{V}}$ une substitution, on pose

$$\mathrm{Var}(\sigma) = \{\mathrm{Var}(\sigma(x)) \mid x \in \mathrm{Dom}(\sigma)\}\,.$$

On définit inductivement sur $\varphi \in \mathrm{F}_\Sigma$ la formule $\varphi\sigma$ pour toute substitution σ telle qu'aucune des variables dans $\mathrm{Var}(\sigma)$ n'apparaît liée dans φ :

- Si $\varphi \in \mathrm{A}_\Sigma$ alors $\varphi\sigma \in \mathrm{A}_\Sigma$ est définie dans la Définition 3.23.
- On pose $\blacksquare\sigma = \blacksquare$ et $\square\sigma = \square$.
- Pour tout $\varphi, \psi \in \mathrm{F}_\Sigma$ on définit $(\neg\varphi)\sigma = \neg(\varphi\sigma)$, $(\varphi \wedge \psi)\sigma = \varphi\sigma \wedge \psi\sigma$, $(\varphi \vee \psi)\sigma = \varphi\sigma \vee \psi\sigma$, $(\varphi \Rightarrow \psi)\sigma = \varphi\sigma \Rightarrow \psi\sigma$ et $(\varphi \Leftrightarrow \psi)\sigma = \varphi\sigma \Leftrightarrow \psi\sigma$.
- Pour tout $\varphi \in \mathrm{F}_\Sigma$ et $x \in \mathcal{V}$ on pose [9] $(\forall x\,\varphi)\sigma = \forall x\,\varphi(\sigma[x \mapsto x])$ et $(\exists x\,\varphi)\sigma = \exists x\,\varphi(\sigma[x \mapsto x])$

La formule $\varphi\sigma \in \mathrm{F}_\Sigma$ est une *instance* de φ (et elle est *fermée* si σ est fermée). △

On remarque que $\mathrm{Var}(\sigma[x \mapsto x]) = \mathrm{Var}(\sigma) \setminus \{x\}$ et donc si aucune des variables dans $\mathrm{Var}(\sigma)$ n'apparaît liée dans $\forall x\,\varphi$ (ou dans $\exists x\,\varphi$) alors aucune variable dans $\mathrm{Var}(\sigma[x \mapsto x])$ n'apparaît liée dans φ ; la Définition 3.32 est donc correcte.

EXEMPLE 3.33 Soient $\varphi = (\forall x\,p(x,y)) \vee (\exists y\,p(x,y))$ et $\sigma = \{x \mapsto a, y \mapsto f(z)\}$ où a est une constante et z une variable. On a $\mathrm{Var}(\sigma) = \{z\}$ et z n'est pas liée dans φ, on peut donc appliquer σ à φ :

$$\varphi\sigma = (\forall x\,p(x, f(z))) \vee (\exists y\,p(a,y)).$$

Les occurrences liées de x et y ne sont pas substituées. On remarque que $\mathrm{VL}(\varphi) = \{x,y\} = \mathrm{Dom}(\sigma)$ et $\mathrm{VL}(\varphi\sigma) = \{z\} = \mathrm{Var}(\sigma)$.

Soit $\tau = \{x \mapsto a, y \mapsto f(x)\}$, on a $\mathrm{Dom}(\tau) = \mathrm{Dom}(\sigma)$ mais $\mathrm{Var}(\tau) = \{x\}$ et x apparaît liée dans φ. Si on appliquait une telle substitution à φ on obtiendrait une formule

$$\varphi\tau = (\forall x\,p(x, f(x))) \vee (\exists y\,p(a,y))$$

qui est fermée puisque la nouvelle occurrence de x est capturée par le quantificateur $\forall x$. C'est pour éviter cette *capture de variable* que l'on restreint les substitutions applicables à une formule. ◊

Nous allons également appliquer des substitutions sur des valuations, ce qui nous permettra ensuite de relier ces notions entre elles.

9. Voir la Définition 3.27 pour la notation $\sigma[x \mapsto x]$.

Définition 3.34 Soient une interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$, une valuation $\theta \in \mathcal{I}^\mathcal{V}$ et une substitution $\sigma \in \mathrm{T}_\Sigma^\mathcal{V}$, on définit la valuation $\sigma\theta \in \mathcal{I}^\mathcal{V}$ en posant, pour tout $x \in \mathcal{V}$, $\sigma\theta(x) = [\![\sigma(x)]\!]_\theta^\mathcal{I}$. △

On remarque en particulier que si $\mathcal{I}$ est l'algèbre de termes $\mathcal{T}_\Sigma$, alors θ et $\sigma\theta$ sont des substitutions, et on a donc pour tout $x \in \mathcal{V}$,

$$x(\sigma\theta) = \sigma\theta(x) = [\![\sigma(x)]\!]_\theta^{\mathcal{T}_\Sigma} = \sigma(x)\theta = (x\sigma)\theta.$$

Cette égalité justifie que l'on note $\sigma\theta$ l'instanciation de θ par σ, plutôt que $\theta\sigma$, et bien sûr pour tout terme t on notera $t\sigma\theta$, sans parenthèses, pour $t(\sigma\theta)$ ou $(t\sigma)\theta$ puisque ces deux termes sont égaux.

Exemple 3.35 Soient

$$\begin{aligned} \sigma &= \{x \mapsto f(x,y), y \mapsto f(x,z), z \mapsto a\} \\ \tau &= \{x \mapsto g(x,z), y \mapsto g(x,y), u \mapsto z\} \end{aligned}$$

on a $\sigma\tau = \{x \mapsto f(g(x,z), g(x,y)), y \mapsto f(g(x,z), z), z \mapsto a, u \mapsto z\}$. ◊

Exercice 43. *Soient $\sigma \in \overline{\mathrm{T}}_\Sigma^\mathcal{V}$ et $\tau \in \mathrm{T}_\Sigma^\mathcal{V}$, montrer que $\sigma\tau = \sigma$.*

Nous avons besoin d'un lemme permettant d'assurer que, sous la même condition que dans la Définition 3.32, on peut appliquer une substitution à une valuation sans provoquer de modification analogue à une capture de variable.

Lemme 3.36 *Soient $\mathcal{I} \in \mathrm{I}_\Sigma$ une interprétation et $u \in \mathcal{I}$, soient $\sigma \in \mathrm{T}_\Sigma^\mathcal{V}$ une substitution et $x \in \mathcal{V}$ telles que $x \notin \mathrm{Var}(\sigma)$, alors*

$$(\sigma[x \mapsto x])(\theta[x \mapsto u]) = (\sigma\theta)[x \mapsto u].$$

Preuve. On pose $\sigma' = \sigma[x \mapsto x]$ et $\theta' = \theta[x \mapsto u]$, on a

$$\sigma'\theta'(x) \;=\; [\![\sigma'(x)]\!]_{\theta'}^\mathcal{I} \;=\; [\![x]\!]_{\theta'}^\mathcal{I} \;=\; \theta'(x) \;=\; u \;=\; (\sigma\theta)[x \mapsto u](x),$$

et pour tout $y \in \mathcal{V} \setminus \{x\}$ on a

$$\begin{aligned} (\sigma\theta)[x \mapsto u](y) &= \sigma\theta(y) \\ &= [\![\sigma(y)]\!]_\theta^\mathcal{I} \\ &= [\![\sigma(y)]\!]_{\theta'}^\mathcal{I} \text{ (puisque } x \notin \mathrm{Var}(\sigma(y))) \\ &= [\![\sigma'(y)]\!]_{\theta'}^\mathcal{I} \text{ (puisque } y \neq x) \\ &= \sigma'\theta'(y). \end{aligned}$$

•

On peut maintenant prouver que l'instanciation des variables dans une formule correspond à leur instanciation dans une valuation.

Théorème 3.37 (d'instanciation) *Soient $\varphi \in \mathrm{F}_\Sigma$, $\sigma \in \mathrm{T}_\Sigma^{\mathcal{V}}$, $\mathcal{I} \in \mathrm{I}_\Sigma$ et $\theta \in \mathcal{I}^{\mathcal{V}}$, si aucune variable dans $\mathrm{Var}(\sigma)$ n'apparaît liée dans φ alors on a*

$$\mathcal{I}, \theta \models_1 \varphi\sigma \text{ si et seulement si } \mathcal{I}, \sigma\theta \models_1 \varphi.$$

PREUVE. On montre d'abord par induction sur t que $\forall t \in \mathrm{T}_\Sigma$, on a $[\![t\sigma]\!]_\theta^{\mathcal{I}} = [\![t]\!]_{\sigma\theta}^{\mathcal{I}}$.

— Pour toute variable $x \in \mathcal{V}$, on a $[\![x\sigma]\!]_\theta^{\mathcal{I}} = [\![\sigma(x)]\!]_\theta^{\mathcal{I}} = \sigma\theta(x) = [\![x]\!]_{\sigma\theta}^{\mathcal{I}}$,

— pour toute constante $a \in \Sigma_0^{\mathrm{F}}$, $[\![a\sigma]\!]_\theta^{\mathcal{I}} = [\![a]\!]_\theta^{\mathcal{I}} = a^{\mathcal{I}} = [\![a]\!]_{\sigma\theta}^{\mathcal{I}}$,

— pour tous $n > 0$, $f \in \Sigma_n^{\mathrm{F}}$ et $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, on a

$$\begin{aligned}
[\![f(t_1, \ldots, t_n)\sigma]\!]_\theta^{\mathcal{I}} &= [\![f(t_1\sigma, \ldots, t_n\sigma)]\!]_\theta^{\mathcal{I}} \\
&= f^{\mathcal{I}}([\![t_1\sigma]\!]_\theta^{\mathcal{I}}, \ldots, [\![t_n\sigma]\!]_\theta^{\mathcal{I}}) \\
&= f^{\mathcal{I}}([\![t_1]\!]_{\sigma\theta}^{\mathcal{I}}, \ldots, [\![t_n]\!]_{\sigma\theta}^{\mathcal{I}}) \text{ (par hypothèse d'induction)} \\
&= [\![f(t_1, \ldots, t_n)]\!]_{\sigma\theta}^{\mathcal{I}}.
\end{aligned}$$

On en déduit que pour toute formule atomique $\alpha \in \mathrm{A}_\Sigma$, on a $[\![\alpha\sigma]\!]_\theta^{\mathcal{I}} = [\![\alpha]\!]_{\sigma\theta}^{\mathcal{I}}$. En effet, si $\alpha \in \Sigma_0^{\mathrm{P}}$ alors $[\![\alpha\sigma]\!]_\theta^{\mathcal{I}} = [\![\alpha]\!]_\theta^{\mathcal{I}} = \alpha^{\mathcal{I}} = [\![\alpha]\!]_{\sigma\theta}^{\mathcal{I}}$. Sinon, on a $\alpha = p(t_1, \ldots, t_n)$, donc

$$\begin{aligned}
[\![\alpha\sigma]\!]_\theta^{\mathcal{I}} &= [\![p(t_1\sigma, \ldots, t_n\sigma)]\!]_\theta^{\mathcal{I}} \\
&= p^{\mathcal{I}}([\![t_1\sigma]\!]_\theta^{I}, \ldots, [\![t_n\sigma]\!]_\theta^{\mathcal{I}}) \\
&= p^{\mathcal{I}}([\![t_1]\!]_{\sigma\theta}^{\mathcal{I}}, \ldots, [\![t_n]\!]_{\sigma\theta}^{\mathcal{I}}) \\
&= [\![p(t_1, \ldots, t_n)]\!]_{\sigma\theta}^{\mathcal{I}} = [\![\alpha]\!]_{\sigma\theta}^{\mathcal{I}}.
\end{aligned}$$

On procède ensuite par induction sur φ.

— Si $\varphi \in \mathrm{A}_\Sigma$, on a $\mathcal{I}, \theta \models_1 \varphi\sigma$ ssi $[\![\varphi\sigma]\!]_\theta^{\mathcal{I}} = \mathrm{V}$ ssi $[\![\varphi]\!]_{\sigma\theta}^{\mathcal{I}} = \mathrm{V}$ ssi $\mathcal{I}, \sigma\theta \models_1 \varphi$.

— On a $\mathcal{I}, \theta \models_1 \blacksquare\sigma$ et $\mathcal{I}, \sigma\theta \models_1 \blacksquare$. De même $\mathcal{I}, \theta \not\models_1 \square\sigma$ et $\mathcal{I}, \sigma\theta \not\models_1 \square$.

— $\mathcal{I}, \theta \models_1 \neg\varphi\sigma$ ssi $\mathcal{I}, \theta \not\models_1 \varphi\sigma$ ssi $\mathcal{I}, \sigma\theta \not\models_1 \varphi$ (par h.i.) ssi $\mathcal{I}, \sigma\theta \models_1 \neg\varphi$.

— $\mathcal{I}, \theta \models_1 (\varphi \vee \psi)\sigma$ ssi $\mathcal{I}, \theta \models_1 \varphi\sigma$ ou $\mathcal{I}, \theta \models_1 \psi\sigma$ ssi $\mathcal{I}, \sigma\theta \models_1 \varphi$ ou $\mathcal{I}, \sigma\theta \models_1 \psi$ (par h.i.) ssi $\mathcal{I}, \sigma\theta \models_1 \varphi \vee \psi$.

— On procède de même pour $\wedge$, $\Rightarrow$ et $\Leftrightarrow$.

— On a les équivalences suivantes :

$$\begin{aligned}
\mathcal{I}, \theta \models_1 (\exists x\, \varphi)\sigma \quad &\text{ssi} \quad \mathcal{I}, \theta \models_1 \exists x\, \varphi(\sigma[x \mapsto x]) \\
&\text{ssi} \quad \exists u \in \mathcal{I} \text{ tel que } \mathcal{I}, \theta[x \mapsto u] \models_1 \varphi(\sigma[x \mapsto x]) \\
&\text{ssi} \quad \exists u \in \mathcal{I} \text{ tel que } \mathcal{I}, \sigma[x \mapsto x]\theta[x \mapsto u] \models_1 \varphi \text{ (par h.i.)} \\
&\text{ssi} \quad \exists u \in \mathcal{I} \text{ tel que } \mathcal{I}, (\sigma\theta)[x \mapsto u] \models_1 \varphi \text{ (Lemme 3.36)} \\
&\text{ssi} \quad \mathcal{I}, \sigma\theta \models_1 \exists x\, \varphi.
\end{aligned}$$

— On procède de même pour $\forall x\, \varphi$. •

On remarque que si φ est une matrice, alors la condition sur σ est toujours vraie puisqu'aucune variable n'est liée dans φ ; on peut donc appliquer ce résultat à toutes les substitutions.

Ce théorème permet d'établir un lien entre l'instanciation syntaxique (le fait de remplacer des variables par des termes) et l'évaluation des variables, ce qui sera très utile pour se débarrasser des variables et ainsi se rapprocher de la logique propositionnelle.

3.4 Le théorème de Herbrand

Nous avons vu dans le Corollaire 3.13 que les matrices fermées sont des formules propositionnelles de $\mathrm{P}(\overline{\mathrm{A}}_\Sigma)$, puis dans le Théorème 3.31 que tout modèle du premier ordre $\mathcal{I}$ d'une matrice fermée peut être transformé en un modèle propositionnel $[\![.]\!]^{\mathcal{I}}$ de cette formule, en ne conservant que la valeur de vérité des atomes fermés. Afin de renforcer le parallèle entre $\overline{\mathrm{M}}_\Sigma$ et $\mathrm{P}(\overline{\mathrm{A}}_\Sigma)$ nous cherchons à faire l'inverse en associant à tout modèle propositionnel $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ d'une formule sans variable un modèle dans I_Σ.

Pour cela nous devons déterminer un domaine dans lequel évaluer les termes. Comme des termes fermés distincts peuvent être interprétés de façon différentes (on peut par exemple avoir $I(p(a)) \neq I(p(b))$), il faut nécessairement pouvoir attribuer des valeurs distinctes à des termes fermés distincts (voir Exemple 3.26). Une façon évidente de s'en assurer serait de faire en sorte que la valeur de chaque terme fermé t soit t lui-même. Cela implique de prendre l'ensemble $\overline{\mathrm{T}}_\Sigma$ comme domaine, mais ce n'est possible que s'il est non vide. D'après le Théorème 3.5 :

nous supposons dans la suite que $\Sigma_0^{\mathrm{F}} \neq \emptyset$.

Il existe évidemment de nombreuses interprétations dont le domaine est $\overline{\mathrm{T}}_\Sigma$. Parmi celles-ci, nous avons vu que l'algèbre des termes fermés $\overline{\mathcal{T}}_\Sigma$ (voir la Définition 3.19) vérifie la condition requise : $[\![t]\!]^{\overline{\mathcal{T}}_\Sigma} = t$ pour tout terme fermé t. Il n'est donc nécessaire que de fixer l'interprétation des symboles de Σ^{F} (ainsi que le domaine) pour obtenir cette propriété. Il nous reste donc à rajouter les interprétations des symboles de Σ^{P} afin d'obtenir un élément de la classe I_Σ, ce qui justifie la définition suivante.

Définition 3.38 Si $\Sigma \subseteq \Sigma'$ et $\mathcal{I} \in \mathrm{I}_\Sigma$, on appelle *extension de* $\mathcal{I}$ *à* Σ' tout $\mathcal{I}' \in \mathrm{I}_{\Sigma'}$ de même domaine que $\mathcal{I}$ et tel que $\forall s \in \Sigma, s^{\mathcal{I}} = s^{\mathcal{I}'}$; on dira alors que $\mathcal{I}$ est la *restriction de* $\mathcal{I}'$ *à* Σ, et on note $\mathcal{I} = \mathcal{I}'|_\Sigma$. △

Nous nous intéressons donc aux interprétations qui sont des extensions de l'algèbre des termes fermés.

Définition 3.39 (Modèle de Herbrand) On appelle *modèle de Herbrand* toute interprétation $\mathcal{H} \in \mathrm{I}_\Sigma$ telle que $\mathcal{H}|_{\Sigma^{\mathrm{F}}} = \overline{\mathcal{T}}_{\Sigma^{\mathrm{F}}}$. On note H_Σ l'ensemble des modèles de Herbrand.

Étant donnée une interprétation propositionnelle $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$, on associe à I le modèle de Herbrand $\mathcal{H}_I$ défini par :

— $\forall p \in \Sigma_0^{\mathrm{P}}$, $p^{\mathcal{H}_I} = I(p)$,
— $\forall n > 0, \forall p \in \Sigma_n^{\mathrm{P}}$, $p^{\mathcal{H}_I}$ est la fonction qui à $t_1, \ldots, t_n \in \overline{\mathrm{T}}_\Sigma$ associe $I(p(t_1, \ldots, t_n))$. △

On voit donc que, pour un Σ fixé, tous les modèles de Herbrand $\mathcal{H} \in \mathrm{H}_\Sigma$ ont le même domaine $\overline{\mathrm{T}}_\Sigma$ et que les symboles de constantes et de fonctions y ont la même interprétation (celle définie dans l'algèbre des termes fermés $\overline{\mathcal{T}}_\Sigma$). Les modèles de Herbrand ne peuvent donc différer que par l'interprétation des symboles de Σ^{P}.

EXEMPLE 3.40 Si $\Sigma^{\mathrm{F}} = \Sigma_0^{\mathrm{F}} = \{a, b\}$ et $\Sigma^{\mathrm{P}} = \Sigma_1^{\mathrm{P}} = \{p\}$, alors $\overline{\mathrm{T}}_\Sigma = \{a, b\}$ et $\overline{\mathrm{A}}_\Sigma = \{p(a), p(b)\}$. Soit $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ l'interprétation propositionnelle définie par $I(p(a)) = I(p(b)) = \mathrm{V}$, alors $\mathcal{H}_I$ est l'interprétation du premier ordre de domaine $\{a, b\}$ et telle que $a^{\mathcal{H}_I} = a$, $b^{\mathcal{H}_I} = b$ et $p^{\mathcal{H}_I}$ est la fonction qui à tout élément de $\{a, b\}$ associe V. On remarque que $I \models_0 p(a) \wedge p(b)$ et que $\mathcal{H}_I \models_1 p(a) \wedge p(b)$.

Soit $J \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ l'interprétation propositionnelle définie par $J(p(a)) = \mathrm{V}$ et $J(p(b)) = \mathrm{F}$, alors $a^{\mathcal{H}_J} = a$, $b^{\mathcal{H}_J} = b$ et $p^{\mathcal{H}_J}$ est la fonction qui à a associe V et à b associe F. On a $J \models_0 p(a) \wedge \neg p(b)$ et $\mathcal{H}_J \models_1 p(a) \wedge \neg p(b)$. ◊

Il est facile de voir que cette transformation des interprétations $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ en modèles de Herbrand $\mathcal{H}_I$ est l'inverse sur H_Σ de la transformation des interprétations $\mathcal{I} \in \mathrm{I}_\Sigma$ en interprétations propositionnelles $[\![.]\!]^{\mathcal{I}} \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ (voir la Définition 3.24) :

Lemme 3.41 $\forall I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$, $[\![.]\!]^{\mathcal{H}_I} = I$.

PREUVE. Soit α un atome fermé dans $\overline{\mathrm{A}}_\Sigma$. Si $\alpha \in \Sigma_0^{\mathrm{P}}$, alors d'après la Définition 3.23 on a $[\![\alpha]\!]^{\mathcal{H}_I} = \alpha^{\mathcal{H}_I} = I(\alpha)$. Sinon il existe $n > 0$, $p \in \Sigma_n^{\mathrm{P}}$ et $t_1, \ldots, t_n \in \overline{\mathrm{T}}_\Sigma$ tels que $\alpha = p(t_1, \ldots, t_n)$. Par définition d'un modèle de Herbrand, pour tout $i = 1, \ldots, n$, on a $[\![t_i]\!]^{\mathcal{H}_I} = t_i$, d'où

$$[\![\alpha]\!]^{\mathcal{H}_I} \;=\; p^{\mathcal{H}_I}([\![t_1]\!]^{\mathcal{H}_I}, \ldots, [\![t_n]\!]^{\mathcal{H}_I}) \;=\; p^{\mathcal{H}_I}(t_1, \ldots, t_n) \;=\; I(p(t_1, \ldots, t_n)) \;=\; I(\alpha).$$

•

EXERCICE 44. *La fonction de* $\mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ *dans* H_Σ *qui à* I *associe* $\mathcal{H}_I$ *est-elle injective ? Est-elle surjective ?*

Nous pouvons donc effectivement transformer un modèle propositionnel d'une matrice fermée en modèle du premier ordre de cette formule.

Corollaire 3.42 *Pour tout* $\varphi \in \overline{\mathrm{M}}_\Sigma$ *et tout* $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$, *on a* $\mathcal{H}_I \models_1 \varphi$ *si et seulement si* $I \models_0 \varphi$.

PREUVE. D'après le Théorème 3.31 on a $\mathcal{H}_I \models_1 \varphi$ si et seulement si $[\![.]\!]^{\mathcal{H}_I} \models_0 \varphi$, et d'après le Lemme 3.41, $I = [\![.]\!]^{\mathcal{H}_I}$. •

En combinant ces deux transformations nous obtenons alors le parallèle souhaité, qui est en fait une correspondance exacte entre logique propositionnelle et logique du premier ordre sans variables.

Théorème 3.43 *Pour toute formule $\varphi \in \overline{\mathrm{M}}_\Sigma$, φ est satisfaisable dans I_Σ si et seulement si elle l'est dans $\mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$.*

PREUVE. S'il existe une interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 \varphi$ alors d'après le Théorème 3.31 on a $[\![.]\!]^{\mathcal{I}} \models_0 \varphi$. S'il existe une interprétation $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ telle que $I \models_0 \varphi$ alors par le Corollaire 3.42 on a $\mathcal{H}_I \models_1 \varphi$. •

Ce théorème signifie que lorsqu'il s'agit de formules sans variables, la satisfaisabilité relativement à $\models_1$ est équivalente à la satisfaisabilité propositionnelle. Nous pourrons donc, pour ces formules, parler de satisfaisabilité sans préciser s'il s'agit de la relation $\models_0$, ou de $\models_1$. Une autre conséquence de ce théorème est que cette propriété est décidable : il existe un algorithme qui termine toujours et décide de la satisfaisabilité d'une formule du premier ordre sans variables.

On aimerait généraliser le Théorème 3.43 autant que possible en présence de variables. Nous allons voir que c'est possible à condition de n'utiliser que des quantifications universelles.

Définition 3.44 On appelle *formule universelle* toute formule de la forme $\forall x_1 \cdots \forall x_n \, \varphi$, où $\{x_1, \ldots, x_n\} = \mathrm{VL}(\varphi)$ et $\varphi \in \mathrm{M}_\Sigma$. On remarque qu'une formule universelle est toujours fermée. Soit $\varphi \in \mathrm{M}_\Sigma$, on note $\forall^\star \varphi$ toute formule de la forme $\forall x_1 \cdots \forall x_n \, \varphi$, où $\{x_1, \ldots, x_n\} = \mathrm{VL}(\varphi)$.

Soit $\sigma \in \overline{\mathrm{T}}_\Sigma^{\mathcal{V}}$ une substitution fermée, nous dirons de $\varphi\sigma$ qu'elle est une *instance fermée de* $\forall^\star \varphi$ (on a évidemment $\varphi\sigma \in \overline{\mathrm{M}}_\Sigma$). Si $U \subseteq \overline{\mathrm{F}}_\Sigma$ est un ensemble de formules universelles, on note $\mathrm{IF}(U) = \left\{ \varphi\sigma \mid \sigma \in \overline{\mathrm{T}}_\Sigma^{\mathcal{V}} \text{ et } \forall^\star \varphi \in U \right\}$. Si $\varphi \in \mathrm{M}_\Sigma$, on pose $\mathrm{IF}(\varphi) = \mathrm{IF}(\forall^\star \varphi) = \mathrm{IF}(\{\forall^\star \varphi\})$. △

Il est facile de voir que les instances fermées d'une formule universelle en sont des conséquences logiques.

Lemme 3.45 *Pour toute matrice $\varphi \in \mathrm{M}_\Sigma$ et toute substitution $\sigma \in \mathrm{T}_\Sigma^{\mathcal{V}}$, on a*

$$\forall^\star \varphi \models_1 \forall^\star \varphi\sigma.$$

PREUVE. Soit $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 \forall^\star \varphi$, alors par définition $\mathcal{I}, \theta \models_1 \varphi$ pour toute valuation $\theta \in \mathcal{I}^{\mathcal{V}}$. Comme $\sigma\theta$ est également une valuation dans $\mathcal{I}$ (voir la Définition 3.34), on en déduit que pour tout $\theta \in \mathcal{I}^{\mathcal{V}}$, on a $\mathcal{I}, \sigma\theta \models_1 \varphi$. D'après le Théorème 3.37 on a $\mathcal{I}, \theta \models_1 \varphi\sigma$ et comme cette propriété est vraie quelle que soit la valuation θ, on conclut que $\mathcal{I} \models_1 \forall^\star \varphi\sigma$. •

EXEMPLE 3.46 D'après ce lemme on a $\forall x \forall y \, p(x, y) \models_1 \forall x \forall y \forall z \, p(f(x, y), f(z, a))$.◇

Ce résultat est valable en particulier pour toutes les substitutions fermées ; on en déduit donc facilement le corollaire suivant.

Corollaire 3.47 $\forall^\star\varphi \models_1 \mathrm{IF}(\varphi)$.

PREUVE. Pour tout $\sigma \in \overline{\mathrm{T}}_\Sigma^\mathcal{V}$, $\varphi\sigma$ est fermée donc $\forall^\star\varphi\sigma = \varphi\sigma$ et $\forall^\star\varphi \models_1 \varphi\sigma$. On a bien $\forall^\star\varphi \models_1 \mathrm{IF}(\varphi)$. •

EXEMPLE 3.48 On a donc $\forall x\forall y\, p(x,y) \models_1 \{p(a,a), p(f(a,a),a), p(a,f(a,a)), \ldots\}$.◊

On voit tout aussi facilement que les quantifications universelles dans un modèle de Herbrand (qui est donc construit sur l'algèbre des termes fermés) correspondent exactement aux instances fermées.

Lemme 3.49 *Pour toute matrice $\varphi \in \mathrm{M}_\Sigma$ et tout modèle de Herbrand $\mathcal{H} \in \mathrm{H}_\Sigma$, on a $\mathcal{H} \models_1 \forall^\star\varphi$ si et seulement si $\mathcal{H} \models_1 \mathrm{IF}(\varphi)$.*

PREUVE. Soit $\tau \in \overline{\mathrm{T}}_\Sigma^\mathcal{V}$ une substitution fermée quelconque, alors pour toute substitution fermée σ, on a $\sigma\tau = \sigma$. D'après le Théorème 3.37, on a $\mathcal{H}, \sigma \models_1 \varphi$ ssi $\mathcal{H}, \tau \models_1 \varphi\sigma$, et puisque $\varphi\sigma$ ne contient pas de variables, ceci est équivalent à $\mathcal{H} \models_1 \varphi\sigma$. Donc :

$$\begin{array}{rll} \mathcal{H} \models_1 \forall^\star\varphi & \text{ssi} & \forall\sigma \in \overline{\mathrm{T}}_\Sigma^\mathcal{V},\ \mathcal{H}, \sigma \models_1 \varphi \\ & \text{ssi} & \forall\sigma \in \overline{\mathrm{T}}_\Sigma^\mathcal{V},\ \mathcal{H} \models_1 \varphi\sigma \\ & \text{ssi} & \mathcal{H} \models_1 \mathrm{IF}(\varphi). \end{array}$$

Ceci prouve le résultat. •

Comme les éléments de $\mathrm{IF}(\varphi)$ sont des matrices fermées on peut appliquer la transformation vers la logique propositionnelle vue précédemment.

Corollaire 3.50 *Pour toute interprétation $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$, on a $\mathcal{H}_I \models_1 \forall^\star\varphi$ si et seulement si $I \models_0 \mathrm{IF}(\varphi)$.*

PREUVE. D'après le Corollaire 3.42, on a $\mathcal{H}_I \models_1 \mathrm{IF}(\varphi)$ ssi $I \models_0 \mathrm{IF}(\varphi)$. •

Ce résultat généralise le Corollaire 3.42, car si $\varphi \in \overline{\mathrm{M}}_\Sigma$ alors $\varphi = \forall^\star\varphi$ et $\mathrm{IF}(\varphi) = \{\varphi\}$. Nous obtenons également la généralisation recherchée du Théorème 3.43 :

Corollaire 3.51 *$\forall^\star\varphi$ est satisfaisable dans I_Σ si et seulement si $\mathrm{IF}(\varphi)$ est satisfaisable dans $\mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$.*

PREUVE. S'il existe une interprétation propositionnelle $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ telle que $I \models_0 \mathrm{IF}(\varphi)$, alors $\mathcal{H}_I \models_1 \forall^\star\varphi$ d'après le Corollaire 3.50.

S'il existe une interprétation du premier ordre $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 \forall^\star\varphi$, alors comme $\forall^\star\varphi \models_1 \mathrm{IF}(\varphi)$ (Corollaire 3.47), on a $\mathcal{I} \models_1 \mathrm{IF}(\varphi)$. D'après le Théorème 3.31, on en déduit que $[\![.]\!]^\mathcal{I} \models_0 \mathrm{IF}(\varphi)$. •

On voit que la présence de quantificateurs universels se paye par le recours à un ensemble de matrices fermées qui peut être infini, ce qui fait bien sûr perdre la propriété de décidabilité. Cependant, le théorème de compacité permet de ramener l'insatisfaisabilité d'un ensemble infini à celle d'un ensemble fini (mais inconnu ; cela ne permet pas de retrouver la décidabilité). En reformulant le résultat précédent pour des ensembles de formules universelles, nous en déduisons le théorème de Herbrand.

Théorème 3.52 (de Herbrand) *Pour tout ensemble $U \subseteq \overline{\mathrm{F}}_\Sigma$ de formules universelles, U est insatisfaisable si et seulement si* $\mathrm{IF}(U)$ *admet un sous-ensemble fini insatisfaisable.*

PREUVE. U est insatisfaisable ssi $\mathrm{IF}(U)$ est insatisfaisable (Corollaire 3.51) ssi $\mathrm{IF}(U)$ admet un sous-ensemble fini insatisfaisable (Théorème 2.40). •

Nous pouvons déjà entrevoir l'importance de ce théorème en l'analysant du point de vue informatique.

EXERCICE 45. *En déduire une procédure de semi-décision pour l'insatisfaisabilité des formules universelles.*

Le théorème de Herbrand prouve donc que l'ensemble des formules universelles insatisfaisables est récursivement énumérable, ce qui suggère l'existence d'un système formel correct et complet pour les formules universelles. Si nous pouvions maintenant le généraliser à toutes les formules du premier ordre, nous aurions alors une procédure de semi-décision pour la logique du premier ordre, ainsi que bien d'autres conséquences. Il faudrait pour cela être capable de « représenter » une formule quelconque par une formule universelle.

Afin de préserver la semi-décidabilité de l'insatisfaisabilité, nous devons exiger deux propriétés de ce représentant universel : qu'il soit satisfaisable si et seulement si la formule d'origine l'est, et qu'il soit calculable à partir de la formule d'origine. Nous allons maintenant voir comment construire un tel représentant.

3.5 Formes normales et skolemisation

Nous allons essayer d'obtenir une formule universelle par des transformations successives d'une formule quelconque, chacune nous rapprochant un peu plus du résultat espéré. Intuitivement, nous devrons remonter les quantificateurs universels et nous débarrasser des quantificateurs existentiels. Dans certains cas on peut effectivement remonter les quantificateurs universels.

EXERCICE 46. *Montrer que* $p \Rightarrow \forall x\, q(x) \;\equiv_1\; \forall x\, (p \Rightarrow q(x))$.

Mais une première difficulté surgit.

EXERCICE 47. *Montrer que* $(\exists x\, p(x)) \Rightarrow q \ \equiv_1 \ \forall x\, (p(x) \Rightarrow q)$.

On voit donc que pour obtenir un quantificateur universel en préfixe, il faut parfois remonter un quantificateur existentiel. Cela est dû au fait que l'implication cache une négation implicite : $p \Rightarrow q \equiv_0 \neg p \vee q$, et c'est la négation qui inverse les quantificateurs.

EXERCICE 48. *Montrer que* $\neg\exists\, xp(x) \equiv_1 \ \forall x\, \neg p(x)$ *et* $\neg\forall x\, p(x) \equiv_1 \ \exists x\, \neg p(x)$.

Si ces deux équivalences permettent de remonter des quantificateurs, on voit également qu'elles permettent de descendre des négations, ce que l'on sait également faire pour les connecteurs $\vee$ et $\wedge$ (lois de De Morgan). On aimerait donc utiliser ces équivalences dans des règles de transformation :

$$\neg\exists x\, \varphi \to \forall x\, \neg\varphi \ \text{ et } \ \neg\forall x\, \varphi \to \exists x\, \neg\varphi$$

applicables aux sous-formules. Mais il se peut que ces sous-formules contiennent des variables libres, et pour justifier l'application de ces règles de transformation, nous devons d'abord étendre l'équivalence logique :

Définition 3.53 Soient $\varphi, \psi \in \mathrm{F}_\Sigma$ des formules non nécessairement fermées, on note $\varphi \models_1 \psi$ lorsque $\forall \mathcal{I} \in \mathrm{I}_\Sigma$, $\forall \theta \in \mathcal{I}^{\mathcal{V}}$, si $\mathcal{I}, \theta \models_1 \varphi$ alors $\mathcal{I}, \theta \models_1 \psi$. On note $\varphi \equiv_1 \psi$ lorsque $\varphi \models_1 \psi$ et $\psi \models_1 \varphi$. △

On remarque que ces relations sont identiques à $\models_1$ et $\equiv_1$ sur $\overline{\mathrm{F}}_\Sigma \times \overline{\mathrm{F}}_\Sigma$, (voir la Définition 3.27), ce qui justifie que nous utilisions la même notation.

Théorème 3.54 *Soient* $\varphi, \varphi', \psi, \psi' \in \mathrm{F}_\Sigma$ *telles que* φ *est une sous-formule de* ψ *et* $\psi' = \psi[\varphi' \to \varphi]$. *Si* $\varphi \equiv_1 \varphi'$ *alors* $\psi \equiv_1 \psi'$.

PREUVE. Par induction sur ψ. Il est évident que si $\psi \equiv_1 \psi'$ alors

$$\begin{array}{lll} \psi \wedge \alpha \equiv_1 \psi' \wedge \alpha & \psi \Rightarrow \alpha \equiv_1 \psi' \Rightarrow \alpha & \forall x\, \psi \equiv_1 \forall x\, \psi' \\ \psi \vee \alpha \equiv_1 \psi' \vee \alpha & \alpha \Rightarrow \psi \equiv_1 \alpha \Rightarrow \psi' & \exists x\, \psi \equiv_1 \exists x\, \psi', \\ \neg\psi \equiv_1 \neg\psi' & \psi \Leftrightarrow \alpha \equiv_1 \psi' \Leftrightarrow \alpha & \end{array}$$

ce qui conclut l'induction. •

Ce théorème est l'équivalent pour le premier ordre du Théorème 2.2.

EXERCICE 49. *Montrer que pour toute formule* φ *on a*

$$\neg\exists x\, \varphi \equiv_1 \ \forall x\, \neg\varphi \text{ et } \neg\forall x\, \varphi \equiv_1 \ \exists x\, \neg\varphi.$$

On peut donc appliquer systématiquement les règles suivantes aux sous-formules d'une formule donnée :

$$\begin{array}{rcl|rcl|rcl} \neg(\varphi \wedge \psi) & \to & \neg\varphi \vee \neg\psi & \neg\forall x\,\varphi & \to & \exists x\,\neg\varphi & \varphi \Rightarrow \psi & \to & (\neg\varphi \vee \psi) \\ \neg(\varphi \vee \psi) & \to & \neg\varphi \wedge \neg\psi & \neg\exists x\,\varphi & \to & \forall x\,\neg\varphi & \varphi \Leftrightarrow \psi & \to & (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi) \\ \neg\neg\varphi & \to & \varphi & \neg\blacksquare & \to & \square & \neg\square & \to & \blacksquare \end{array}$$

Le lecteur vérifiera facilement que ces règles préservent l'équivalence logique des sous-formules, et donc que la formule obtenue est logiquement équivalente à la formule d'origine (d'après le Théorème 3.54).

On voit que lorsqu'aucune règle ne peut plus être appliquée (ce qui arrive après un nombre fini d'application de ces règles), la formule obtenue ne présente plus que les connecteurs $\forall, \exists, \vee, \wedge$ et $\neg$, mais le connecteur $\neg$ ne peut précéder un autre connecteur et donc seuls les atomes peuvent être niés. Une telle formule est dite *sous forme normale négative*[10]. Plus généralement, lorsqu'on définit une transformation de formules au moyen d'un ensemble de règles, toute formule qui ne peut être transformée est appelée *forme normale* pour cet ensemble de règles.

Exercice 50. *Considérons la formule Ψ définie de la façon suivante :*

$$\psi \;=\; \forall x \exists y\, P(x, g(y)) \;\Rightarrow\; \exists z\, Q(g(z)).$$

Mettre ψ sous forme normale négative. Cette forme normale est-elle unique ?

La forme normale négative permet de généraliser le Théorème 3.54.

Lemme 3.55 *Soient ψ une forme normale négative, φ une sous-formule de ψ qui n'y est pas précédée d'une négation, $\mathcal{I}$ une interprétation et φ' une formule telle que, pour toute valuation θ, si $\mathcal{I}, \theta \models_1 \varphi$ alors $\mathcal{I}, \theta \models_1 \varphi'$. On a alors, pour toute valuation θ, si $\mathcal{I}, \theta \models_1 \psi$ alors $\mathcal{I}, \theta \models_1 \psi[\varphi' \to \varphi]$.*

Preuve. Par induction sur ψ, sachant que les connecteurs au dessus de φ, s'il y en a, ne peuvent être que $\wedge, \vee, \forall$ et $\exists$. S'il n'y en a pas c'est que $\psi = \varphi$ et donc $\psi[\varphi' \to \varphi] = \varphi'$; la propriété est donc vraie pour $\psi = \varphi$.

Si $\psi = \phi \vee \alpha$ où φ est une sous-formule de ϕ ; l'hypothèse d'induction est donc que pour tout θ si $\mathcal{I}, \theta \models_1 \phi$ alors $\mathcal{I}, \theta \models_1 \phi[\varphi' \to \varphi]$. Pour toute valuation θ, si $\mathcal{I}, \theta \models_1 \psi$ alors $\mathcal{I}, \theta \models_1 \phi$ (donc $\mathcal{I}, \theta \models_1 \phi[\varphi' \to \varphi]$ par hypothèse d'induction) ou $\mathcal{I}, \theta \models_1 \alpha$. On a donc $\mathcal{I}, \theta \models_1 \phi[\varphi' \to \varphi] \vee \alpha$, or $\psi[\varphi' \to \varphi] = \phi[\varphi' \to \varphi] \vee \alpha$.

Les cas où ψ est $\alpha \vee \phi$, $\phi \wedge \alpha$ ou $\alpha \wedge \phi$ sont similaires.

Si $\psi = \exists x\, \phi$, alors pour tout θ tel que $\mathcal{I}\theta \models_1 \psi$ il existe un $v \in \mathcal{I}$ tel que $\mathcal{I}, \theta[x \mapsto v] \models_1 \phi$, donc par hypothèse d'induction $\mathcal{I}, \theta[x \mapsto v] \models_1 \phi[\varphi' \to \varphi]$, et donc $\mathcal{I}, \theta \models_1 \exists x\, \phi[\varphi' \to \varphi]$, or $\psi[\varphi' \to \varphi] = \exists x\, \phi[\varphi' \to \varphi]$.

Le cas où ψ est $\forall x\, \phi$ est similaire, ce qui complète l'induction : la propriété est vraie pour toute formule ψ vérifiant les hypothèses. •

10. « positive » aurait été plus approprié, mais la tradition prévaut parfois sur la raison.

Corollaire 3.56 *Si $\varphi \models_1 \varphi'$ alors $\psi \models_1 \psi[\varphi' \to \varphi]$.*

PREUVE. Si $\varphi \models_1 \varphi'$ alors la condition du Lemme 3.55 sur φ et φ' est vraie quelque soit $\mathcal{I}$, donc sa conclusion également, ce qui prouve que $\psi \models_1 \psi[\varphi' \to \varphi]$. •

L'intérêt de la forme normale négative est aussi qu'elle permet de remonter les quantificateurs sans qu'ils ne puissent plus traverser une négation et s'inverser. Cependant ceci signifie que si nous remontons un quantificateur existentiel ($\exists$), il demeurera toujours un quantificateur existentiel dans le préfixe. Comment alors s'en débarrasser ?

EXERCICE 51. *Soient deux signatures $\Sigma \subset \Sigma'$, et soient $p \in \Sigma_1^{\mathrm{P}}$ et $a \in \Sigma_0'^{\mathrm{F}} \setminus \Sigma_0^{\mathrm{F}}$. Montrer que pour toute interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 \exists x\, p(x)$ il existe une extension $\mathcal{J}$ de $\mathcal{I}$ à Σ' telle que $\mathcal{J} \models_1 p(a)$.*

On voit dans l'Exercice 51 que la satisfaisabilité de $\exists x\, p(x)$ est équivalente à celle de $p(a)$ (puisqu'on a toujours $p(a) \models_1 \exists x\, p(x)$), à condition que la constante a ne soit pas déjà « utilisée » et qu'il soit donc possible de lui attribuer librement la valeur d'un x tel que $p(x)$. Peut-on alors remplacer toute variable quantifiée existentiellement par une constante encore inutilisée ?

Considérons la formule $\forall x \exists y\, (p(x,y) \wedge \neg p(y,x))$; elle est évidemment satisfaisable, par exemple en interprétant p par l'ordre strict sur $\mathbb{N}$. Mais si on remplace y par a, la formule obtenue $\forall x\, (p(x,a) \wedge \neg p(a,x))$ est insatisfaisable (elle ne peut être vraie pour $x = a$). Le problème est que la valeur de y dépend de celle de x, alors que la valeur attribuée à a, bien que libre, ne peut dépendre de x. Énoncer le problème, c'est trouver la solution : y doit être remplacée par un terme qui dépend de x, ce sera $f(x)$ où f est un symbole de fonction inutilisé. De fait, la formule $\forall x\, (p(x, f(x)) \wedge \neg p(f(x), x))$ est satisfaite par le même modèle que ci-dessus étendu à f, le symbole f étant interprété par la fonction successeur.

Mais est-il toujours possible de remplacer une variable liée y par un terme $f(x)$ contenant une variable ? Si dans la formule précédente on remplace la formule atomique $p(x,y)$ par $q(x) \vee \forall x\, r(x,y)$ (qui a les mêmes variables libres), alors on constate que la substitution de y par $f(x)$ engendre la capture de x par ce nouveau quantificateur (voir l'Exemple 3.33).

Afin d'empêcher ces captures, la solution la plus simple est de commencer par renommer les variables liées de sorte que chaque variable n'est quantifiée qu'une fois dans une formule fermée. Nous avons déjà vu (après l'Exercice 41) que ce renommage de variables préserve l'équivalence logique.

Définition 3.57 Une formule $\varphi \in \mathrm{F}_\Sigma$ est dite *normalisée* si pour toute variable $x \in \mathcal{V}$ qui a une occurrence liée dans φ,

1. il n'existe qu'une sous-formule de φ de préfixe $\forall x$ ou $\exists x$,
2. $x \notin \mathrm{VL}(\varphi)$. △

Théorème 3.58 *Pour toute formule $\varphi \in \mathrm{F}_\Sigma$ il existe une formule normalisée $\varphi' \in \mathrm{F}_\Sigma$ telle que $\varphi \equiv_1 \varphi'$. Si φ est sous forme normale négative alors φ' l'est également.*

EXERCICE 52. *Prouver le Théorème 3.58.*

Il est facile de voir que toute sous-formule d'une formule normalisée est également normalisée. Nous pouvons maintenant définir la transformation qui permet d'éliminer les quantificateurs existentiels.

Définition 3.59 Soit $\psi \in \overline{\mathrm{F}}_\Sigma$ une forme normale négative normalisée admettant une sous-formule $\exists x\, \varphi$ telle que $\mathrm{VL}(\exists x\, \varphi) = \{x_1, \ldots, x_n\}$. On note σ_φ toute substitution de la forme $\{x \mapsto f(x_1, \ldots, x_n)\}$ où $f \notin \Sigma$. Comme $\mathrm{Var}(\sigma) = \{x_1, \ldots, x_n\}$ et que φ est normalisée aucune de ces variables n'apparaît liée dans φ, on peut donc appliquer σ_φ à φ (selon la Définition 3.32).

On appelle *skolemisée de ψ en φ* toute formule ψ' obtenue de ψ en y remplaçant la sous-formule $\exists x\, \varphi$ par $\varphi\sigma_\varphi$. On note $\Sigma + f$ toute signature Σ' telle que $\Sigma \subset \Sigma'$ et $f \in \Sigma'^{\mathrm{F}}_n$, de sorte que la skolemisée ψ' de ψ en φ est dans $\overline{\mathrm{F}}_{\Sigma+f}$.

On appelle *skolemisée de ψ* toute formule ψ' sans quantificateur existentiel telle qu'il existe deux séquences finies $\psi_0, \ldots, \psi_k$ et $\varphi_0, \ldots, \varphi_{k-1}$ telles que $\psi_0 = \psi$, $\psi_k = \psi'$, et ψ_{i+1} est la skolemisée de ψ_i en φ_i pour tout $0 \leq i < k$. △

Grâce à la normalisation les variables $x_1, \ldots, x_n$ ne peuvent pas être liées dans φ et on a donc $\mathrm{VL}(\varphi\sigma_\varphi) = \{x_1, \ldots, x_n\}$. De plus, il est évident que toute skolemisée ψ' de ψ en φ est sous forme normale négative normalisée, ce qui permet d'enchaîner les skolemisations ; il existe donc toujours une skolemisée de ψ.

EXEMPLE 3.60 Considérons la formule suivante :

$$\psi_0 \;=\; \exists x \forall y \forall u \exists z\, p(f(u, x)) \vee \neg q(y, g(x, f(z, a))).$$

Soit $\varphi_0 = \forall y \forall u \exists z\, p(f(u, x)) \vee \neg q(y, g(b, f(z, a)))$, alors la skolemisée de ψ_0 en φ_0 est

$$\psi_1 \;=\; \forall y \forall u \exists z\, p(f(u, c)) \vee \neg q(y, g(c, f(z, a))),$$

où c est une constante qui n'apparaît pas dans Σ. Posons $\varphi_1 = p(f(u, c)) \vee \neg q(y, g(c, f(z, a)))$, alors la skolemisée de ψ_1 en φ_1 est

$$\psi_2 \;=\; \forall y \forall u\, p(f(u, c)) \vee \neg q(y, g(c, f(h(y, u), a))),$$

où h est un symbole de fonction d'arité 2 qui n'apparaît pas dans Σ, et ψ_2 est une skolemisée de ψ_0. ♢

Les φ_i correspondent aux sous-formules quantifiées existentiellement dans ψ ; il est donc évident que l'entier k dans la Définition 3.59 correspond au nombre de quantificateurs existentiels dans ψ.

Il est également clair que le résultat de la skolemisation n'est pas unique : il dépend non seulement des symboles utilisés à chaque étape mais également de l'ordre dans lequel on élimine les quantificateurs. Par exemple, on peut procéder comme suit :

$$\exists x\, \exists y\, p(x,y) \to \exists y\, p(a,y) \to p(a,b),$$

mais également de la façon suivante :

$$\exists x\, \exists y\, p(x,y) \to \exists x\, p(x,f(x)) \to p(a,f(a)).$$

Les formules $p(a,b)$ et $p(a,f(a))$ sont toutes deux des skolemisées de $\exists x\, \exists y\, p(x,y)$.

Nous allons maintenant montrer que cette transformation (qui est évidemment calculable) préserve la satisfaisabilité. En fait, nous allons obtenir une propriété légèrement plus forte en établissant un rapport entre les modèles d'une skolemisée et ceux de la formule d'origine. Nous commençons par la relation la plus simple.

Lemme 3.61 *Soient une formule normalisée $\varphi \in \mathrm{F}_\Sigma$ et $x, x_1, \ldots, x_n \in \mathcal{V}$ telles que $\mathrm{VL}(\exists x\, \varphi) = \{x_1, \ldots, x_n\}$, on a $\varphi\sigma_\varphi \models_1 \exists x\, \varphi$.*

Preuve. Pour toutes interprétation $\mathcal{I}$ et valuation $\theta \in \mathcal{I}^\mathcal{V}$ telles que $\mathcal{I}, \theta \models_1 \varphi\sigma_\varphi$, d'après le Théorème 3.37 on a donc $\mathcal{I}, \sigma_\varphi\theta \models_1 \varphi$. Pour toute variable $y \neq x$ on a $\sigma_\varphi\theta(y) = [\![\sigma_\varphi(y)]\!]^\mathcal{I}_\theta = [\![y]\!]^\mathcal{I}_\theta = \theta(y)$, donc $\sigma_\varphi\theta = \theta[x \mapsto v]$ où $v = \sigma_\varphi\theta(x)$. On a donc $\mathcal{I}, \theta[x \mapsto v] \models_1 \varphi$, et donc $\mathcal{I}, \theta \models_1 \exists x\, \varphi$ ce qui prouve que $\varphi\sigma_\varphi \models_1 \exists x\, \varphi$. •

Dans la direction inverse il est nécessaire d'étendre les modèles en donnant une valeur idoine au nouveau symbole de fonction f.

Lemme 3.62 *Soient une formule normalisée $\varphi \in \mathrm{F}_\Sigma$ et $x, x_1, \ldots, x_n \in \mathcal{V}$ telles que $\mathrm{VL}(\exists x\, \varphi) = \{x_1, \ldots, x_n\}$. Pour tout $\mathcal{I} \in \mathrm{I}_\Sigma$, il existe une extension $\mathcal{J}$ de $\mathcal{I}$ à $\Sigma + f$ telle que pour tout $\theta \in \mathcal{I}^\mathcal{V}$, si $\mathcal{J}, \theta \models_1 \exists x\, \varphi$ alors $\mathcal{J}, \theta \models_1 \varphi\sigma_\varphi$.*

Preuve. On considère une extension $\mathcal{J}$ de $\mathcal{I}$ qui interprète le symbole f de la façon suivante : pour tout $v_1, \ldots, v_n \in \mathcal{I}$ et pour toute valuation θ dans $\mathcal{I}$ telle que $\theta(x_1) = v_1, \ldots, \theta(x_n) = v_n$, s'il existe une valeur $v \in \mathcal{I}$ telle que

$$\mathcal{I}, \theta[x \mapsto v] \models_1 \varphi \tag{3.1}$$

alors $f^\mathcal{J}(v_1, \ldots, v_n) = v$. On remarque qu'il peut exister plusieurs valeurs possibles pour v ; on en choisit arbitrairement une parmi celles-ci. S'il n'y a aucune valeur qui vérifie la propriété (3.1) alors on choisit une valeur quelconque pour $f^\mathcal{J}(v_1, \ldots, v_n)$.

Soit $\theta \in \mathcal{I}^\mathcal{V}$, supposons que $\mathcal{J}, \theta \models_1 \exists x\, \varphi$. Alors $\mathcal{I}, \theta \models_1 \exists x\, \varphi$ puisque cette formule est construite uniquement sur la signature Σ (par hypothèse $\varphi \in \mathrm{F}_\Sigma$). Il existe donc une valeur $v \in \mathcal{I}$ telle que $\mathcal{I}, \theta[x \mapsto v] \models_1 \varphi$. Donc la propriété (3.1) est vraie et on a $\mathcal{I}, \theta' \models_1 \varphi$, avec $\theta' = \theta[x \mapsto f^\mathcal{J}(\theta(x_1), \ldots, \theta(x_n))]$, par construction de $f^\mathcal{J}$. Et puisque $\mathcal{J}$ est une extension de $\mathcal{I}$, on a également $\mathcal{J}, \theta' \models_1 \varphi$. Mais

$$\sigma_\varphi\theta(x) = [\![\sigma_\varphi(x)]\!]^\mathcal{J}_\theta = [\![f(x_1, \ldots, x_n)]\!]^\mathcal{J}_\theta = f^\mathcal{J}(\theta(x_1), \ldots, \theta(x_n)) = \theta'(x),$$

et comme ci-dessus pour toute variable $y \neq x$ on a $\sigma_\varphi\theta(y) = [\![y]\!]^{\mathcal{J}}_\theta = \theta(y) = \theta'(y)$, donc $\theta' = \sigma_\varphi\theta$. On a donc $\mathcal{J}, \sigma_\varphi\theta \models_1 \varphi$ et d'après le Théorème 3.37 on obtient $\mathcal{J}, \theta \models_1 \varphi\sigma_\varphi$. •

EXEMPLE 3.63 Soit $\varphi = p(x, x_1)$ et $\mathcal{I}$ l'interprétation où $p^{\mathcal{I}}$ est l'ordre strict $<$ sur $\mathbb{N}$. Pour toute valuation θ dans $\mathbb{N}$ si $\theta(x_1) > 0$ alors il existe $v \in \mathbb{N}$ tel que (3.1) est vraie ; on peut prendre par exemple $v = 0$. Si par contre $\theta(x_1) = 0$ alors il n'existe pas de $v \in \mathbb{N}$ qui vérifie (3.1) ; on peut donc prendre également $v = 0$. On étend donc $\mathcal{I}$ à $\mathcal{J}$ en choisissant pour $f^{\mathcal{J}}$ la fonction constante nulle. On vérifie alors que pour toute valuation θ, si $\mathcal{J}, \theta \models_1 \exists x\, p(x, x_1)$ alors $\theta(x_1) > 0$ donc $[\![f(x_1)]\!]^{\mathcal{J}}_\theta = 0 < [\![x_1]\!]^{\mathcal{J}}_\theta$ et donc $\mathcal{J}, \theta \models_1 p(f(x_1), x_1)$.

On remarque que si $\mathcal{J}, \theta \not\models_1 \exists x\, p(x, x_1)$ (donc si $\theta(x_1) = 0$) alors $\mathcal{J}, \theta \not\models_1 p(f(x_1), x_1)$; c'est simplement une conséquence du Lemme 3.62 qui dit que $p(f(x_1), x_1) \models_1 \exists x\, p(x, x_1)$. ◊

Ces propriétés concernent les sous-formules modifiées par la skolemisation. Grâce à la forme normale négative nous allons pouvoir les remonter à la formule fermée ψ. Nous introduisons auparavant une notation concise.

Définition 3.64 Soient des ensembles de formules $E \subseteq \overline{\mathrm{F}}_\Sigma$ et $E' \subseteq \overline{\mathrm{F}}_{\Sigma'}$ où $\Sigma \subseteq \Sigma'$. On dit que *les modèles de E sont extensibles à E',* et on note $E \models_1^\Sigma E'$ si pour toute interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 E$, il existe une extension $\mathcal{I}' \in \mathrm{I}_{\Sigma'}$ de $\mathcal{I}$ à Σ' telle que $\mathcal{I}' \models_1 E'$. Étant données des formules $\psi \in \overline{\mathrm{F}}_\Sigma$ et $\psi' \in \overline{\mathrm{F}}_{\Sigma'}$, on notera $\psi \models_1^\Sigma \psi'$ à la place de $\{\psi\} \models_1^\Sigma \{\psi'\}$. △

Théorème 3.65 *Pour toute forme normale négative normalisée $\psi \in \overline{\mathrm{F}}_\Sigma$ admettant une sous-formule $\exists x\, \varphi$, si $\psi' \in \overline{\mathrm{F}}_{\Sigma'}$ est une skolemisée de ψ en φ alors $\psi' \models_1 \psi$ et $\psi \models_1^\Sigma \psi'$.*

PREUVE. D'après le Lemme 3.61 on a $\varphi\sigma_\varphi \models_1 \exists x\varphi$ et on peut écrire $\psi = \psi'[\exists x\, \varphi \to \varphi\sigma_\varphi]$, donc d'après le Corollaire 3.56 on a $\psi' \models_1 \psi$.

Soit $\mathcal{I} \models_1 \psi$, d'après le Lemme 3.62 il existe une extension $\mathcal{J}$ de $\mathcal{I}$ à $\Sigma + f$ telle que pour tout $\theta \in \mathcal{J}^{\mathcal{V}}$, si $\mathcal{J}, \theta \models_1 \exists x\, \varphi$ alors $\mathcal{J}, \theta \models_1 \varphi\sigma_\varphi$. On peut aussi écrire $\psi' = \psi[\varphi\sigma_\varphi \to \exists x\, \varphi]$, donc d'après le Lemme 3.55 on a pour tout $\theta \in \mathcal{J}^{\mathcal{V}}$, si $\mathcal{J}, \theta \models_1 \psi$ alors $\mathcal{J}, \theta \models_1 \psi'$. Mais ψ et donc ψ' sont fermées, donc si $\mathcal{J} \models_1 \psi$ alors $\mathcal{J} \models_1 \psi'$. Comme $\mathcal{J}$ est une extension de $\mathcal{I}$ et que $\mathcal{I} \models_1 \psi$ par hypothèse, on a bien $\mathcal{J} \models_1 \psi$, d'où $\mathcal{J} \models_1 \psi'$, ce qui prouve que $\psi \models_1^\Sigma \psi'$. •

EXEMPLE 3.66 D'après l'exemple précédent on peut en conclure que $\forall x_1 \exists x\, p(x, x_1) \models_1^\Sigma \forall x_1\, p(f(x_1), x_1)$, où $\Sigma = \Sigma_2^{\mathrm{P}} = \{p\}$. On a également $\forall x_1\, p(f(x_1), x_1) \models_1 \forall x_1 \exists x\, p(x, x_1)$. ◊

Afin d'étendre cette propriété aux skolemisations successives de sous-formules de ψ, nous établissons une forme de transitivité de $\models_1^\Sigma$.

Lemme 3.67 *Soient $\Sigma \subseteq \Sigma' \subseteq \Sigma''$, $\psi \in \overline{\mathrm{F}}_\Sigma$, $\psi' \in \overline{\mathrm{F}}_{\Sigma'}$ et $\psi'' \in \overline{\mathrm{F}}_{\Sigma''}$, si $\psi \models_1^\Sigma \psi'$ et $\psi' \models_1^{\Sigma'} \psi''$ alors $\psi \models_1^\Sigma \psi''$.*

PREUVE. Identique à la preuve du Lemme 2.21. •

Corollaire 3.68 *Pour toute forme normale négative $\psi \in \overline{\mathrm{F}}_\Sigma$, soit $\psi' \in \overline{\mathrm{F}}_{\Sigma'}$ une skolemisée de ψ, on a $\psi' \models_1 \psi$ et $\psi \models_1^\Sigma \psi'$.*

PREUVE. Nous avons une séquence $\psi = \psi_0, \ldots, \psi_k = \psi'$ avec pour tout $1 \leq i \leq k$, $\psi_i \in \overline{\mathrm{F}}_{\Sigma_i}$ est une forme normale négative et $\Sigma = \Sigma_0 \subseteq \cdots \subseteq \Sigma_k = \Sigma'$. On a $\psi_{i+1} \models_1 \psi_i$, donc par transitivité $\psi' \models_1 \psi$. On a également $\psi_i \models_1^{\Sigma_i} \psi_{i+1}$, donc $\psi_0 \models_1^{\Sigma_0} \psi_k$, i.e., $\psi \models_1^\Sigma \psi'$. •

EXERCICE 53. *Montrer que toute skolemisée de ψ est satisfaisable si et seulement si ψ est satisfaisable.*

Le processus de skolemisation décrit ci-dessus permet donc de se débarrasser des quantificateurs existentiels en préservant la satisfaisabilité. Il ne nous reste plus qu'à remonter les quantificateurs universels en appliquant les transformations suivantes aux sous-formules d'une formule skolemisée :

$$\begin{array}{llll} \varphi \wedge \forall x\, \psi & \to \forall x\, (\varphi \wedge \psi) & \varphi \vee \forall x\, \psi & \to \forall x\, (\varphi \vee \psi) \\ \forall x\, \psi \wedge \varphi & \to \forall x\, (\psi \wedge \varphi) & \forall x\, \psi \vee \varphi & \to \forall x\, (\psi \vee \varphi). \end{array}$$

Comme ces règles ne peuvent mener à des captures (car x n'apparaît jamais dans φ puisque la formule est normalisée), elles préservent l'équivalence logique des sous-formules. On obtient donc après un nombre fini d'applications de ces règles de mise sous *forme normale universelle* une formule universelle qui est logiquement équivalente (Théorème 3.54) à la formule skolemisée de départ. C'est le représentant universel que nous cherchions.

Définition 3.69 Pour toute formule $\psi \in \overline{\mathrm{F}}_\Sigma$, on note $\mathrm{u}_\Sigma(\psi)$ une formule universelle obtenue en appliquant à ψ les transformations successives décrites ci-dessus :

1. la mise sous forme normale négative,
2. la normalisation,
3. la skolemisation (en utilisant des symboles de constantes ou de fonctions n'appartenant pas à Σ),
4. et la mise sous forme universelle.

On notera alors Σ_ψ une signature telle que $\Sigma \subseteq \Sigma_\psi$ et $\mathrm{u}_\Sigma(\psi) \in \overline{\mathrm{F}}_{\Sigma_\psi}$. △

Théorème 3.70 *Pour toute formule $\psi \in \overline{\mathrm{F}}_\Sigma$, on a $\mathrm{u}_\Sigma(\psi) \models_1 \psi$ et $\psi \models_1^\Sigma \mathrm{u}_\Sigma(\psi)$.*

PREUVE. Soient ψ_1 la forme normale négative de ψ, ψ_2 une normalisée de ψ_1 et ψ_3 une skolemisée de ψ_2 telles que $\mathrm{u}_\Sigma(\psi)$ est une forme normale universelle de ψ_3. On a $\psi \equiv_1 \psi_1$, $\psi_1 \equiv_1 \psi_2$ et $\psi_3 \equiv_1 \mathrm{u}_\Sigma(\psi)$. D'après le Corollaire 3.68 on a $\psi_3 \models_1 \psi_2$, donc $\mathrm{u}_\Sigma(\psi) \models_1 \psi$. De même on a $\psi_2 \models_1^\Sigma \psi_3$, on obtient donc également $\psi \models_1^\Sigma \mathrm{u}_\Sigma(\psi)$. •

Il est donc clair que, comme ci-dessus, le représentant universel $\mathrm{u}_\Sigma(\psi)$ est satisfaisable si et seulement si ψ est satisfaisable.

3.6 Conséquences générales du théorème de Herbrand

Au travers des transformations décrites ci-dessus, le théorème de Herbrand s'applique indirectement à toutes les formules fermées, donc à la logique du premier ordre. Mais afin d'en tirer des conséquences générales, nous devons appliquer la transformation u_Σ à des ensembles de formules, ce qui requiert une certaine prudence afin d'assurer que les symboles de fonctions introduits par des skolemisations distinctes soient distincts.

Définition 3.71 Soit $E \subseteq \overline{\mathrm{F}}_\Sigma$ un ensemble de formules fermées, on définit l'ensemble $\mathrm{u}_\Sigma(E)$ de la façon suivante. Si E est infini (resp. fini) on énumère ses éléments : $E = \{\psi_i \mid i \in \mathbb{N}\}$ (resp. $E = \{\psi_i \mid 0 \leq i < n\}$ où $n = |E|$). Soit $\Sigma_0 = \Sigma$, et pour tout $i \geq 0$ (resp. $0 \leq i < n$) on pose $\varphi_i = \mathrm{u}_{\Sigma_i}(\psi_i)$ et $\Sigma_{i+1} = (\Sigma_i)_{\psi_i}$ (voir la Définition 3.69). On note $\mathrm{u}_\Sigma(E)$ l'ensemble $\{\varphi_i \mid i \in \mathbb{N}\}$ (resp. $\{\varphi_i \mid 0 \leq i < n\}$) et Σ_E la signature $\bigcup_{i\in\mathbb{N}} \Sigma_i$ (resp. Σ_n). △

Σ_E est bien une signature puisque la suite des Σ_i est croissante ($\Sigma_i \subseteq \Sigma_{i+1}$) et parce qu'une union au plus dénombrable d'ensembles au plus dénombrables est au plus dénombrable. Il est donc clair que $\mathrm{u}_\Sigma(E) \subseteq \overline{\mathrm{F}}_{\Sigma_E}$. Le Théorème 3.70 se généralise alors aux ensembles de formules.

Corollaire 3.72 *Pour tout $E \subseteq \overline{\mathrm{F}}_\Sigma$ on a $\mathrm{u}_\Sigma(E) \models_1 E$ et $E \models_1^\Sigma \mathrm{u}_\Sigma(E)$.*

PREUVE. On reprend les notations de la Définition 3.71. Pour tout i on a $\varphi_i \models_1 \psi_i$, donc $\mathrm{u}_\Sigma(E) \models_1 E$.

Soit $\mathcal{I} \in \mathrm{I}_\Sigma$ telle que $\mathcal{I} \models_1 E$; donc pour tout i on a $\mathcal{I} \models_1 \psi_i$. On construit par induction sur $i \in \mathbb{N}$ (ou $i < |E|$ si E est fini) une suite croissante d'interprétations $\mathcal{I}_i$ telles que

$$\mathcal{I}_i \in \mathrm{I}_{\Sigma_i},\ \mathcal{I}_i|_\Sigma = \mathcal{I} \text{ et } \mathcal{I}_i \models_1 \{\varphi_j \mid j < i\}\,. \tag{3.2}$$

La propriété (3.2) est trivialement vraie pour $i = 0$ en prenant $\mathcal{I}_0 = \mathcal{I}$. On la suppose maintenant vraie pour un i donné. Comme $\mathcal{I}_i|_\Sigma = \mathcal{I}$, on a $\mathcal{I}_i \models_1 \psi_i$ et puisque $\psi_i \models_1^{\Sigma_i} \varphi_i$ il existe une extension $\mathcal{I}_{i+1} \in \mathrm{I}_{\Sigma_{i+1}}$ de $\mathcal{I}_i$ telle que $\mathcal{I}_{i+1} \models_1 \varphi_i$. Comme $\mathcal{I}_{i+1}|_{\Sigma_i} = \mathcal{I}_i$, on en déduit que $\mathcal{I}_{i+1}|_\Sigma = \mathcal{I}_i|_\Sigma = \mathcal{I}$ et que $\mathcal{I}_{i+1} \models_1 \{\varphi_j \mid j < i\}$,

d'où $\mathcal{I}_{i+1} \models_1 \{\varphi_j \mid j < i+1\}$. La propriété (3.2) est donc vraie en $i+1$. On peut donc construire l'interprétation $\mathcal{J} = \bigcup_{i\in\mathbb{N}} \mathcal{I}_i$ (ou $\mathcal{J} = \mathcal{I}_{|E|}$ si E est fini), qui est élément de I_{Σ_E} et telle que $\mathcal{J}|_\Sigma = \mathcal{I}$ et $\mathcal{J} \models_1 \mathrm{u}_\Sigma(E)$, ce qui prouve le résultat. •

Nous pouvons maintenant facilement obtenir pour la logique du premier ordre le résultat déjà connu pour la logique propositionnelle (Théorème 2.40) :

Théorème 3.73 (de compacité) *Pour tout ensemble de formules fermées $E \subseteq \overline{\mathrm{F}}_\Sigma$, si E est insatisfaisable alors il existe un sous ensemble fini de E qui est insatisfaisable.*

Preuve. Ceci est évident lorsque E est fini ; on suppose donc E infini. Comme $\mathrm{u}_\Sigma(E) \models_1 E$, l'ensemble de formules universelles $\mathrm{u}_\Sigma(E)$ est insatisfaisable. D'après le Théorème 3.52, $\mathrm{IF}(\mathrm{u}_\Sigma(E))$ admet un sous-ensemble fini H qui est insatisfaisable. Il existe donc un sous-ensemble fini $F \subset E$ tel que $H \subseteq \mathrm{IF}(\mathrm{u}_\Sigma(F))$. Supposons que F admet un modèle $\mathcal{I} \in \mathrm{I}_\Sigma$. Comme $F \models_1^\Sigma \mathrm{u}_\Sigma(F)$ d'après le Corollaire 3.72, il existe une interprétation $\mathcal{J} \in \mathrm{I}_{\Sigma_F}$ telle que $\mathcal{J} \models_1 \mathrm{u}_\Sigma(F)$, d'où $\mathcal{J} \models_1 \mathrm{IF}(\mathrm{u}_\Sigma(F))$ d'après le Corollaire 3.47. Ceci implique que $\mathcal{J} \models_1 H$ ce qui est impossible. Donc F est insatisfaisable. •

Ce résultat implique qu'on a $E \models_1 \psi$ si et seulement s'il existe un sous-ensemble fini $H \subseteq E$ tel que $H \models_1 \psi$, i.e., qu'une conséquence logique n'utilise qu'un ensemble fini d'hypothèses, exactement comme une déduction. Si ce n'était pas le cas, la logique du premier ordre ne pourrait pas être complète. Nous avons donc là un indice de sa complétude.

Nous pouvons aussi obtenir facilement un résultat beaucoup plus surprenant.

Théorème 3.74 (de Löwenheim-Skolem) *Pour tout ensemble de formules $E \subseteq \overline{\mathrm{F}}_\Sigma$, si E est satisfaisable alors E admet un modèle au plus dénombrable.*

Preuve. Soit $\mathcal{I} \in \mathrm{I}_\Sigma$ un modèle de E, d'après le Corollaire 3.72 il existe un modèle $\mathcal{J} \in \mathrm{I}_{\Sigma_E}$ de $\mathrm{u}_\Sigma(E)$, donc $\mathcal{J} \models_1 \mathrm{IF}(\mathrm{u}_\Sigma(E))$. Mais $\mathrm{IF}(\mathrm{u}_\Sigma(E)) \subseteq \mathrm{F}_{\Sigma_E}$, donc d'après le Théorème 3.43, il existe un modèle propositionnel $I \in \mathrm{I_p}(\overline{\mathrm{A}}_{\Sigma_E})$ tel que $I \models_0 \mathrm{IF}(\mathrm{u}_\Sigma(E))$. D'après le Corollaire 3.50, on a $\mathcal{H}_I \models_1 \mathrm{u}_\Sigma(E)$, et comme $\mathrm{u}_\Sigma(E) \models_1 E$ on en déduit que $\mathcal{H}_I \models_1 E$. Or $\mathcal{H}_I$ est un modèle de Herbrand dont le domaine est $\overline{\mathrm{T}}_{\Sigma_E}$, un ensemble au plus dénombrable, d'où le résultat. •

On voit ainsi que tout modèle, même non dénombrable, d'un ensemble de formules peut être « représenté » par (ou « réduit » à) un modèle de Herbrand. Par exemple, si on peut montrer en logique du premier ordre qu'une fonction est continue sur $\mathbb{R}$, alors on peut interpréter ce résultat de continuité dans une algèbre de termes, ce qui peut paraître étrange.

Plus surprenant, on peut exprimer par un ensemble de formules [11] que deux ensembles sont infinis et qu'ils n'admettent pas de bijection entre eux. Cet ensemble est

11. Un tel ensemble serait trop complexe et surtout trop long pour être exposé ici.

évidemment satisfaisable dans un modèle où l'un des ensembles est dénombrable et l'autre non. Il en existe donc un modèle dénombrable, ce qui semble contradictoire ; c'est le *paradoxe de Skolem*. Mais ce n'est un paradoxe qu'en apparence : ce modèle dénombrable ne peut contenir toutes les fonctions possibles entre deux ensembles dénombrables (puisqu'il y en a une infinité indénombrable), le fait de ne pas y trouver de bijection ne signifie pas qu'il n'en existe pas.

Il est donc possible dans cette logique d'exprimer et de déduire des propriétés portant sur des modèles non dénombrables, mais nous ne pourrons déduire des propriétés spécifiques aux modèles non dénombrables, puisqu'elles seront toujours vraies dans un modèle au plus dénombrable. Ce théorème expose donc une limite intrinsèque à la logique du premier ordre.

Mais le but de la Section 3.5 était de généraliser une procédure de semi-décision au problème de l'insatisfaisabilité en logique du premier ordre. On peut effectivement généraliser le théorème de Herbrand aux formules quelconques grâce à la transformation u_Σ.

Théorème 3.75 *Tout ensemble $E \subseteq \overline{\mathrm{F}}_\Sigma$ est insatisfaisable si et seulement si $\mathrm{IF}(\mathrm{u}_\Sigma(E))$ admet un sous-ensemble fini insatisfaisable.*

PREUVE. D'après le Théorème 3.52 et le Corollaire 3.72. •

On peut donc énumérer une suite croissante $(F_n)_{n\in\mathbb{N}}$ de sous-ensembles finis de $\mathrm{IF}(\mathrm{u}_\Sigma(E))$, et pour chaque n tester si F_n est insatisfaisable (ce test est décidable) jusqu'à trouver une réponse positive. Si tout sous-ensemble fini de $\mathrm{IF}(\mathrm{u}_\Sigma(E))$ est inclus dans un F_n, condition facile à réaliser, alors cette procédure termine si E est insatisfaisable.

Corollaire 3.76 *Pour toute signature Σ, l'ensemble des formules valides de $\overline{\mathrm{F}}_\Sigma$ est récursivement énumérable.*

PREUVE. Nous venons de montrer que c'est le cas de l'ensemble des formules insatisfaisables de $\overline{\mathrm{F}}_\Sigma$, on conclut par la Proposition 1.16. •

Considérons par exemple l'ensemble $E_t = \{\neg p(t), \forall x\, p(x)\}$ où $t \in \overline{\mathrm{T}}_\Sigma$ et Σ contient au moins un symbole de fonction. Dans ce cas l'ensemble des termes fermés est infini dénombrable et on peut écrire $\overline{\mathrm{T}}_\Sigma = \{t_i \mid i \in \mathbb{N}\}$. On a donc $\mathrm{IF}(\mathrm{u}_\Sigma(E_t)) = \{\neg p(t), p(t_0), p(t_1), \ldots\}$ et on peut prendre $F_n = \{\neg p(t), p(t_0), p(t_1), \ldots, p(t_n)\}$. On obtient un ensemble insatisfaisable à partir de l'entier n tel que $t_n = t$, et pas avant.

Par conséquent, si on se base sur une énumération fixe de $\overline{\mathrm{T}}_\Sigma$ (indépendante de t), le temps requis pour réfuter E_t est arbitrairement grand (pour tout n on peut choisir $t = t_n$). Il est pourtant évident qu'il suffit de ne considérer que l'instance $x = t$ pour obtenir la contradiction souhaitée, quel que soit t.

Il en est de même si on part de $E = \{\forall y\, \neg p(f(y)), \forall x\, p(x)\}$. Il est inutile d'énumérer les instances fermées de x et y pour voir que, quel que soit le terme t qui

instancie y, on obtient une contradiction en instanciant x avec $f(t)$. Autrement dit, on voit que l'équation $x = f(y)$ admet une solution dans $\overline{\mathrm{T}}_\Sigma$ quelle que soit la valeur de y, et que cette solution permet d'obtenir directement la contradiction.

Mais prenons l'exemple $E = \{\forall x\, p(x, f(x)), \forall y\, \neg p(f(y), y)\}$. Pour obtenir une contradiction il faut résoudre dans $\overline{\mathrm{T}}_\Sigma$ le système

$$\begin{cases} x = f(y) \\ f(x) = y \end{cases}$$

donc en remplaçant x par $f(y)$ dans la deuxième équation il faut trouver y tel que $f(f(y)) = y$ ce qui est évidemment impossible : il n'existe aucun terme t dans $\overline{\mathrm{T}}_\Sigma$ (ni même dans T_Σ) vérifiant $f(f(t)) = t$. De fait, cet ensemble E est satisfaisable dans $\mathbb{N}$, avec l'ordre strict pour p et la fonction successeur pour f, comme nous l'avons déjà vu.

3.7 Unification

La discussion précédente montre que pour espérer construire une procédure de semi-décision efficace permettant de tester l'insatisfaisabilité d'un ensemble de formules, il semble essentiel de savoir résoudre des systèmes d'équations dans $\overline{\mathrm{T}}_\Sigma$, afin de déterminer rapidement les instances fermées qui sont importantes. Chaque équation est de la forme $t \simeq t'$, où $t, t' \in \mathrm{T}_\Sigma$. On utilise le symbole $\simeq$ plutôt que $=$ afin d'éviter toute confusion : les membres d'une équation ne sont pas égaux, mais ils peuvent éventuellement être rendus égaux par une « solution ». Dans $\overline{\mathrm{T}}_\Sigma$, c'est une substitution $\sigma \in \overline{\mathrm{T}}_\Sigma^{\mathcal{V}}$ telle que $t\sigma = t'\sigma$. Nous pouvons cependant faire deux remarques :

1. nous n'avons besoin de trouver une valeur que pour les variables qui apparaissent dans le système d'équations et qui sont donc en nombre fini,
2. si on trouvait une solution σ dans $\mathrm{T}_\Sigma^{\mathcal{V}}$ on serait sûr qu'il y en a une dans $\overline{\mathrm{T}}_\Sigma^{\mathcal{V}}$ (par exemple la substitution $\sigma\bar{a}$ où $\bar{a}$ est la substitution qui associe à toute variable la constante $a \in \Sigma_0^{\mathrm{F}}$). La réciproque étant évidemment vraie (puisque $\overline{\mathrm{T}}_\Sigma^{\mathcal{V}} \subseteq \mathrm{T}_\Sigma^{\mathcal{V}}$) il n'est donc ni plus difficile ni plus facile de résoudre ces systèmes d'équations dans T_Σ que dans $\overline{\mathrm{T}}_\Sigma$.

Intuitivement, il est préférable de choisir comme solution à l'équation $x \simeq f(y)$ la substitution $\{x \mapsto f(y)\}$ plutôt qu'une substitution fermée $\{x \mapsto f(t), y \mapsto t\}$, car celle-ci oblige à choisir arbitrairement un terme $t \in \overline{\mathrm{T}}_\Sigma$. Nous choisissons donc de résoudre les systèmes d'équations dans T_Σ, ce qui avec les remarques précédentes nous mène naturellement aux définitions suivantes.

Définition 3.77 Un *problème d'unification* P est un ensemble fini d'*équations*[12] $t \simeq t'$ où $t, t' \in \mathrm{T}_\Sigma$. On pose :

$$\mathrm{Var}(t \simeq t') = \mathrm{Var}(t) \cup \mathrm{Var}(t') \text{ et } \mathrm{Var}(P) = \bigcup_{t \simeq t' \in P} \mathrm{Var}(t \simeq t').$$

Une substitution $\sigma \in \mathrm{T}_\Sigma^{\mathcal{V}}$ est un *unificateur* de P si pour tout $t \simeq t'$ dans P on a $t\sigma = t'\sigma$. On note $\mathrm{Unif}(P)$ l'ensemble des unificateurs de P. Si $\mathrm{Unif}(P) \neq \emptyset$ on dit que P est *unifiable* et de même si $\mathrm{Unif}(\{t \simeq t'\}) \neq \emptyset$ on dit que t et t' sont *unifiables*. △

EXERCICE 54. *D'après cette définition l'ensemble vide est un problème d'unification. Calculer* $\mathrm{Unif}(\emptyset)$.

Nous définissons des règles de transformation des problèmes d'unification, dont nous nous servirons pour résoudre ces problèmes.

Définition 3.78 Nous considérons les règles conditionnelles suivantes, qu'on peut considérer comme des relations binaires sur l'ensemble des problèmes d'unification augmenté d'un élément $\times$:

$$\begin{array}{lll}
\textsc{dec}: & P \cup \{f(t_1,\ldots,t_n) \simeq f(t'_1,\ldots,t'_n)\} \to P \cup \{t_1 \simeq t'_1,\ldots,t_n \simeq t'_n\} & \\
\textsc{cfl}: & P \cup \{f(t_1,\ldots,t_n) \simeq g(t'_1,\ldots,t'_{n'})\} \to \times & \text{si } f \neq g \\
\textsc{eff}: & P \cup \{t \simeq t\} \to P & \\
\textsc{inv}: & P \cup \{t \simeq x\} \to P \cup \{x \simeq t\} & \text{si } x \in \mathcal{V} \text{ et } t \notin \mathcal{V} \\
\textsc{occ}: & P \cup \{x \simeq t\} \to \times & \text{si } x \in \mathrm{Var}(t) \text{ et } x \neq t \\
\textsc{rpl}: & P \cup \{x \simeq t\} \to P[t/x] \cup \{x \simeq t\} & \text{si } x \in \mathrm{Var}(P) \setminus \mathrm{Var}(t).
\end{array}$$

On note $\to_{\mathbf{U}}$ l'union de ces relations, et $\to_{\mathbf{U}}^\star$ sa clôture réflexive et transitive. △

Dans les règles DEC (décomposition) et CFL (conflit), les symboles f et g sont quelconques et peuvent être des constantes si $n = 0$ ou $n' = 0$. La règle OCC (test d'occurrence) vient simplement du fait qu'un terme ne peut être un sous-terme strict de lui-même. La règle EFF permet d'effacer les égalités triviales.

EXEMPLE 3.79 Pour résoudre le système d'équations suivant, on encadre l'équation utilisée à chaque étape.

$$\left\{\begin{array}{l}\boxed{f(x,y) \simeq f(a,x)}\\ z \simeq g(y)\end{array}\right. \to_{\textsc{dec}} \left\{\begin{array}{l}\boxed{x \simeq a}\\ y \simeq x\\ z \simeq g(y)\end{array}\right. \to_{\textsc{rpl}} \left\{\begin{array}{l}x \simeq a\\ \boxed{y \simeq a}\\ z \simeq g(y)\end{array}\right. \to_{\textsc{rpl}} \left\{\begin{array}{l}x \simeq a\\ y \simeq a\\ z \simeq g(a)\end{array}\right.$$

Aucune règle ne s'applique sur ce dernier système. ◊

12. Une équation est ordonnée : elle a un membre gauche et un membre droit.

Afin de justifier la règle RPL (remplacement), nous démontrons le lemme suivant.

Lemme 3.80 *Quels que soient $s, t \in \mathrm{T}_\Sigma$, $x \in \mathcal{V}$ et σ une substitution telle que $\sigma(x) = t\sigma$, on a $s\sigma = s[t/x]\sigma$.*

PREUVE. On procède par induction sur s :

— si $s \in \mathcal{V}$, soit $s \neq x$ et alors $s[t/x] = s$, soit $s = x$ et alors $s[t/x]\sigma = t\sigma = \sigma(x) = s\sigma$,

— si $s \in \Sigma_0^{\mathrm{F}}$ alors $s[t/x] = s = s\sigma$,

— si $s = f(s_1, \ldots, s_n)$ pour $n > 0$, $f \in \Sigma_n^{\mathrm{F}}$ et $s_1, \ldots, s_n \in \mathrm{T}_\Sigma$, alors

$$\begin{aligned} s\sigma &= f(s_1\sigma, \ldots, s_n\sigma) \\ &= f(s_1[t/x]\sigma, \ldots, s_n[t/x]\sigma) \text{ (par hypothèse d'induction)} \\ &= f(s_1[t/x], \ldots, s_n[t/x])\sigma \\ &= f(s_1, \ldots, s_n)[t/x]\sigma \\ &= s[t/x]\sigma. \end{aligned}$$

D'où le résultat. •

Une conséquence de ce lemme est que pour une substitution σ telle que $\sigma(x) = t\sigma$, on a $\sigma \in \mathrm{Unif}(s \simeq s')$ si et seulement si $\sigma \in \mathrm{Unif}(s[t/x] \simeq s'[t/x])$. On peut maintenant prouver que ces règles de transformation d'un système d'équations sont correctes dans la mesure où elles préservent exactement les solutions, sans en retrancher ni en ajouter.

Lemme 3.81 *Pour tous problèmes d'unification P et P', si $P \to_{\mathbf{U}}{}^{\star} P'$ alors $\mathrm{Unif}(P) = \mathrm{Unif}(P')$ et si $P \to_{\mathbf{U}}{}^{\star} \times$ alors $\mathrm{Unif}(P) = \emptyset$.*

PREUVE. Par induction sur la longueur de la transformation de P en $\times$ ou en P'. Le résultat est trivial si la longueur est 0 car alors $P = P'$. On suppose maintenant que le résultat est vrai pour toute transformation de longueur l, et on considère une transformation de longueur $l+1$ de sorte qu'il existe P'' tel que $P \to_{\mathbf{U}}{}^{l} P'' \to_{\mathbf{U}} P'$ (ou $P'' \to_{\mathbf{U}} \times$). Si la règle appliquée à P'' est OCC ou CFL (i.e. si $P'' \to_{\mathbf{U}} \times$), alors P'' n'a pas de solution. En effet, pour toute substitution σ on a $x\sigma \neq t\sigma$ si $x \in \mathrm{Var}(t)$ avec $t \neq x$, et on a $f(t_1, \ldots, t_n)\sigma \neq g(t'_1, \ldots, t'_{n'})\sigma$ si $f \neq g$. Donc $\mathrm{Unif}(P) = \mathrm{Unif}(P'') = \emptyset$ par hypothèse d'induction.

Sinon la règle transformant P'' en P' est EFF, INV, DEC ou RPL. Si cette règle est EFF, alors on a $\mathrm{Unif}(P'') = \mathrm{Unif}(P')$ puisque $t\sigma = t\sigma$ pour tout σ. Il en est de même pour INV car $t\sigma = x\sigma$ si et seulement si $x\sigma = t\sigma$, et pour DEC puisque $f(t_1, \ldots, t_n)\sigma = f(t'_1, \ldots, t'_n)\sigma$ si et seulement si $t_i\sigma = t'_i\sigma$ pour tout $1 \leq i \leq n$. Enfin, si la règle est RPL alors il existe Q, x et t tels que $P'' = Q \cup \{x = t\}$ et $P' = Q[t/x] \cup \{x = t\}$, et donc pour tout σ on a

$$\begin{aligned} \sigma \in \mathrm{Unif}(P'') \quad &\text{ssi} \quad \sigma(x) = t\sigma \text{ et } \sigma \in \mathrm{Unif}(Q) \\ &\text{ssi} \quad \sigma(x) = t\sigma \text{ et } \sigma \in \mathrm{Unif}(Q[t/x]) \text{ (d'après le Lemme 3.80)} \\ &\text{ssi} \quad \sigma \in \mathrm{Unif}(P'). \end{aligned}$$

On a donc $\mathrm{Unif}(P) = \mathrm{Unif}(P'') = \mathrm{Unif}(P')$ par hypothèse d'induction. •

Mais il reste à savoir si ces règles peuvent nous mener à une solution ou si elles peuvent tourner en rond en ne menant jamais à rien. En fait, nous allons montrer qu'elles mènent toujours à une solution et peuvent donc être appliquées sans réfléchir, dans n'importe quel ordre.

Définition 3.82 Soit P un problème d'unification, on appelle *taille* de P la somme des tailles [13] des termes de ses équations.

Une variable $x \in \mathrm{Var}(P)$ est dite *résolue dans* P s'il y a une équation $x \simeq t$ dans P, et que x n'a qu'une occurrence dans P. Le problème P est *sous forme résolue* s'il est de la forme $\{x_1 \simeq t_1, \ldots, x_n \simeq t_n\}$ et les x_i sont toutes résolues dans P. Dans ce cas, on note $\sigma_P = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. △

Il est donc évident que si P est sous forme résolue alors $\sigma_P \in \mathrm{Unif}(P)$. De plus, on voit facilement qu'aucune règle ne s'applique à P. Un problème d'unification sous forme résolue est donc sous forme normale pour $\rightarrow_{\mathbf{U}}$ (ou simplement sous forme $\rightarrow_{\mathbf{U}}$-normale).

Lemme 3.83 *$\rightarrow_{\mathbf{U}}$ termine et toute forme $\rightarrow_{\mathbf{U}}$-normale est résolue ou égale à $\times$.*

Preuve. Soient P et P' deux problèmes d'unification tels que $P \rightarrow_{\mathbf{U}} P'$, on note n (resp. n') le nombre de variables dans $\mathrm{Var}(P)$ (resp. dans $\mathrm{Var}(P')$) qui ne sont pas résolues dans P (resp. dans P'), l (resp. l') la taille de P (resp. de P') et m (resp. m') le nombre d'équations de P (resp. de P') dont le membre gauche n'est pas une variable. Seules les règles EFF, INV, DEC ou RPL ont pu être utilisées.

Si c'est EFF alors $P' = P \setminus \{t \simeq t\}$ donc $\mathrm{Var}(P') \subseteq \mathrm{Var}(P)$ et $l' < l$ (deux occurrences du terme t sont supprimées). Comme P' a exactement les mêmes variables résolues que P, on a $n' \leq n$.

Si la règle utilisée est INV alors $P' = (P \setminus \{t \simeq x\}) \cup \{x \simeq t\}$ avec $t \notin \mathcal{V}$, donc $\mathrm{Var}(P') = \mathrm{Var}(P)$, $m' = m - 1$ et $l' = l$. Toutes les variables résolues dans P le sont également dans P', donc $n' \leq n$.

Si la règle utilisée est DEC alors $P' = (P \setminus \{f(t_1, \ldots, t_n) \simeq f(t'_1, \ldots, t'_n)\}) \cup \{t_1 \simeq t'_1, \ldots, t_n \simeq t'_n\}$, donc $\mathrm{Var}(P') = \mathrm{Var}(P)$ et $l' < l$ (deux occurrences du symbole f sont supprimées). Les variables résolues dans P le sont dans P', donc $n' \leq n$.

Si la règle utilisée est RPL, alors il existe Q, x, t tels que $P = Q \cup \{x \simeq t\}$, $P' = Q[t/x] \cup \{x \simeq t\}$ et $x \in \mathrm{Var}(Q) \setminus \mathrm{Var}(t)$. La variable x n'est pas résolue dans P puisque $x \in \mathrm{Var}(Q)$, mais elle l'est dans P' puisque $x \notin \mathrm{Var}(t) \cup \mathrm{Var}(Q[t/x])$. De plus, si une variable y est résolue dans P, i.e., s'il existe $y \simeq s$ dans Q sans autre occurrence de y dans P (et donc sans occurrence de y dans t), alors l'équation $y \simeq s[t/x]$ est dans $Q[t/x]$. S'il existait une autre occurrence de y dans $Q[t/x]$, alors cette variable apparaîtrait également dans t, ce qui est impossible ; donc y est résolue

13. Voir l'Exercice 32

dans P'. Il y a donc au moins une variable résolue de plus dans P' que dans P, et comme $\mathrm{Var}(P) = \mathrm{Var}(P')$ on obtient $n' < n$.

On voit donc qu'il ne peut exister de suite infinie $(P_i)_{i\in\mathbb{N}}$ telle que $\forall i \in \mathbb{N}$, $P_i \to_{\mathbf{U}} P_{i+1}$ puisque le nombre de variables non résolues ne peut pas croître et que s'il ne décroît pas alors c'est la taille qui décroît, ou qui reste constante mais alors c'est m qui décroît. Ceci prouve que $\to_{\mathbf{U}}$ termine.

Supposons que P est une forme $\to_{\mathbf{U}}$-normale autre que $\times$, soit $t \simeq t'$ une équation quelconque dans P. Si $t \notin \mathcal{V}$ et $t' \notin \mathcal{V}$ alors on peut appliquer la règle DEC ou la règle CFL, ce qui est impossible puisque P est $\to_{\mathbf{U}}$-normale. Si $t \notin \mathcal{V}$ alors $t' \in \mathcal{V}$ et on peut appliquer la règle INV, donc t est une variable et on la note x. Si $t' = x$ alors EFF s'applique, donc $t' \neq x$. Si $x \in \mathrm{Var}(t')$ alors OCC s'applique, donc $x \in \mathrm{Var}(P)\backslash\mathrm{Var}(t')$. Si x apparaît dans une autre équation alors RPL s'applique, donc x est résolue dans P. Donc P est sous forme résolue. •

On voit donc que les règles de transformation permettent effectivement de résoudre P.

Théorème 3.84 *Pour tout problème d'unification P, on a $\mathrm{Unif}(P) = \emptyset$ si et seulement si $P \to_{\mathbf{U}}{}^{\star} \times$, et si $\mathrm{Unif}(P) \neq \emptyset$ alors il existe une forme résolue P' telle que $P \to_{\mathbf{U}}{}^{\star} P'$ et $\sigma_{P'} \in \mathrm{Unif}(P)$.*

PREUVE. Si $\mathrm{Unif}(P) = \emptyset$ alors une suite de transformations à partir de P ne peut terminer sur une forme résolue d'après le Lemme 3.81, donc elle doit terminer sur $\times$ d'après le Lemme 3.83 (la réciproque est évidente). De même si $\mathrm{Unif}(P) \neq \emptyset$ alors une suite de transformations doit terminer sur une forme résolue P' et on a $\sigma_{P'} \in \mathrm{Unif}(P') = \mathrm{Unif}(P)$. •

EXERCICE 55. *Pour tout entier $n \geq 1$ on considère le problème d'unification*

$$P_n = \{x_1 = f(x, x),\ x_2 = f(x_1, x_1), \ldots, x_n = f(x_{n-1}, x_{n-1})\}$$

où $x, x_1, \ldots, x_n$ sont des variables. Quelles sont les variables résolues de P_n ? Quelle est la taille de P_n ? On définit la suite de termes $t_0 = x$ et $t_n = f(t_{n-1}, t_{n-1})$ pour tout $n \geq 1$. Calculer un unificateur σ de P_n en utilisant les termes $t_1, \ldots, t_n$. Quelle est la taille de $\sigma(x_n)$?

Les règles d'unification pouvant être appliquées dans n'importe quel ordre on peut obtenir des solutions distinctes (autrement dit, notre algorithme est non déterministe). Cela signifie-t-il que pour obtenir toutes les solutions possibles il faut explorer toutes les formes résolues accessibles à partir d'un problème d'unification ? Nous allons voir que c'est inutile, et d'autant plus que cela ne permettrait pas d'obtenir toutes les solutions.

Nous avons expliqué au début de la section l'intérêt de résoudre des équations dans T_Σ afin d'éviter de recourir à des valeurs arbitraires pour des unificateurs. Cette notion peut être formalisée de la façon suivante.

Définition 3.85 Une substitution σ est dite *plus générale* qu'une substitution σ' s'il existe une substitution τ telle que $\sigma' = \sigma\tau$. Pour tout problème d'unification P, une substitution σ est dite *unificateur plus général* de P si $\sigma \in \mathrm{Unif}(P)$ et σ est plus général que tout $\sigma' \in \mathrm{Unif}(P)$. △

On remarque que si σ est un unificateur plus général de P alors $\mathrm{Unif}(P)$ est l'ensemble des $\sigma\tau$ pour toute substitution τ ; on peut donc considérer que σ est un générateur de $\mathrm{Unif}(P)$. Mais un ensemble quelconque de substitutions n'admet pas forcément un générateur unique. Nous allons montrer que c'est le cas de tout ensemble non vide de la forme $\mathrm{Unif}(P)$ quelque soit P, ce qui est très simple à partir des résultats précédents.

Lemme 3.86 *Pour toute forme résolue P, σ_P est un unificateur plus général de P.*

PREUVE. Soit $\sigma \in \mathrm{Unif}(P)$, où P est de la forme $\{x_1 \simeq t_1, \ldots, x_n \simeq t_n\}$. Pour tout $1 \leq i \leq n$ on a $x_i\sigma = t_i\sigma$, donc $\sigma(x_i) = t_i\sigma = \sigma_P(x_i)\sigma$. De plus, pour toute autre variable y on a $\sigma(y) = y\sigma = \sigma_P(y)\sigma$. On a donc $\sigma = \sigma_P\sigma$, ce qui prouve que σ_p est plus général que σ. •

Il est donc évident que tout problème qui a un unificateur en a un plus général.

Théorème 3.87 *Pour tout problème d'unification P, si $\mathrm{Unif}(P) \neq \emptyset$ alors pour toute forme résolue P' telle que $P \to_{\mathbf{U}}^{\star} P'$, $\sigma_{P'}$ est un unificateur plus général de P.*

PREUVE. Par le Lemme 3.86 $\sigma_{P'}$ est unificateur plus général de P', donc de P puisque $\mathrm{Unif}(P) = \mathrm{Unif}(P')$ d'après le Lemme 3.81. •

Ce résultat justifie la définition suivante.

Définition 3.88 Pour tout problème d'unification P, si P est unifiable alors on note $\mathrm{unif}(P)$ un unificateur plus général de P. △

Ce qui précède montre qu'il existe une fonction partielle calculable unif, mais elle n'est pas unique. En effet, si tout unificateur obtenu par les règles de transformation est plus général que tous les autres et suffit donc à les engendrer tous, on peut pourtant obtenir des résultats différents selon l'ordre d'application des règles : si $P \to_{\mathbf{U}}^{\star} P'$ et $P \to_{\mathbf{U}}^{\star} P''$, on peut avoir $P' \neq P''$ et donc $\sigma_{P'} \neq \sigma_{P''}$. Mais chacun étant plus général que l'autre, il existe deux substitutions τ et τ' telles que $\sigma_{P''} = \sigma_{P'}\tau$ et $\sigma_{P'} = \sigma_{P''}\tau'$. On a donc $\sigma_{P'} = \sigma_{P'}\tau\tau'$, ce qui prouve que $\tau\tau'$ est l'identité sur $\mathcal{V}$. Il en est de même de $\tau'\tau$; on voit donc que τ et τ' sont des permutations de $\mathcal{V}$ inverses l'une de l'autre. Cela signifie que $\sigma_{P'}$ et $\sigma_{P''}$ ne diffèrent que par les variables qui apparaissent dans les termes $\sigma_{P'}(x)$ et $\sigma_{P''}(x)$ pour $x \in \mathrm{Var}(P)$. En fait, on voit facilement que ces variables peuvent être choisies arbitrairement.

EXEMPLE 3.89 Soient $x, y, z \in \mathcal{V}$ et $P = \{x \simeq y,\ y \simeq z,\ z \simeq x\}$, on peut résoudre P de la façon suivante :

$$\left\{\begin{array}{l} \boxed{x \simeq y} \\ y \simeq z \\ z \simeq x \end{array}\right. \rightarrow_{\text{RPL}} \left\{\begin{array}{l} x \simeq y \\ \boxed{y \simeq z} \\ z \simeq y \end{array}\right. \rightarrow_{\text{RPL}} \left\{\begin{array}{l} x \simeq z \\ y \simeq z \\ \boxed{z \simeq z} \end{array}\right. \rightarrow_{\text{EFF}} \left\{\begin{array}{l} x \simeq z \\ y \simeq z \end{array}\right.$$

ce qui donne l'unificateur $\sigma_1 = \{x \mapsto z, y \mapsto z\}$. On peut utiliser un ordre différent :

$$\left\{\begin{array}{l} x \simeq y \\ y \simeq z \\ \boxed{z \simeq x} \end{array}\right. \rightarrow_{\text{RPL}} \left\{\begin{array}{l} x \simeq y \\ \boxed{y \simeq x} \\ z \simeq x \end{array}\right. \rightarrow_{\text{RPL}} \left\{\begin{array}{l} \boxed{x \simeq x} \\ y \simeq x \\ z \simeq x \end{array}\right. \rightarrow_{\text{EFF}} \left\{\begin{array}{l} y \simeq x \\ z \simeq x \end{array}\right.$$

ce qui donne l'unificateur $\sigma_2 = \{y \mapsto x, z \mapsto x\}$. Soit $\pi = \{x \mapsto z, z \mapsto x\}$, on a $\pi^{-1} = \pi$ et on vérifie aisément que $\sigma_1\pi = \sigma_2$ et $\sigma_2\pi = \sigma_1$. ◊

EXERCICE 56. *Soit $\sigma_3 = \{x \mapsto u, y \mapsto u, z \mapsto u\}$ où u est une variable distincte de x, y, z. Montrer que σ_3 est un unificateur plus général de P dans l'exemple précédent. Peut-on obtenir σ_3 en appliquant les règles de transformation à P ?*

3.8 Forme clausale

Maintenant que nous savons résoudre des équations sur les termes, il nous faut encore déterminer quelles équations nous devons résoudre pour prouver l'insatisfaisabilité d'un ensemble de formules. Nous avons vu qu'il suffit pour cela de trouver les bonnes instances fermées d'une formule universelle. Supposons par exemple qu'on obtient la formule universelle $\forall x(\neg p(f(x)) \wedge p(x))$. Cette formule est évidemment insatisfaisable : il suffit d'instancier x avec $a \in \Sigma_0^{\text{F}}$ et avec $f(a)$ pour obtenir un ensemble fini insatisfaisable d'instance fermées $\{\neg p(f(a)) \wedge p(a), \neg p(f(f(a))) \wedge p(f(a))\}$.

Cependant, l'équation $f(x) \simeq x$ n'a pas de solution, ce qui exprime le fait qu'on ne peut obtenir de contradiction avec une seule instance de x. Une équation peut-elle rendre compte de la possibilité d'obtenir une contradiction avec des instances distinctes de x ? Cela nécessiterait de dissocier les instances de $p(x)$ de celles de $\neg p(f(x))$ en remplaçant l'une des occurrences de x par une nouvelle variable y. Mais on ne peut pas en général séparer ainsi une variable en deux.

EXERCICE 57. *Considérons la formule universelle $\forall x[(p(x) \wedge \neg p(a)) \vee (\neg p(x) \wedge p(b))]$. Cette formule est-elle satisfaisable ? Considérons maintenant la formule universelle $\forall x \forall y[(p(x) \wedge \neg p(a)) \vee (\neg p(y) \wedge p(b))]$ obtenue en dissociant les deux occurrences de x ; cette formule est-elle satisfaisable ?*

Cependant, nous pouvons nous appuyer sur le fait que le quantificateur universel est une généralisation de la conjonction, d'où on déduit facilement :

Lemme 3.90 *Pour toutes formules $\varphi, \psi \in \mathrm{F}_\Sigma$, on a $\forall x\,(\varphi \wedge \psi) \equiv_1 \forall x\,\varphi \wedge \forall x\,\psi$.*

PREUVE. Pour toute interprétation $\mathcal{I} \in \mathrm{I}_\Sigma$ et pour toute valuation $\theta \in \mathcal{I}^\mathcal{V}$, on a

$$\begin{array}{lll}
\mathcal{I}, \theta \models_1 \forall x\,\varphi \wedge \forall x\,\psi & \text{ssi} & \mathcal{I}, \theta \models_1 \forall x\,\varphi \text{ et } \mathcal{I}, \theta \models_1 \forall x\,\psi \\
& \text{ssi} & \forall v \in \mathcal{I},\ \mathcal{I}, \theta[x \mapsto v] \models_1 \varphi \text{ et } \forall v \in \mathcal{I},\ \mathcal{I}, \theta[x \mapsto v] \models_1 \psi \\
& \text{ssi} & \forall v \in \mathcal{I},\ \mathcal{I}, \theta[x \mapsto v] \models_1 \varphi \text{ et } \mathcal{I}, \theta[x \mapsto v] \models_1 \psi \\
& \text{ssi} & \forall v \in \mathcal{I},\ \mathcal{I}, \theta[x \mapsto v] \models_1 \varphi \wedge \psi \\
& \text{ssi} & \mathcal{I}, \theta \models_1 \forall x\,(\varphi \wedge \psi).
\end{array}$$

On a bien $\forall x\,(\varphi \wedge \psi) \equiv_1 \forall x\,\varphi \wedge \forall x\,\psi$. •

Nous avons également vu (après l'Exercice 41) que par renommage de variables,

$$\forall x\,\varphi \wedge \forall x\,\psi \equiv_1 \forall x\,\varphi \wedge \forall y\,\psi[y/x].$$

En suivant notre exemple, nous pouvons donc obtenir

$$\forall x\,(\neg p(f(x)) \wedge p(x)) \equiv_1 \forall x\,\neg p(f(x)) \wedge \forall y\,p(y),$$

que nous pouvons maintenant transformer en un ensemble de formules universelles (comme dans l'Exercice 20) :

$$\{\forall x\,\neg p(f(x)),\ \forall y\,p(y)\}\,.$$

Nous pouvons alors facilement trouver une instance fermée insatisfaisable de $\{\forall x\,\neg p(f(x)),\ \forall y\,p(y)\}$ en résolvant l'équation $f(x) \simeq y$, ce qui donne l'unificateur $y \mapsto f(x)$. On peut donc instancier x par un terme $t \in \overline{\mathrm{T}}_\Sigma$ quelconque, et y par $f(t)$; cela donne forcément un ensemble insatisfaisable $\{\neg p(f(t)),\ p(f(t))\}$.

Il est donc possible d'obtenir par unification des instanciations contradictoires en dissociant les variables, ce qui n'est possible que si la matrice commence par une conjonction. Nous allons donc commencer par mettre la matrice sous forme normale conjonctive, ce qui se fait exactement comme en logique propositionnelle puisque les matrices sont des formules de $\mathrm{P}(\mathrm{A}_\Sigma)$ (Théorème 3.12). Par exemple, la première formule de l'Exercice 57 est équivalente à :

$$\forall x\,[(p(x) \vee \neg p(x)) \ \wedge\ (\neg p(a) \vee \neg p(x)) \ \wedge\ (p(x) \vee p(b)) \ \wedge\ (\neg p(a) \vee p(b))]$$

et donc à

$$\forall x\,(p(x) \vee \neg p(x)) \ \wedge\ \forall y\,(\neg p(a) \vee \neg p(y)) \ \wedge\ \forall z\,(p(z) \vee p(b)) \ \wedge\ (\neg p(a) \vee p(b))$$

qui est satisfaisable si et seulement si l'ensemble de formules universelles

$$\{\forall x\,(p(x) \vee \neg p(x)),\ \forall y\,(\neg p(a) \vee \neg p(y)), \forall z\,(p(z) \vee p(b)), \neg p(a) \vee p(b)\}$$

est satisfaisable. De même, la deuxième formule de l'Exercice 57 est satisfaisable si et seulement si l'ensemble

$$\{\forall x \forall y\,(p(x) \vee \neg p(y)),\ \forall y\,(\neg p(a) \vee \neg p(y)), \forall z\,(p(z) \vee p(b)), \neg p(a) \vee p(b)\}$$

est satisfaisable.

Nous allons donc réutiliser les notions déjà développées pour les formules propositionnelles, en prenant l'ensemble de symboles propositionnels $\mathcal{S} = \mathrm{A}_\Sigma$. En particulier, nous utiliserons la notion de littéral (voir la Définition 2.7) et de clause (voir la Définition 2.8). Il nous faut cependant modifier la notion de forme clausale.

Définition 3.91 Un *littéral* (resp. une *clause*) dans F_Σ est un littéral (resp. une clause) dans $\mathrm{P}(\mathrm{A}_\Sigma)$. Pour toute clause C dans F_Σ on note $\mathrm{Var}(C) = \mathrm{VL}(C)$.

Pour toute formule fermée φ, on appelle *forme clausale* de φ tout ensemble fini F de clauses tel que φ est satisfaisable si et seulement si $\forall^\star F$ est satisfaisable, où $\forall^\star F = \{\forall^\star C \mid C \in F\}$. △

$\forall^\star F$ est donc un ensemble fini de formules universelles obtenues à partir des clauses de F.

Théorème 3.92 *Pour toute formule $\varphi \in \overline{\mathrm{F}}_\Sigma$ on peut calculer une forme clausale F de φ.*

Preuve. D'après le Théorème 3.70, le Lemme 3.90 et l'Exercice 20. •

Corollaire 3.93 *Pour tout ensemble fini de formules $E \subseteq \overline{\mathrm{F}}_\Sigma$ on peut calculer un ensemble de clauses F telle que E est satisfaisable si et seulement si $\forall^\star F$ est satisfaisable.*

Cela suggère que nous pouvons généraliser la méthode de résolution propositionnelle aux clauses du premier ordre. Nous allons donc définir des règles d'inférence permettant de déduire des clauses à partir d'autres clauses, mais en gardant à l'esprit qu'on établit implicitement des déductions sur des formules universelles. Une clause C doit toujours être vue comme une abréviation de la formule $\forall^\star C$. Ces règles d'inférence utiliseront l'unification entre littéraux qu'il nous faut donc définir.

Définition 3.94 Soient m littéraux $\alpha_1, \ldots, \alpha_m \in \mathrm{A}_\Sigma$ avec $m \geq 2$. S'il existe $n \geq 0$, un prédicat $p \in \Sigma^{\mathrm{P}}_n$ et des termes $t^j_i \in \mathrm{T}_\Sigma$ tels que pour tout $1 \leq j \leq m$ on a $\alpha_j = p(t^j_1, \ldots, t^j_n)$, et si le problème d'unification $P = \left\{t^1_i \simeq t^j_i \mid 1 \leq i \leq n,\ 1 < j \leq m\right\}$ admet une solution, alors on dit que les littéraux $\alpha_1, \ldots, \alpha_m$ (ainsi que leur négation) sont *unifiables*, et on note $\mathrm{unif}(\alpha_1, \ldots, \alpha_m)$ et $\mathrm{unif}(\neg\alpha_1, \ldots, \neg\alpha_m)$ pour un unificateur le plus général de P. Dans le cas contraire, ces littéraux sont non unifiables. △

EXEMPLE 3.95 $\text{unif}(\neg p(x, f(x)), \neg p(g(y), y), \neg p(g(f(z)), z))$ est un unificateur le plus général du problème $P = \{x \simeq g(y), x \simeq g(f(z)), f(x) \simeq y, f(x) \simeq z\}$. ◊

Afin de dissocier les variables de clauses distinctes nous auront besoin de substitutions particulières.

Définition 3.96 On note $\text{Sym}(\mathcal{V})$ l'ensemble des bijections de $\mathcal{V}$ dans $\mathcal{V}$. Pour toute clause C et tout $\pi \in \text{Sym}(V)$, la clause $C\pi$ est une *variante* de C. △

Comme $\mathcal{V} \subseteq \text{T}_\Sigma$, les éléments de $\text{Sym}(\mathcal{V})$ sont bien des substitutions. De plus, comme $\mathcal{V}$ est infini, pour toutes clauses C_1 et C_2 il existe une bijection $\pi \in \text{Sym}(\mathcal{V})$ telle que $\text{Var}(C_1) \cap \text{Var}(C_2\pi) = \emptyset$. Par exemple, si $C_1 = \neg p(f(x))$ et $C_2 = p(x)$, soit $\pi = \{x \mapsto y, y \mapsto x\}$, alors $C_2\pi = p(y)$. La clause $p(y)$ est une variante de $p(x)$. Le nom des variables liées n'ayant pas d'importance, on a $\forall^\star C_2\pi \equiv_1 \forall^\star C_2$.

3.9 Résolution et factorisation

Nous pouvons maintenant formaliser une règle de résolution pour les clauses du premier ordre.

Définition 3.97 (Règle de résolution) Soient deux clauses C_1, C_2, pour toute clause $R \in \text{M}_\Sigma$ on a $\langle C_1, C_2, R\rangle \in \text{R}_1$ si et seulement s'il existe deux clauses D_1, D_2, deux littéraux L_1, L_2, une substitution σ et $\pi \in \text{Sym}(\mathcal{V})$ tels que :
- $C_1 = D_1 \dot\vee L_1$,
- $C_2 = D_2 \dot\vee L_2$,
- $\text{Var}(C_1) \cap \text{Var}(C_2\pi) = \emptyset$
- L_1 et $\overline{L_2\pi}$ sont unifiables,
- $\sigma = \text{unif}(L_1, \overline{L_2\pi})$,
- $R = (D_1 \vee D_2\pi)\sigma$.

La clause R est alors une *résolvante* de C_1 et C_2. On a donc

$$\frac{D_1 \dot\vee L_1 \quad D_2 \dot\vee L_2}{(D_1 \vee D_2\pi)\sigma} \text{ si } \begin{cases} \pi \in \text{Sym}(\mathcal{V}) \\ \text{Var}(D_1 \dot\vee L_1) \cap \text{Var}(D_2\pi \dot\vee L_2\pi) = \emptyset \\ \sigma = \text{unif}(L_1, \overline{L_2\pi}) \end{cases}$$ △

EXEMPLE 3.98 On peut obtenir l'inférence suivante :

$$\frac{\neg p(f(x)) \quad p(x) \vee q(x)}{q(f(x))}$$

Il faut pour cela appliquer $\pi = \{x \mapsto y, y \mapsto x\}$ à la prémisse $p(x) \vee q(x)$ qui devient donc $p(y) \vee q(y)$. L'unification de $\neg p(f(x))$ avec $\overline{p(y)} = \neg p(y)$ donne $\sigma = \{y \mapsto f(x)\}$. La résolvante est donc $q(y)\sigma = q(f(x))$. ◊

EXEMPLE 3.99 On peut réfuter la forme clausale suivante

$$F \;=\; \{p(a,b),\ \neg p(x,y) \vee \neg p(y,z) \vee p(x,z),\ \neg p(x,y) \vee p(y,x),\ \neg p(a,a)\}$$

en utilisant la règle de résolution :

$$\dfrac{\dfrac{\dfrac{p(a,b) \qquad \neg p(x,y) \vee \neg p(y,z) \vee p(x,z)}{\neg p(b,z) \vee p(a,z)} \qquad \dfrac{p(a,b) \qquad \neg p(x,y) \vee p(y,x)}{p(b,a)}}{p(a,a)} \qquad \neg p(a,a)}{\square}$$

le Théorème 3.103 permettra d'en déduire l'insatisfaisabilité de F. ◊

Il est facile de voir que cette règle généralise la règle de résolution propositionnelle. Plus exactement, la restriction de R_1 aux formules propositionnelles est exactement Res_0. Cependant, cette règle ne suffit pas pour obtenir la complétude réfutationnelle. Considérons par exemple les deux clauses $C_1 = p(x) \vee p(y)$ et $C_2 = \neg p(x) \vee \neg p(y)$, on obtient comme résolvante la clause $C_3 = p(x) \vee \neg p(y)$. Mais à partir de ces trois clauses les seules résolvantes possibles sont des variantes de C_1, C_2 ou C_3. Il est donc impossible d'engendrer la clause vide à l'aide de la règle R_1. Pourtant, la forme clausale $\{C_1, C_2\}$ est insatisfaisable, comme on peut s'en rendre compte en instanciant toutes les variables par un même terme a : on obtient ainsi les clauses $p(a) \vee p(a)$ et $\neg p(a) \vee \neg p(a)$, qui sont contradictoires.

On voit dans cet exemple que la règle de résolution ne permet pas d'instancier les variables x et y de C_1 par le même terme (parce qu'on ne peut utiliser qu'un unificateur le plus général), alors qu'il est indispensable de réaliser que les deux littéraux de C_1 ont au moins une instance commune (par exemple $p(a)$). Il ne suffit donc pas de trouver des instances communes (complémentaires) entre clauses, il faut aussi pouvoir trouver des instances communes entre les littéraux d'une même clause. C'est pourquoi il est nécessaire d'utiliser également une règle de *factorisation*.

Définition 3.100 (Règle de factorisation) Pour toutes clauses C, C' on a $\langle C, C' \rangle \in \mathrm{F}_1$ si et seulement s'il existe une clause D et $m \geq 2$ littéraux unifiables $L_1, \ldots, L_m$ tels que $C = D \dot{\vee} L_1 \dot{\vee} \cdots \dot{\vee} L_m$ et $C' = (D \vee L_1)\sigma$, où $\sigma = \mathrm{unif}(L_1, \ldots, L_m)$. La clause C' est un *facteur* de C. On a donc

$$\frac{D \dot{\vee} L_1 \dot{\vee} \cdots \dot{\vee} L_m}{(D \vee L_1)\sigma} \text{ si } \sigma = \mathrm{unif}(L_1, \ldots, L_m).$$

On note $\vdash_{\mathrm{R}}$ la relation de déduction basée sur les règles R_1 et F_1 entre clauses.△

EXEMPLE 3.101 $q(f(y), y) \vee p(f(y))$ est un facteur de $p(x) \vee q(x,y) \vee p(f(y))$. ◊

EXEMPLE 3.102 En combinant les deux règles on peut facilement réfuter l'ensemble $\{C_1,\ C_2\}$ évoqué ci-dessus :

$$\dfrac{\dfrac{p(x) \vee p(y)}{p(x)} \qquad \dfrac{\neg p(x) \vee \neg p(y)}{\neg p(x)}}{\square}$$

$\Diamond$

Afin de prouver la correction de ces règles d'inférence, nous rappelons qu'elles portent implicitement sur des formules universelles ; c'est donc au moyen de celles-ci que l'on doit énoncer la correction.

Théorème 3.103 *Pour toute forme clausale F et toute clause D, si $F \vdash_{\mathrm{R}} D$ alors $\forall^\star F \models_1 \forall^\star D$.*

PREUVE. Il suffit de le prouver pour les deux règles d'inférence. Si D est obtenue par résolution alors il existe des clauses D_1, D_2, une bijection $\pi \in \mathrm{Sym}(\mathcal{V})$ et des littéraux L_1, L_2 tels que $D_1 \dot\vee L_1 \in F$, $D_2 \dot\vee L_2 \in F$, $\mathrm{Var}(D_1 \dot\vee L_1) \cap \mathrm{Var}(D_2 \dot\vee L_2) = \emptyset$, L_1 et $\overline{L_2\pi}$ sont unifiables et $D = (D_1 \vee D_2\pi)\sigma$ où $\sigma = \mathrm{unif}(L_1, \overline{L_2\pi})$.

Soit une interprétation $\mathcal{I}$ telle que $\mathcal{I} \models \forall^\star F$, d'après le Lemme 3.45 on a $\forall^\star(D_1 \vee L_1) \models_1 \forall^\star(D_1\sigma \vee L_1\sigma)$, donc pour toute valuation $\theta \in \mathcal{I}^\mathcal{V}$ on a $\mathcal{I}, \theta \models_1 D_1\sigma \vee L_1\sigma$. De plus on a $\forall^\star(D_2 \vee L_2) \models_1 \forall^\star(D_2 \vee L_2)\pi$ et de même que ci-dessus on a donc $\mathcal{I}, \theta \models_1 D_2\pi\sigma \vee L_2\pi\sigma$. Mais $L_1\sigma = \overline{L_2\pi\sigma}$, il est donc évident que $\mathcal{I}, \theta \models_1 D_1\sigma \vee D_2\pi\sigma$ (comme dans le cas de la résolution propositionnelle), et donc que $\mathcal{I} \models_1 \forall^\star D$.

Si D est obtenue par factorisation, alors il existe une clause C et m littéraux $L_1, \ldots, L_m$ tels que $C \dot\vee L_1 \dot\vee \cdots \dot\vee L_m \in F$, les littéraux $L_1, \ldots, L_m$ sont unifiables et $D = (C \vee L_1)\sigma$ où $\sigma = \mathrm{unif}(L_1, \ldots, L_m)$. Soit $\mathcal{I} \models \forall^\star F$, on a $\forall^\star(C \vee L_1 \vee \cdots \vee L_m) \models_1 \forall^\star(C\sigma \vee L_1\sigma \vee \cdots \vee L_m\sigma)$, mais $L_1\sigma = \cdots = L_m\sigma$, donc $\mathcal{I} \models_1 \forall^\star D$. On a donc $\forall^\star F \models_1 \forall^\star D$. •

Il est donc clair qu'on ne peut engendrer de clause vide qu'à partir d'une forme clausale insatisfaisable. Afin de prouver la complétude de cette méthode de preuve, autrement dit qu'on peut toujours obtenir la clause vide en partant d'une forme clausale insatisfaisable, nous allons nous appuyer sur la résolution propositionnelle. Il s'agit de calquer une réfutation de clauses du premier ordre sur une réfutation propositionnelle de leurs instances fermées. Chaque inférence propositionnelle peut être calquée grâce au lemme suivant.

Lemme 3.104 (de relèvement) *Pour toutes clauses C_1, C_2, toutes substitutions fermées τ_1, τ_2 et toute clause fermée R telles que $\langle C_1\tau_1, C_2\tau_2, R\rangle \in \mathrm{Res}_0$, il existe une clause D dont R est une instance fermée et telle que $C_1, C_2 \vdash_{\mathrm{R}} D$.*

PREUVE. Par définition de Res_0 il existe deux clauses fermées C_1', C_2' et un littéral fermé l tels que $C_1\tau_1 = C_1' \dot\vee l$, $C_2\tau_2 = C_2' \dot\vee \bar l$ et $R = C_1' \vee C_2'$.

Comme l'instance $C_1\tau_1$ de C_1 est $C_1' \dot\vee l$, il doit y avoir $m \geq 1$ littéraux distincts $L_1, \ldots, L_m$ dans C_1 tels que $L_1\tau_1 = \cdots = L_m\tau_1 = l$, et donc une clause B_1 telle que

$C_1 = B_1 \dot\vee L_1 \dot\vee \cdots \dot\vee L_m$ et $B_1\tau_1 = C'_1$. Les littéraux $L_1, \ldots, L_m$ sont unifiables par τ_1, ce qui garantit l'existence d'un unificateur $\sigma_1 = \mathrm{unif}(L_1, \ldots, L_m)$ et d'une substitution μ_1 tels que $\tau_1 = \sigma_1\mu_1$. Si $m \geq 2$ et on pose $D_1 = (B_1 \vee L_1)\sigma_1$ qui est un facteur de C_1 ; on a donc $C_1 \vdash_{\mathrm{R}} D_1$. Si $m = 1$ on pose $D_1 = C_1$; dans ce cas σ_1 est l'identité et a donc également $D_1 = (B_1 \dot\vee L_1)\sigma_1$ et $C_1 \vdash_{\mathrm{R}} D_1$.

De même il y a $n \geq 1$ littéraux $L'_1, \ldots, L'_n$ dans C_2 tels que $L'_1\tau_2 = \cdots = L'_n\tau_2 = \overline{l}$, et une clause B_2 telle que $C_2 = B_2 \dot\vee L'_1 \dot\vee \cdots \dot\vee L'_n$ et $B_2\tau_2 = C'_2$. Il existe donc un unificateur $\sigma_2 = \mathrm{unif}(L'_1, \ldots, L'_n)$ et une substitution μ_2 tels que $\tau_2 = \sigma_2\mu_2$. La clause $D_2 = (B_2 \vee L'_1)\sigma_2$ est un facteur de C_2 (ou est égale à C_2 si $n = 1$) et on a $C_2 \vdash_{\mathrm{R}} D_2$.

Soient $\pi \in \mathrm{Sym}(\mathcal{V})$ telle que $\mathrm{Var}(D_1) \cap \mathrm{Var}(D_2\pi) = \emptyset$ et μ une substitution égale à μ_1 sur $\mathrm{Var}(D_1)$ et à $\pi^{-1}\mu_2$ sur $\mathrm{Var}(D_2\pi)$. Comme $D_1 = B_1\sigma_1 \vee L_1\sigma_1$ on a $B_1\sigma_1\mu = B_1\sigma_1\mu_1 = B_1\tau_1 = C'_1$ et $L_1\sigma_1\mu = L_1\sigma_1\mu_1 = L_1\tau_1 = l$. On obtient de même $B_2\sigma_2\pi\mu = B_2\sigma_2\pi\pi^{-1}\mu_2 = C'_2$ et $L'_1\sigma_2\pi\mu = L'_1\tau_2 = \overline{l}$.

Si $L_1\sigma_1 \in \mathrm{Lit}(B_1\sigma_1)$ alors $L_1\sigma_1\mu \in \mathrm{Lit}(B_1\sigma_1\mu)$ et donc $l \in \mathrm{Lit}(C'_1)$, mais cela est impossible puisque $C_1\tau_1 = C'_1 \dot\vee l$; on peut donc écrire $D_1 = B_1\sigma_1 \dot\vee L_1\sigma_1$. De la même façon on peut écrire $D_2 = B_2\sigma_2 \dot\vee L'_1\sigma_2$. De plus, il est clair que μ est un unificateur de $L_1\sigma_1$ et $\overline{L'_1\sigma_2\pi}$, donc il existe un unificateur $\sigma = \mathrm{unif}(L_1\sigma_1, \overline{L'_1\sigma_2\pi})$ et une substitution τ tels que $\mu = \sigma\tau$. En posant $D = (B_1\sigma_1 \vee B_2\sigma_2\pi)\sigma$ on a $\langle D_1, D_2, D\rangle \in \mathrm{R}_1$ et donc $C_1, C_2 \vdash_{\mathrm{R}} D$. De plus

$$D\tau \;=\; (B_1\sigma_1 \vee B_2\sigma_2\pi)\sigma\tau \;=\; B_1\sigma_1\mu \vee B_2\sigma_2\pi\mu \;=\; C'_1 \vee C'_2 \;=\; R. \qquad \bullet$$

Exemple 3.105 Considérons les deux clause $C_1 = q(x, y) \vee p(x) \vee p(y)$ et $C_2 = \neg p(f(x)) \vee p(x)$ auxquelles nous appliquons les substitutions fermées $\tau_1 = \{x \mapsto f(a), y \mapsto f(a)\}$ et $\tau_2 = \{x \mapsto a\}$. On peut alors appliquer la résolution propositionnelle à $C_1\tau_1$ et $C_2\tau_2$, avec $l = p(f(a))$, $C'_1 = q(f(a), f(a))$ et $C'_2 = p(a)$, pour obtenir la résolvante $R = q(f(a), f(a)) \vee p(a)$.

Nous avons deux littéraux distincts dans C_1 qui s'instancient en l par τ_1, c'est $p(x)$ et $p(y)$; il faut donc appliquer la factorisation à C_1. On calcule donc $\sigma_1 = \mathrm{unif}(p(x), p(y)) = \{y \mapsto x\}$ ce qui donne $\mu_1 = \{x \mapsto f(a)\}$ (de sorte que $\tau_1 = \sigma_1\mu_1$). Cette factorisation produit la clause $D_1 = q(x, x) \vee p(x)$. La clause C_2 n'a pas besoin d'être factorisée, on a donc $D_2 = C_2$, σ_2 est l'identité et $\mu_2 = \tau_2$.

D_1 et D_2 ont la variable x en commun, nous utilisons donc $\pi = \{x \mapsto z, z \mapsto x\}$, de sorte que $D_2\pi = \neg p(f(z)) \vee p(z)$. Par construction on obtient $\mu = \{x \mapsto f(a), z \mapsto a\}$, $\sigma = \mathrm{unif}(p(x), p(f(z))) = \{x \mapsto f(z)\}$ et donc $\tau = \{z \mapsto a\}$ (de sorte que $\mu = \sigma\tau$). La clause $D = (q(x, y)\sigma_1 \vee p(x)\sigma_2\pi)\sigma = q(f(z), f(z)) \vee p(z)$ est une résolvante (du premier ordre) des clauses D_1 et D_2, et on a $D\tau = R$.

En résumé, l'inférence propositionnelle

$$\frac{q(f(a), f(a)) \vee p(f(a)) \vee p(f(a)) \qquad \neg p(f(a)) \vee p(a)}{q(f(a), f(a)) \vee p(a)}$$

peut être calquée au premier ordre par les inférences suivantes

$$\dfrac{\dfrac{q(x,y) \vee p(x) \vee p(y)}{q(x,x) \vee p(x)} \qquad \neg p(f(x)) \vee p(x)}{q(f(z), f(z)) \vee p(z)}$$

dont la conclusion est plus générale que la précédente. C'est pour garantir cette propriété que les règles de factorisation et de résolution utilisent un unificateur le plus général. ♢

EXERCICE 58. *On peut aussi appliquer la résolution sur $C_2\tau_1$ et $C_2\tau_2$:*

$$\dfrac{\neg p(f(f(a))) \vee p(f(a)) \qquad \neg p(f(a)) \vee p(a)}{\neg p(f(f(a))) \vee p(a)}$$

Comme dans l'exemple précédent, donner les valeurs de C'_1, C'_2, l, π, μ, σ, τ et D.

Nous allons donc pouvoir construire une réfutation d'une forme clausale insatisfaisable F avec les règles R_1 et F_1 à partir d'une réfutation de $\mathrm{IF}(F)$ avec la règle Res_0.

Théorème 3.106 (Complétude réfutationnelle de la résolution) *Si F est une forme clausale telle que $\forall^\star F$ est insatisfaisable, alors $F \vdash_{\mathrm{R}} \square$.*

PREUVE. D'après le Théorème 3.52 $\mathrm{IF}(F)$ admet un sous-ensemble fini insatisfaisable E, et d'après le Théorème 2.30 on a $E \vdash_{\mathrm{r}} \square$: il existe une déduction par résolution propositionnelle $(C_1, \ldots, C_n)$ de $C_n = \square$ à partir de E. On montre par induction sur i que, pour tout $1 \leq i \leq n$ il existe une clause D_i et une substitution fermée τ_i telles que $F \vdash_{\mathrm{R}} D_i$ et $D_i\tau_i = C_i$.

Supposons la propriété vraie pour tout $j < i$. Si $C_i \in E$ alors $C_i \in \mathrm{IF}(F)$ donc il existe une clause $D_i \in F$ et une substitution fermé τ_i telles que $D_i\tau_i = C_i$; on a donc trivialement $F \vdash_{\mathrm{R}} D_i$. Sinon, il existe deux clauses C_j, C_k, avec $j, k < i$ telles que $\langle C_j, C_k, C_i\rangle \in \mathrm{Res}_0$. Par hypothèse d'induction il existe deux clauses D_j, D_k et deux substitutions fermées τ_j, τ_k telles que $F \vdash_{\mathrm{R}} \{D_j, D_k\}$, $D_j\tau_j = C_j$ et $D_k\tau_k = C_k$. On a donc $\langle D_j\tau_j, D_k\tau_k, C_i\rangle \in \mathrm{Res}_0$ et d'après le Lemme 3.104 il existe une clause D_i telle que $F \vdash_{\mathrm{R}} D_i$ et une substitution fermée τ_i telle que $D_i\tau_i = C_i$, ce qui conclut l'induction.

Il existe donc une clause D_n et une substitution τ_n telles que $F \vdash_{\mathrm{R}} D_n$ et $D_n\tau_n = \square$. Il est donc évident que $D_n = \square$ et on a donc $F \vdash_{\mathrm{R}} \square$. •

La méthode de résolution est donc une procédure de semi-décision pour l'insatisfaisabilité en logique du premier ordre. Elle est bien plus efficace que celle qui nous avait servi à prouver le Corollaire 3.76.

EXERCICE 59. *On présente parfois la méthode de résolution avec une seule règle :*

$$\frac{D_1 \dot\vee L_1 \dot\vee \cdots \dot\vee L_n \quad D_2 \dot\vee L'_1 \dot\vee \cdots \dot\vee L'_m}{(D_1 \vee D_2\pi)\sigma}$$

$$\text{si} \begin{cases} \pi \in \mathrm{Sym}(\mathcal{V}) \\ \mathrm{Var}(D_1 \dot\vee L_1 \dot\vee \cdots \dot\vee L_n) \cap \mathrm{Var}((D_2 \dot\vee L'_2 \dot\vee \cdots \dot\vee L'_m)\pi) = \emptyset \\ n, m \geq 1 \\ \sigma = \mathrm{unif}(L_1, \ldots, L_n, \overline{L'_1\pi}, \ldots, \overline{L'_m\pi}) \end{cases}$$

Montrer que cette règle est complète pour la réfutation.

Chapitre 4

Exercices corrigés

Exercice 1.

1. Vrai. Si $A \models_{\mathcal{L}} C \cup D$ alors pour tout $f \in C \cup D$ on a $A \models_{\mathcal{L}} f$, c'est donc vrai en particulier pour tout $f \in C$, donc $A \models_{\mathcal{L}} C$.
2. Faux. Si $A = C = \emptyset$ et $D = \{f\}$ où f est une formule non valide, on a $A \models_{\mathcal{L}} C$ et $A \not\models_{\mathcal{L}} C \cup D$.
3. Vrai. On suppose $A \models_{\mathcal{L}} C$; pour toute interprétation I telle que $I \models_{\mathcal{L}} A \cup B$, on a en particulier $I \models_{\mathcal{L}} A$ donc $I \models_{\mathcal{L}} C$ d'après l'hypothèse, ce qui prouve que $A \cup B \models_{\mathcal{L}} C$.
4. Vrai. Si $A \models_{\mathcal{L}} C$ et $B \models_{\mathcal{L}} D$ alors $A \cup B \models_{\mathcal{L}} C$ et $A \cup B \models_{\mathcal{L}} D$ d'après le point précédent. Pour tout $f \in C \cup D$, on a $f \in C$ ou $f \in D$, et dans les deux cas $A \cup B \models_{\mathcal{L}} f$, donc $A \cup B \models_{\mathcal{L}} C \cup D$.
5. Vrai. Si $A \models_{\mathcal{L}} B$ et $B \models_{\mathcal{L}} C$ alors pour toute interprétation I telle que $I \models_{\mathcal{L}} A$, on obtient $I \models_{\mathcal{L}} B$ puis $I \models_{\mathcal{L}} C$ par les hypothèses ; on a donc $A \models_{\mathcal{L}} C$.

Exercice 2. Soit H un ensemble de formules. On procède par récurrence forte sur la longueur des déductions : on suppose que toute déduction à partir de H dans $\mathcal{S}$ de longueur strictement inférieure à $n \in \mathbb{N}$ est correcte pour $\mathcal{L}$. Soit $(f_1, \ldots, f_n)$ une déduction à partir de H dans $\mathcal{S}$. Il y a deux possibilités pour f_n : soit $f_n \in H$ et alors $H \models_{\mathcal{L}} f_n$, soit il existe m prémisses $f_{i_1}, \ldots, f_{i_m}$ telles que $i_1 < i, \ldots, i_m < n$ et $(f_{i_1}, \ldots, f_{i_m}, f_n)$ est une inférence dans $\mathcal{S}$, et on a $f_{i_1}, \ldots, f_{i_m} \models_{\mathcal{L}} f_n$ puisque $\mathcal{S}$ est correct pour $\mathcal{L}$ ainsi que $H \models_{\mathcal{L}} f_{i_1}$, …, $H \models_{\mathcal{L}} f_{i_m}$ par hypothèse d'induction, donc par transitivité on a $H \models_{\mathcal{L}} f_n$. La déduction $(f_1, \ldots, f_n)$ est donc correcte dans $\mathcal{L}$.

Exercice 6. On montre la contraposée. Si $\mathcal{L}$ n'est pas incomplète alors elle admet un système d'inférence $\mathcal{S}$ correct et complet pour $\mathcal{L}$, donc tel que l'ensemble des formules valides de $\mathcal{L}$ est exactement l'ensemble des théorèmes de $\mathcal{S}$. D'après le Théorème 1.5, cet ensemble est récursivement énumérable.

EXERCICE 7. Si $\mathcal{S}$ n'a pas de théorème alors il est consistant. Sinon, il existe un théorème f de $\mathcal{S}$, comme $\mathcal{S}$ est correct pour $\mathcal{L}$ alors $\models_{\mathcal{L}} f$, donc $\neg f$ est insatisfaisable (Proposition 1.16) et ne peut donc être un théorème de $\mathcal{S}$ qui est donc consistant.

EXERCICE 8.

1. On a $\mathbf{0} \vdash_{D_1} \begin{smallmatrix}\bullet&\circ&\circ\\ \circ&\bullet&\circ\\ \circ&\circ&\bullet\end{smallmatrix} \vdash_{D_2} \begin{smallmatrix}\bullet&\circ&\bullet\\ \circ&\circ&\circ\\ \bullet&\circ&\bullet\end{smallmatrix} \vdash_{C_1} \begin{smallmatrix}\circ&\circ&\bullet\\ \bullet&\circ&\circ\\ \circ&\circ&\bullet\end{smallmatrix} \vdash_{C_3} \begin{smallmatrix}\circ&\circ&\circ\\ \bullet&\circ&\bullet\\ \circ&\circ&\circ\end{smallmatrix} \vdash_{L_2} \begin{smallmatrix}\circ&\circ&\circ\\ \circ&\bullet&\circ\\ \circ&\circ&\circ\end{smallmatrix}$
   On n'arrive pas à prouver l'autre.
2. Il est facile de voir que $\mathcal{S}$ est correct pour $\mathcal{L}_1$, car la somme $|a_1|+|a_3|+|a_7|+|a_9|[2]$ est préservée par les règles d'inférences. Par exemple, pour la règle L_1 on a :
   $$|a_1| + |a_3| + |a_7| + |a_9| \equiv |\bar{a_1}| + |\bar{a_3}| + |a_7| + |a_9|[2].$$
   De même, les règles L_3, C_1, C_3, D_1, D_2 modifient exactement deux lettres parmi a_1, a_3, a_7, a_9 et les règles L_2 et C_2 n'en modifient aucune. On a donc, pour toutes formules f et f' et toute interprétation i, si $f \vdash_{\mathcal{S}} f'$ alors $i \models_1 f$ ssi $i \models_1 f'$, et donc $f \models_1 f'$, ce qui prouve que $\mathcal{S}$ est correct pour $\mathcal{L}_1$.
   Les théorèmes de $\mathcal{S}_{\{\mathbf{0}\}}$ sont les formules f telles que $\mathbf{0} \vdash_{\mathcal{S}} f$. Comme $0 \models_1 \mathbf{0}$, alors pour tous les théorèmes f de $\mathcal{S}_{\{\mathbf{0}\}}$ on a $0 \models_1 f$. Or on a $1 \models_1 \begin{smallmatrix}\bullet&\circ&\circ\\ \circ&\circ&\circ\\ \circ&\circ&\circ\end{smallmatrix}$, donc on a $\mathbf{0} \not\vdash_{\mathcal{S}} \begin{smallmatrix}\bullet&\circ&\circ\\ \circ&\circ&\circ\\ \circ&\circ&\circ\end{smallmatrix}$ et par conséquent $\mathcal{S}_{\{\mathbf{0}\}}$ est consistant.
3. Si $\mathcal{S}$ est fortement complet pour $\mathcal{L}_1$ alors pour toute formule f telle que $0 \models_1 f$ on a $\mathbf{0} \models_1 f$ donc $\mathbf{0} \vdash_{\mathcal{S}} f$.
   Comme précédemment on voit que $\mathcal{S}$ est correct pour $\mathcal{L}_2$ car toutes les règles d'inférences modifient exactement deux lettres parmi $a_1, a_2, a_4, a_6, a_8, a_9$ sauf D_2 qui n'en modifie aucune. On a bien sûr $0 \models_2 \mathbf{0}$, donc pour toute formule f telle que $\mathbf{0} \vdash_{\mathcal{S}} f$ on a $0 \models_2 f$. Comme $1 \models_2 \begin{smallmatrix}\circ&\bullet&\circ\\ \circ&\circ&\circ\\ \circ&\circ&\circ\end{smallmatrix}$, on a donc $\mathbf{0} \not\vdash_{\mathcal{S}} \begin{smallmatrix}\circ&\bullet&\circ\\ \circ&\circ&\circ\\ \circ&\circ&\circ\end{smallmatrix}$. Pourtant on a $0 \models_1 \begin{smallmatrix}\circ&\bullet&\circ\\ \circ&\circ&\circ\\ \circ&\circ&\circ\end{smallmatrix}$, ce qui prouve que $\mathcal{S}$ n'est pas fortement complet pour $\mathcal{L}_1$.

EXERCICE 34. Ces ensembles sont tous inclus dans $\mathrm{V}_\Sigma^\star$, nous commençons donc par prouver que cet ensemble est dénombrable. L'ensemble V_Σ est infini dénombrable, il suffit donc de prouver que $\mathbb{N}^\star$ est dénombrable. Pour cela, il suffit de trouver une famille d'ensembles finis E_n pour $n \in \mathbb{N}$ telle que $\mathbb{N}^\star = \bigcup_{n\in\mathbb{N}} E_n$ (qu'on peut énumérer entièrement par énumérations successives finies de E_0, E_1,...). Soit E_n l'ensemble des mots d'entiers $< n$ et de longueur $\leq n$; il est évident que $|E_n| \leq (n+1)^n$. De plus, pour tout mot $m \in \mathbb{N}^\star$ non vide, soit l sa longueur et n le plus grand entier apparaissant dans m, on $m \in E_{\max(l,n+1)}$. Le mot vide est aussi élément des E_n, on a donc bien $\mathbb{N}^\star = \bigcup_{n\in\mathbb{N}} E_n$, ce qui prouve que nos ensembles sont au plus dénombrables.

Comme $\mathcal{V} \subseteq \mathrm{T}_\Sigma$ cet ensemble est infini. De même, l'ensemble des formules $\forall x \, \Box$ pour tout $x \in \mathcal{V}$ est infini et inclus dans F_Σ qui est donc infini.

A_Σ est fini ssi $\Sigma = \Sigma_0^{\mathrm{P}}$ est fini. En effet, si $\Sigma \neq \Sigma_0^{\mathrm{P}}$ alors il y a au moins un atome par terme.

EXERCICE 36. Par induction sur t. Si t est une variable $x \in \mathcal{V}$ alors $\mathrm{Var}(t) = \{x\}$, donc $[\![t]\!]^{\mathcal{I}}_{\theta} = \theta(x) = \theta'(x) = [\![t]\!]^{\mathcal{I}}_{\theta'}$. Si t est une constante $a \in \Sigma_0^{\mathrm{F}}$ alors $[\![t]\!]^{\mathcal{I}}_{\theta} = a^{\mathcal{I}} = [\![t]\!]^{\mathcal{I}}_{\theta'}$.

Soit $t = f(t_1, \ldots, t_n)$, supposons que c'est vrai pour les termes $t_1, \ldots, t_n$. Si $\theta =_{|\mathrm{Var}(t)} \theta'$ alors $\theta =_{|\mathrm{Var}(t_i)} \theta'$ pour tout $1 \leq i \leq n$ puisque $\mathrm{Var}(t_i) \subseteq \mathrm{Var}(t)$, donc par hypothèse d'induction $[\![t_i]\!]^{\mathcal{I}}_{\theta} = [\![t_i]\!]^{\mathcal{I}}_{\theta'}$, donc

$$[\![t]\!]^{\mathcal{I}}_{\theta} \;=\; f^{\mathcal{I}}([\![t_1]\!]^{\mathcal{I}}_{\theta}, \ldots, [\![t_n]\!]^{\mathcal{I}}_{\theta}) \;=\; f^{\mathcal{I}}([\![t_1]\!]^{\mathcal{I}}_{\theta'}, \ldots, [\![t_n]\!]^{\mathcal{I}}_{\theta'}) \;=\; [\![t]\!]^{\mathcal{I}}_{\theta'}.$$

EXERCICE 38. Cette fonction n'est pas injective : avec $\Sigma = \Sigma_0^{\mathrm{P}} = \{p\}$, soient $\mathcal{I}$ (resp. $\mathcal{J}$) de domaine $\{0\}$ (resp. $\{0, 1\}$) telle que $p^{\mathcal{I}} = \mathrm{V}$ (resp. $p^{\mathcal{J}} = \mathrm{V}$). On a $\overline{\mathrm{A}}_\Sigma = \{p\}$ et $[\![p]\!]^{\mathcal{I}} = [\![p]\!]^{\mathcal{J}}$, donc $[\![.]\!]^{\mathcal{I}} = [\![.]\!]^{\mathcal{J}}$, mais $\mathcal{I} \neq \mathcal{J}$.

EXERCICE 39. Par induction sur φ.

— Si $\varphi \in \{\blacksquare, \Box\}$ alors le résultat est évident puisque ni $\mathcal{I} \models_1 \varphi$ ni $[\![.]\!]^{\mathcal{I}} \models_0 \varphi$ ne dépendent de $\mathcal{I}$.

— Si $\varphi \in \overline{\mathrm{A}}_\Sigma$, on a $\mathcal{I} \models_1 \varphi$ ssi $[\![\varphi]\!]^{\mathcal{I}} = \mathrm{V}$ ssi $[\![.]\!]^{\mathcal{I}} \models_0 \varphi$.

— Si φ est de la forme $\neg\varphi_1$ alors $\mathcal{I} \models_1 \varphi$ ssi $\mathcal{I} \not\models_1 \varphi_1$ ssi $[\![.]\!]^{\mathcal{I}} \not\models_0 \varphi_1$ (par hypothèse d'induction) ssi $[\![.]\!]^{\mathcal{I}} \models_0 \varphi$.

— Si φ est de la forme $\varphi_1 \wedge \varphi_2$, alors

$$\begin{array}{rll} \mathcal{I} \models_1 \varphi & \text{ssi} & \mathcal{I} \models_1 \varphi_1 \text{ et } \mathcal{I} \models_1 \varphi_2 \\ & \text{ssi} & [\![.]\!]^{\mathcal{I}} \models_0 \varphi_1 \text{ et } [\![.]\!]^{\mathcal{I}} \models_0 \varphi_2 \text{ (par h.i.)} \\ & \text{ssi} & [\![.]\!]^{\mathcal{I}} \models_0 \varphi. \end{array}$$

— On procède de même pour $\vee$, $\Rightarrow$ et $\Leftrightarrow$.

EXERCICE 44. La fonction est injective : soient $I, J \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$, si $\mathcal{H}_I = \mathcal{H}_J$ alors $[\![.]\!]^{\mathcal{H}_I} = [\![.]\!]^{\mathcal{H}_J}$ et donc $I = J$ d'après le Lemme 3.41.

Elle est également surjective : pour tout $\mathcal{H} \in \mathrm{H}_\Sigma$, soit $I = [\![.]\!]^{\mathcal{H}}$, par définition de H_Σ les interprétations $\mathcal{H}$ et $\mathcal{H}_I$ ont même domaine $\overline{\mathrm{T}}_\Sigma$ et interprètent de la même façon les symboles de Σ^{F}. De plus, par le Lemme 3.41 on a $[\![.]\!]^{\mathcal{H}} = [\![.]\!]^{\mathcal{H}_I}$, donc pour

tous $n \in \mathbb{N}$, $p \in \Sigma_n^{\mathrm{P}}$ et $t_1, \ldots, t_n \in \overline{\mathrm{T}}_\Sigma$ on a

$$\begin{aligned} p^{\mathcal{H}}(t_1, \ldots, t_n) &= p^{\mathcal{H}}([\![t_1]\!]^{\mathcal{H}}, \ldots, [\![t_n]\!]^{\mathcal{H}}) \\ &= [\![p(t_1, \ldots, t_n)]\!]^{\mathcal{H}} \\ &= [\![p(t_1, \ldots, t_n)]\!]^{\mathcal{H}_I} \\ &= p^{\mathcal{H}_I}([\![t_1]\!]^{\mathcal{H}_I}, \ldots, [\![t_n]\!]^{\mathcal{H}_I}) \\ &= p^{\mathcal{H}_I}(t_1, \ldots, t_n), \end{aligned}$$

donc $\mathcal{H}$ et $\mathcal{H}_I$ interprètent de la même façon les symboles de Σ^{P}, ce qui prouve que $\mathcal{H} = \mathcal{H}_I$ et donc que I est l'antécédent de $\mathcal{H}$. Il y a donc une correspondance exacte entre $\mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ et H_Σ.

EXERCICE 45. D'après le Théorème de Herbrand, une formule universelle $\forall^\star \varphi$ (où $\varphi \in \mathrm{M}_\Sigma$) est insatisfaisable ssi il existe un sous-ensemble fini insatisfaisable de $\mathrm{IF}(\varphi)$. D'après l'exercice 34, il est facile de voir que l'ensemble $\overline{\mathrm{T}}_\Sigma^n$ est dénombrable pour tout $n \in \mathbb{N}$ et donc que $\mathrm{IF}(\varphi)$ est dénombrable (si n est le nombre de variables de φ il y a par définition de $\mathrm{IF}(\varphi)$ une surjection de $\mathrm{IF}(\varphi)$ dans $\overline{\mathrm{T}}_\Sigma^n$). On peut en déduire que l'ensemble $\mathrm{IF}(\varphi)^\star$ est dénombrable, et donc l'ensemble des parties finies de $\mathrm{IF}(\varphi)$ également : on peut donc l'exprimer sous la forme $\{F_i \mid i \in \mathbb{N}\}$. Enfin, l'insatisfaisabilité de chaque F_i est décidable puisque c'est un ensemble fini de formules propositionnelles.

On peut donc tester l'insatisfaisabilité des ensembles F_i pour $i = 0, 1, 2 \ldots$ et on s'arrête dès qu'on en trouve un, ce qui garantit l'insatisfaisabilité de $\forall^\star \varphi$. De plus, si $\forall^\star \varphi$ est insatisfaisable, alors il existe $F \subseteq \mathrm{IF}(\varphi)$ fini et insatisfaisable, donc il existe $i \in \mathbb{N}$ tel que $F = F_i$, donc cette procédure termine en détectant l'insatisfaisabilité.

On remarque que cette procédure ne détecte jamais la satisfaisabilité et ne termine jamais sur une formule satisfaisable ; ce n'est pas très satisfaisant d'un point de vue pratique.

EXERCICE 53. Soit ψ' une skolemisée de ψ. Si ψ' est satisfaisable alors ψ également puisque $\psi' \models_1 \psi$ d'après le Théorème 3.65. Réciproquement, si ψ est satisfaisable alors il existe une interprétation $\mathcal{I}$ telle que $\mathcal{I} \models_1 \psi$, mais $\psi \models_1^\Sigma \psi'$ d'après le Théorème 3.65, donc il existe une interprétation $\mathcal{J}$ (extension de $\mathcal{I}$) telle que $\mathcal{J} \models_1 \psi'$, et donc ψ' est satisfaisable.

Logique du premier ordre : abrégé des terminologies et notations

$\boldsymbol{\Sigma^{\mathrm{F}}_n}$ ensemble des *symboles de fonction* $f \in \Sigma^{\mathrm{F}}$ tels que l'*arité* de f est n. $\boldsymbol{\Sigma^{\mathrm{P}}_n}$ ensemble des *symboles de prédicat* $p \in \Sigma^{\mathrm{P}}$ tels que l'arité de p est n. *Signature du premier ordre* $\boldsymbol{\Sigma} = \Sigma^{\mathrm{F}} \uplus \Sigma^{\mathrm{P}}$.

$\mathbf{T_\Sigma}$ ensemble des *termes* $t = f(t_1, \ldots, t_n)$, $\overline{\mathbf{T}}_{\boldsymbol{\Sigma}}$ *termes fermés.*

$\mathbf{A_\Sigma}$ ensemble des *atomes ou formules atomiques* $\alpha = p(t_1, \ldots, t_n)$. $\overline{\mathbf{A}}_{\boldsymbol{\Sigma}}$ *atomes fermés.*

$\mathbf{Var(}\boldsymbol{t}\mathbf{)}$, $\mathbf{Var(}\boldsymbol{\alpha}\mathbf{)}$ ensemble des variables de t, α.

$\mathbf{F_\Sigma}$ ensemble des *formules du premier ordre* φ. $\mathbf{M_\Sigma}$ ensemble des *matrices* (= formules sans quantificateurs). On a $\mathrm{M}_\Sigma = \mathrm{P}(\mathrm{A}_\Sigma)$ (formules propositionnelles sur l'ensemble de symboles propositionnels A_Σ), donc $\overline{\mathrm{M}}_\Sigma = \mathrm{P}(\overline{\mathrm{A}}_\Sigma)$.

$\mathbf{I_\Sigma}$ ensemble des *interprétations ou modèles du premier ordre* $\mathcal{I}$. $\boldsymbol{f^{\mathcal{I}}}$ fonction de $\mathcal{I}^n$ dans (le domaine de) $\mathcal{I}$, $\boldsymbol{p^{\mathcal{I}}}$ fonction de $\mathcal{I}^n$ dans $\{\mathrm{V}, \mathrm{F}\}$.

$\boldsymbol{\mathcal{I}^{\mathcal{V}}}$ ensemble des *valuations* θ dans $\mathcal{I}$, i.e., des fonctions de $\mathcal{V}$ dans $\mathcal{I}$.

$\boldsymbol{[\![t]\!]^{\mathcal{I}}_{\theta}}$ *valeur de* t *dans* $\mathcal{I}, \theta$; $\boldsymbol{[\![t]\!]^{\mathcal{I}}}$ si t fermé.

L'*algèbre des termes* $\boldsymbol{\mathcal{T}_\Sigma} \in \mathrm{I}_{\Sigma^{\mathrm{F}}}$ a pour domaine T_Σ et vérifie $f^{\mathcal{T}_\Sigma}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$. La sous-algèbre des termes fermés est $\overline{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\Sigma}}$. Pour tout terme fermé $t \in \overline{\mathrm{T}}_\Sigma$, on a $[\![t]\!]^{\overline{\mathcal{T}}_\Sigma} = t$.

Une valuation $\sigma \in \mathrm{T}_\Sigma^{\mathcal{V}}$ est une *substitution*, $\boldsymbol{t\sigma} = [\![t]\!]^{\mathcal{T}_\Sigma}_{\sigma}$ est une *instance de* t. Le *domaine* de σ est $\mathbf{Dom(}\boldsymbol{\sigma}\mathbf{)} = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$. $\sigma \in \overline{\mathcal{T}}_\Sigma^{\mathcal{V}}$ est une *substitution fermée* et alors $t\sigma$ une *instance fermée de* t.

Instance d'atomes : $p(t_1, \ldots, t_n)\sigma = p(t_1\sigma, \ldots, t_n\sigma)$.

$\boldsymbol{[\![\alpha]\!]^{\mathcal{I}}_{\theta}} \in \{\mathrm{V}, \mathrm{F}\}$ est la *valeur de* α *dans* $\mathcal{I}, \theta$; noté $\boldsymbol{[\![\alpha]\!]^{\mathcal{I}}}$ si $\alpha \in \overline{\mathrm{A}}_\Sigma$.

$\boldsymbol{[\![.]\!]^{\mathcal{I}}}$ est la fonction qui à $\alpha \in \overline{\mathrm{A}}_\Sigma$ associe $[\![\alpha]\!]^{\mathcal{I}}$; c'est une interprétation propositionnelle sur l'ensemble de symboles propositionnels $\overline{\mathrm{A}}_\Sigma$, donc $[\![.]\!]^{\mathcal{I}} \in \mathrm{I}_{\mathrm{p}}(\overline{\mathrm{A}}_\Sigma)$.

Pour $x \in \mathcal{V}$ et $u \in \mathcal{I}$, on note $\boldsymbol{\theta[x \mapsto u]}$ la valuation θ' telle que $\theta'(x) = u$ et $\forall y \in \mathcal{V}$, si $y \neq x$ alors $\theta'(y) = \theta(y)$. On a $\boldsymbol{\mathcal{I}, \theta \models_1 \alpha}$ ssi $[\![\alpha]\!]^{\mathcal{I}}_{\theta} = \mathrm{V}$, et $\mathcal{I}, \theta \models_1 \forall x\, \varphi$ ssi pour tout $u \in \mathcal{I}$ on a $\mathcal{I}, \theta[x \mapsto u] \models_1 \varphi$.

$\mathbf{VL(}\boldsymbol{\varphi}\mathbf{)}$ ensemble des *variables libres* de φ. $\overline{\mathbf{F}}_{\boldsymbol{\Sigma}} = \{\varphi \in \mathrm{F}_\Sigma \mid \mathrm{VL}(\varphi) = \emptyset\}$ ensemble des *formules fermées*. Pour $\varphi \in \overline{\mathrm{F}}_\Sigma$ on a $\mathcal{I}, \theta \models_1 \varphi$ ssi $\mathcal{I}, \theta' \models_1 \varphi$, on note donc $\boldsymbol{\mathcal{I} \models_1 \varphi}$ si $\mathcal{I}, \theta \models_1 \varphi$ pour une valuation $\theta \in \mathcal{I}^{\mathcal{V}}$ quelconque.

Si aucune des variables de $\mathbf{Var(}\boldsymbol{\sigma}\mathbf{)} = \{\mathrm{Var}(\sigma(x)) \mid x \in \mathrm{Dom}(\sigma)\}$ n'est liée dans φ, on peut définir $\boldsymbol{\varphi\sigma}$ *instance de* φ.

On définit $\boldsymbol{\sigma\theta} \in \mathcal{I}^{\mathcal{V}}$ par $\sigma\theta(x) = [\![\sigma(x)]\!]^{\mathcal{I}}_{\theta}$ pour tout $x \in \mathcal{V}$ (si θ est une substitution et $\mathcal{I} = \mathcal{T}_\Sigma$, alors pour tout t on a $t(\sigma\theta) = (t\sigma)\theta$, noté $\boldsymbol{t\sigma\theta}$). On a $\mathcal{I}, \theta \models_1 \varphi\sigma$ ssi $\mathcal{I}, \sigma\theta \models_1 \varphi$.

Pour $\Sigma \subseteq \Sigma'$ et $\mathcal{I} \in \mathrm{I}_\Sigma$, $\mathcal{I}' \in \mathrm{I}_{\Sigma'}$ est une *extension de* $\mathcal{I}$ *à* Σ' si $\mathcal{I}'$ est de mÃªme domaine que $\mathcal{I}$ et $\forall s \in \Sigma, s^{\mathcal{I}} = s^{\mathcal{I}'}$. On note $\boldsymbol{\mathcal{I}'|_\Sigma = \mathcal{I}}$ la *restriction de* $\mathcal{I}'$ *à* Σ.

$\mathbf{H_\Sigma}$ ensemble des *modèles* $\mathcal{H}$ *de Herbrand*, i.e., tels que $\mathcal{H}|_{\Sigma^{\mathrm{F}}} = \overline{\mathcal{T}}_{\Sigma^{\mathrm{F}}}$. À $I \in \mathrm{I_p}(\overline{\mathrm{A}}_\Sigma)$ on associe le modèle de Herbrand $\boldsymbol{\mathcal{H}_I}$ tel que $\forall p \in \Sigma_0^{\mathrm{P}}$, $p^{\mathcal{H}_I} = I(p)$, et $\forall n > 0, \forall p \in \Sigma_n^{\mathrm{P}}$, $p^{\mathcal{H}_I}(t_1 \ldots, t_n) = I(p(t_1, \ldots, t_n))$. On a $[\![.]\!]^{\mathcal{H}_I} = I$.

Soit $\varphi \in \mathrm{M}_\Sigma$, on note $\boldsymbol{\forall^\star \varphi}$ toute formule de la forme $\forall x_1 \cdots \forall x_n \varphi$, où $\{x_1, \ldots, x_n\} = \mathrm{VL}(\varphi)$; $\forall^\star \varphi$ est une *formule universelle*. $\mathbf{IF(\varphi) = IF(\forall^\star \varphi)} = \left\{\varphi\sigma \,|\, \sigma \in \overline{\mathcal{T}}_\Sigma^{\mathcal{V}}\right\}$ ensemble des instances fermées.

On note $\boldsymbol{\varphi \models_1 \psi}$ lorsque $\forall \mathcal{I} \in \mathrm{I}_\Sigma$, $\forall \theta \in \mathcal{I}^{\mathcal{V}}$, si $\mathcal{I}, \theta \models_1 \varphi$ alors $\mathcal{I}, \theta \models_1 \psi$. On note $\boldsymbol{\varphi \equiv_1 \psi}$ si $\varphi \models_1 \psi$ et $\psi \models_1 \varphi$.

φ est *normalisée* si pour tout x qui a une occurrence liée dans φ, il n'existe qu'une sous-formule de φ de préfixe $\forall x$ ou $\exists x$, et $x \notin \mathrm{VL}(\varphi)$.

On appelle *skolemisée de* $\psi \in \overline{\mathrm{F}}_\Sigma$ (sous forme normale négative et normalisée) *en* φ toute formule ψ' obtenue de ψ en y remplaÃ§ant la sous-formule $\exists x\, \varphi$ par $\varphi[f(x_1, \ldots, x_n)/x]$ avec $f \notin \Sigma$. On note $\boldsymbol{\Sigma + f}$ toute signature contenant Σ telle que $\psi' \in \overline{\mathrm{F}}_{\Sigma+f}$. On appelle *skolemisée de* ψ toute formule sans quantificateur existentiel obtenue par skolemisations successives.

Pour $\Sigma \subseteq \Sigma'$, les modèles de $E \subseteq \overline{\mathrm{F}}_\Sigma$ sont *extensibles à* $E' \subseteq \overline{\mathrm{F}}_{\Sigma'}$, noté $\boldsymbol{E \models_1^\Sigma E'}$, si $\forall \mathcal{I} \in \mathrm{I}_\Sigma$ tq $\mathcal{I} \models_1 E$, il existe une extension $\mathcal{I}'$ de $\mathcal{I}$ à Σ' tq $\mathcal{I}' \models_1 E'$.

On note $\mathbf{u_\Sigma(\psi)}$ une formule universelle telle que $\mathrm{u}_\Sigma(\psi) \models_1 \psi$ et $\psi \models_1^\Sigma \mathrm{u}_\Sigma(\psi)$, et $\boldsymbol{\Sigma_\psi}$ une signature contenant Σ et telle que $\mathrm{u}_\Sigma(\psi) \in \overline{\mathrm{F}}_{\Sigma_\psi}$. Pour $E \subseteq \overline{\mathrm{F}}_\Sigma$, on note $\mathbf{u_\Sigma(E)}$ un ensemble de formules universelles tel que $\mathrm{u}_\Sigma(E) \models_1 E$ et $E \models_1^\Sigma \mathrm{u}_\Sigma(E)$, et $\boldsymbol{\Sigma_E}$ une signature contenant Σ et telle que $\mathrm{u}_\Sigma(E) \subseteq \overline{\mathrm{F}}_{\Sigma_E}$.

Un *problème d'unification* P est un ensemble fini d'équations $t \simeq t'$. $\mathbf{Unif(P)}$ est l'ensemble des *unificateurs de* P, i.e., des σ tels que $t\sigma = t'\sigma$ pour tout $t \simeq t'$ dans P. Une variable x est *résolue dans* P s'il y a une équation $x \simeq t$ dans P et x n'a qu'une occurrence dans P. Le problème P est *sous forme résolue* s'il est de la forme $\{x_1 \simeq t_1, \ldots, x_n \simeq t_n\}$ et les x_i sont toutes résolues dans P ; on note alors $\boldsymbol{\sigma_P} = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$; on a $\sigma_P \in \mathrm{Unif}(P)$.

σ est *plus générale que* σ' s'il existe τ tel que $\sigma' = \sigma\tau$. $\sigma \in \mathrm{Unif}(P)$ est un *unificateur plus général de* P si σ est plus général que tout $\sigma' \in \mathrm{Unif}(P)$. On note $\mathbf{unif(P)}$ un unificateur plus général de P.

Une *clause* C est une clause propositionnelle dans $\mathrm{P}(\mathrm{A}_\Sigma)$. Une *forme clausale* est un ensemble fini F de clauses et $\boldsymbol{\forall^\star F} = \{\forall^\star C \,|\, C \in F\}$.

$$\textit{résolution}\frac{D_1 \dot\vee L_1 \quad D_2 \dot\vee L_2}{(D_1 \vee D_2\pi)\mathrm{unif}(L_1, \overline{L_2\pi})} \qquad \textit{factorisation}\frac{C' \dot\vee L_1 \dot\vee \cdots \dot\vee L_m}{(C' \vee L_1)\mathrm{unif}(L_1, \ldots, L_m)}$$

où $\pi \in \mathrm{Sym}(\mathcal{V})$ vérifie $\mathrm{Var}(D_1 \vee L_1) \cap \mathrm{Var}(D_2\pi \vee L_2\pi) = \emptyset$.

Index