NOTES SUR LA LOGIQUE

PREPRINT

Dominique Méry [*]
LORIA
Université de Lorraine
dominique.mery@loria.fr
https://members.loria.fr/Mery
https://mery54.github.io/mery/

9 janvier 2025

ABSTRACT

Ces notes concernent une description du langage logique du calcul ds propositions et du calcul des prédicats du premier ordre. On explique les concepts principaux de cohérence, de complétude, de preuve.

[*] Supported by the ANR Project EBRP-EventB-Rodin-Plus (ANR-19-CE25-0010) and by the ANR Project DISCONT (ANR-17-CE25-0005)

TABLE DES MATIÈRES

CHAPITRE 1

CALCUL DES PROPOSITIONS

Sommaire

Le calcul des propositions est un premier cadre dans lequel nous allons définir les notions relatives à la syntaxe et à la sémantique comme les modèles, la consistance, la complétude, la déduction. .... Il parait à première vue rudimentaire mais néanmoins permet de bien poser les problèmes des systèmes formels.

1.1 Généralités syntaxiques

Nous donnons dans cette partie quelques notations très utiles par la suite. Nous définissons la notion de formules qui sont les termes manipulés dans ce système. On note $\mathcal{V}$ un ensemble infini dénombrable d'éléments appelés symboles de variables propositionnelles :

$\mathcal{V} = \{A, B, C, D, \ldots, A_i, B_i, \ldots, P, Q, R, S, \ldots\}$.

On note $\mathcal{C}$ un ensemble d'éléments appelés symboles de connection et de séparation :

$\mathcal{C} = \{\vee, \wedge, \sim, \Rightarrow, \equiv, (,), \ldots\}$

On suppose que $\mathcal{C} \cap \mathcal{P} = \varnothing$.

☼**Définition 1.1** Syntaxe des formules

Une formule est une suite finie de symboles de $\mathcal{C} \cup \mathcal{P}$ obtenue par application finie des règles suivantes :

1. tout symbole de variable propositionnelle est une formule dite atomique.
2. si φ est une formule, alors $\sim (\varphi)$ est une formule.
3. si φ et ψ sont deux formules, alors $(\varphi) \wedge (\psi)$, $(\varphi) \Rightarrow (\psi)$, $(\varphi) \vee (\psi)$, $(\varphi) \equiv (\psi)$ sont des formules.

Une formule sur $\mathcal{C} \cup \mathcal{P}$ est appelée une formule propositionnelle. Dans ce chapitre, une formule sera implicitement propositionnelle. Le parenthésage permet de rendre l'analyse non ambigue. Nous pourrions aussi utiliser une grammaire pour définir l'ensemble des formules.

Exercice 1 *Construire une grammaire définissant les formules ci-dessus.*

Une suite finie de formules $(\varphi_0, \varphi_1, \ldots, \varphi_n)$ est une construction si elle vérifie pour tout i de $\{0, \ldots, n\}$, l'une des conditions suivantes :

1. φ_i est un symbole de variable propositionnelle.
2. il existe $j < i$ tel que φ_i est $\sim (\varphi_j)$.
3. il existe j et k tels que $j < i$ et $k < i$ et φ_i est (φ_j) *op* (φ_k) où *op* $\in \{\wedge, \vee, \Rightarrow, \equiv\}$

La construction $(\varphi_0, \varphi_1 \ldots \varphi_n)$ est une construction de φ_n. Toute sous-suite $(\varphi_0, \varphi_1 \ldots \varphi_i)$ est une construction de φ_i, pour tout i de $\{0, \ldots, n\}$. Une formule φ admet une infinité dénombrable de constructions, il suffit d'ajouter des variables propositionnelles de $\mathcal{P}$ dans une construction de φ. Le nombre d'application des règles (2) ou (3) est l'ordre de complexité de la construction.

Exemple 1.1

1. $(P, (P) \wedge (P), P)$ est une construction de P d'ordre 1.
2. $(P, Q, R, (P) \wedge (Q), ((P) \wedge (Q)) \Rightarrow (R))$ est une construction de $((P) \wedge (Q)) \Rightarrow (R)$ d'ordre 2.
3. $(P, ((P) \wedge (P)) \Rightarrow (P), P)$ n'est pas une construction.

Exercice 2 *Quelle est la relation entre la notion d'ordre d'une construction et la hauteur de l'arbre associé à une formule donnée ?*

Une formule φ se décompose suivant les règles de construction. On définit parallèlement la notion d'arbre de décomposition d'une formule φ. Cette décomposition correspond à la construction d'un arbre syntaxique associé à φ en utilisant les règles de grammaire adéquates.

☼**Définition 1.2** Décomposition d'une formule φ

La décomposition d'une formule φ et la construction d'un arbre de décomposition associé obéissent aux règles suivantes :

1. si φ est un symbole de variable propositionnelle, alors φ ne se décompose pas et son arbre de décomposition est réduit à φ.
2. si φ est obtenue par la règle 2, alors φ se décompose en ψ où φ est $\sim\ (\psi)$ et son arbre de décomposition est :
   $\sim$
   |
   ψ
3. si φ est obtenue par la règle 3, alors φ se décompose en ψ_1 et ψ_2 où φ est (ψ_1) *op* (ψ_2) et son arbre de décomposition est :
   op

   ψ_1 ψ_2

On peut ajouter aux formules une formule de base qui est la constante false. Son statut est celui d'un connecteur d'arité 0. Nous le définirons ainsi $FAUX \stackrel{def}{=} \sim(P) \wedge (P)$. De même, nous pouvons définir une constante true comme suit : $VRAI \stackrel{def}{=} \sim(P) \vee (P)$. le propositioon P est quelconque. Cette définition sera justifiée par la suite.

Exercice 3 *Ecrire une procédure ou une fonction retournant l'arbre de décomposition d'une formule.*

On note $\mathcal{P}rop(\mathcal{P})$, l'ensemble des formules sur $\mathcal{P} \cup \mathcal{C}$.

1.2 Sémantique des propositions

Les formules de $\mathcal{P}rop(\mathcal{P})$ ont une existence purement syntaxique. Nous leur associons un sens permettant de signifier les connecteurs. On note $\mathcal{B} = \{0, 1\}$ et $\mathcal{B}_n$, les fonctions n-aires à valeur dans $\mathcal{B}$. Soit $f \in \mathcal{B}_n$ une table de vérité de f est une structure ayant n+1 colonnes et 2^n lignes.

Exemple 1.2

$$f \in B_1 : \begin{array}{cc} 0 & \epsilon_0 \\ 1 & \epsilon_1 \end{array}$$

$$f \in B_2 : \begin{array}{ccc} 0 & 0 & \epsilon_0 \\ 0 & 1 & \epsilon_1 \\ 1 & 0 & \epsilon_2 \\ 1 & 1 & \epsilon_3 \end{array}$$

A tout symbole de $\mathcal{C}$, on associe une fonction de $\bigcup_{i\geq 0} \mathcal{B}_i$ ou une table de vérité. Soit $\mathcal{B} = (\mathcal{B}, \bigcup_{i\geq 0} \mathcal{B}_i$. $\mathcal{B}$ est une algèbre. $\mathcal{B}$ est l'algèbre booléenne. Soit φ une formule de $\mathcal{P}rop(\mathcal{P})$. Une fonction booléenne $[\![\varphi]\!]$, dont l'arité est le nombre de variables figurant dans φ, peut être associée à φ. On dit que $[\![\varphi]\!]$ est la fonction de vérité de φ.
Donnons une justification à cette définition. Pour justifier cette définition, nous en donnons une version constructive.
$[\![\ ]\!]\ :\ \mathcal{P}rop(\mathcal{P}) \longrightarrow \bigcup_{i\geq 0} \mathcal{B}_i$

1. Pour toute variable propositionnelle P de $\mathcal{P}$, on associe la fonction notée $[\![P]\!]$ et définie par : $[\![P]\!](\varepsilon) = \varepsilon$, pour tout $\varepsilon \in \mathcal{B}$.

2. Si la formule φ est de la forme $\sim(\psi)$, alors $[\![\varphi]\!] = \overline{[\![\psi]\!]}$.
3. Si la formule φ est de la forme $(\varphi_1)\ op\ (\varphi_2)$, alors $[\![\varphi]\!] = [\![\varphi_1]\!]\ fb(op)\ [\![\varphi_2]\!]$ où $fb(op)$ est la fonction booléenne associée à *op*.

Une formule φ a donc une sémantique complètement définie.
Une formule φ est une tautologie, si $[\![\varphi]\!]$ est la fonction booléenne uniformément égale à 1. Une formule φ est une antilogie, si $[\![\varphi]\!]$ est la fonction booléenne uniformément égale à 0. Deux formules ayant la même table de vérité sont dites synonymes.

► **Théorème 1.1**

Pour toute fonction booléenne f de $\mathcal{B}^n$, il existe une formule φ écrite avec $\sim, \wedge, \vee$, à partir de n symboles de proposition et telle que $[\![\varphi]\!] = f$.

PREUVE:
1. $f = 0$:
$f = [\![(A_1 \wedge \sim A_1)) \vee \ldots (A_n \wedge \sim A_n)]\!]$
2. $f \neq 0$:
Soit $(\varepsilon_1, \ldots, \varepsilon_n)$ telle que $f(\varepsilon_1, \ldots, \varepsilon_n) = 1$. On associe à $(\varepsilon_1, \ldots, \varepsilon_n)$ la formule suivante :
$\alpha_1 \wedge \ldots \wedge \alpha_n$ où $\alpha_i = \sim A_i$ si $\varepsilon_i = 0$ et $\alpha_i = A_i$, si $\varepsilon_i = 1$
Alors $f = [\![\bigvee_{j=\{1,\ldots,2^n\}} \bigwedge_{i\in\{1,n\}} \alpha_i^j]\!]$. On suppose que les termes non nuls figurent dans la sommation.

□

Corollaire 1 *Toute formule φ est synonyme d'une forme dite en forme normale disjonctive ou conjonctive.*

On appelle valuation ou réalisation, une application de $\mathcal{P}$ dans $\mathcal{B}$, qui associe à tout symbole de $\mathcal{P}$ une valeur 0 ou 1. On appelle valeur d'une formule φ pour une valuation δ donnée et on note $[\![\varphi]\!](\delta)$, la valeur de $[\![\varphi]\!]$ pour δ.

☼**Définition 1.3** modèle d'une formule propositionnelle

Une valuation δ est un modèle pour φ, si $[\![\varphi]\!](\delta) = 1$, que l'on notera aussi sous la forme "$\delta \models \varphi$".

Soit φ une formule. $\models \varphi$ dénotera le fait que φ est une tautologie c'est-à-dire que toutes les valuations possibles sont modèles de cette formule. Soient deux ensembles de formules Σ_1 et Σ_2 telles que $\Sigma_1 \subseteq \Sigma_2$. Toute valuation modèle des formules de Σ_2 est un modèle des formules de Σ_1.

1.3 Déduction sémantique

On dit que φ se déduit sémantiquement de Σ, un ensemble de formules, si tout modèle de Σ est un modèle de φ. On notera [1] $\Sigma \models \varphi$.

☺ **Propriété 1.1**

1. $\varnothing \models \varphi$ ssi φ est une tautologie
2. Si $\Sigma_1 \subseteq \Sigma_2$ et $\Sigma_2 \models \varphi$, alors $\Sigma_1 \models \varphi$
3. $\Sigma \cup \{\varphi\} \models \psi$ ssi $\Sigma \models \varphi \Rightarrow \psi$
4. $\Sigma \models \varphi$ ssi $\Sigma \cup \{\sim \varphi\}$ n'a pas de modèle (on dit aussi contradictoire).

---

1. J. L. KRIVINE a critiqué cette notation qui ne lui semblait pas adaptée. L'objectif est de montrer le lien entre les deux types de déductions

$\models$ est le symbole de déduction sémantique. Nous allons montrer que la déduction a deux formes : une forme sémantique et une forme syntaxique. Nous réécrivons les propriétés précédentes. Soient φ et ψ deux formules. On dit que ψ se déduit sémantiquement de φ, et on note $\{\varphi\} \models \psi$, si tout modèle de φ est un modèle de ψ.

Proposition 1.1

1. $\{\varphi, \varphi \Rightarrow \psi\} \models \psi$
2. $\{\varphi\} \models \psi$ ssi $\models \varphi \Rightarrow \psi$ (théorème de la déduction)
3. $\Sigma \cup \{\varphi\} \models \psi$ ssi $\Sigma \models \varphi \Rightarrow \psi$

"$\models \varphi \Rightarrow \psi$" exprime le fait que $\varphi \Rightarrow \psi$ est une tautologie, alors que $\models$ est une relation de déduction. Les preuves sont simples.

Exercice 4 *Montrer la proposition précédente.*

Soit un ensemble de formules . Σ est dit cohérent ou consistant, s'il existe un modèle pour Σ.

☺ **Propriété 1.2**

1. Soit $\Sigma_1 \subseteq \Sigma_2$. Si Σ_1 est inconsistant, alors Σ_2 est inconsistant.
2. Soit $\Sigma_1 \subseteq \Sigma_2$. Si Σ_2 est consistant, alors Σ_1 est consistant.
3. Σ est inconsistant ssi, pour toute formule φ de $\mathcal{P}rop(\mathcal{P})$, $\Sigma \models \varphi$
4. Σ est inconsistant ssi, il existe une antilogie φ, telle que $\Sigma \models \varphi$
5. $\{\varphi\}$ est inconsistant ssi φ est une antilogie.
6. $\Sigma \models \sim \varphi$ ssi $\Sigma \cup \{\varphi\}$ est inconsistant.
7. $\Sigma \models \varphi$ ssi $\Sigma \cup \{\sim \varphi\}$ est inconsistant.

Le théorème que nous abordons maintenant est un outil précis pour prouver d'autres résultats très utiles. Sa démonstration peut se faire à l'aide de propriétés topologiques.

► **Théorème 1.2** Théorème de finitude
Soit Σ un ensemble de formules. Σ est consistante ssi toutes les parties finies de Σ sont consistantes.

PREUVE:
1. Soit Σ consistante. Il existe δ un modèle de Σ. Soit $\Sigma_1 \subseteq \Sigma$, Σ_1 finie. Alors δ est un modèle de Σ_1.
2. Soit Σ tel que toute partie finie de Σ est consistante. On suppose que les variables propositionnelles sont rangées en ordre : $P_0, P_1, \ldots, P_i \ldots$. On construit une valuation satisfaisant Σ. Si Σ contient un nombre fini de formules, la preuve est terminée.
Supposons que Σ est infinie. Une suite $(\varepsilon_0\varepsilon_1 \ldots \varepsilon_i)$ est une bonne suite, si, pour toute partie finie Σ_0 de Σ, il existe un modèle δ de Σ_0 tel que $\delta(P_j) = \varepsilon_j$, pour tout $j \in \{0, \ldots, i\}$. On construit une suite infinie $\varepsilon_0 \ldots \varepsilon_i$ telle que toute sous-suite $\varepsilon_0 \ldots \varepsilon_i$ soit bonne. Ainsi, on va recouvrir Σ.
Construction de la suite

Cas 1
[Cas de base]

(0) est une bonne suite : pour toute partie finie de Σ, il existe une valuation δ modèle de cette partie finie telle que $\delta(P_0) = 0$.

(0) n'est pas une bonne suite : il existe une partie finie $\Sigma_0 \subseteq \Sigma$, telle que pour toute valuation modèle de Σ_0, $\delta(P_0) = 1$. Dans ce cas, (1) est une bonne suite car, si Σ_1 est une partie finie de Σ, alors $\Sigma_1 \cup \Sigma_0$ est une partie finie qui admet par hypothèse un modèle δ_0. Dans ce cas, δ_0 vérifie Σ_0 et de plus, $\delta_0(P_0) = 1$.

◇

Cas 2
Soit $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_k)$ une bonne suite. Deux cas se présentent :

$(\varepsilon_0, \ldots, \varepsilon_k, 0)$ est une bonne suite.
$(\varepsilon_0, \ldots, \varepsilon_k, 0)$ n'est pas une bonne suite.

Supposons que $(\varepsilon_0, \ldots, \varepsilon_k, 0)$ n'est pas une bonne suite.
Dans ce cas, il existe $\Sigma_0 \subseteq \Sigma$ partie finie telle que pour toute valuation δ satisfaisant Σ_0, $\delta(P_{k+1}) = 1$.
$(\varepsilon_0, \ldots, \varepsilon_k, 1)$ est une bonne suite car
soit Σ_1 une partie quelconque finie de Σ. Dans ce cas, $\Sigma_1 \cup \Sigma_0$ est aussi finie. Il existe δ un modèle de $\Sigma_1 \cup \Sigma_0$ qui est un modèle de Σ_0. Dans ce cas, $\delta(P_{k+1}) = 1$.
Nous avons construit une suite infinie $(\varepsilon_0 \ldots \varepsilon_i \ldots)$ telle que les sous-suites $(\varepsilon_0 \ldots \varepsilon_i)$ sont toutes de bonnes suites. Cette suite infinie définit une valuation δ, modèle de Σ.
Soit φ une formule de Σ. Supposons que P_n est la variable de plus grand numéro figurant dans φ. Dans ce cas, $(\varepsilon_0 \ldots \varepsilon_n)$ est une bonne suite et vérifie φ par construction. $\diamond$

□

Un corollaire à cette proposition permet de caractériser la déduction sémantique.

Corollaire 2 *Soit φ une formule et $\Sigma \subseteq \mathcal{P}rop(\mathcal{P})$.*
$\Sigma \models \varphi$ ssi il existe $\Sigma_0 \subseteq \Sigma$ une partie finie telle que $\Sigma_0 \models \varphi$

Preuve

1. Soient Σ et φ tel que $\Sigma \models \{\varphi\}$. $\Sigma \cup \{\sim \varphi\}$ est inconsistante ssi il existe une partie finie de Σ, Σ_0, telle que
$\Sigma_0 \cup \{\sim \varphi\}$ est inconsistante ssi $\Sigma_0 \models \varphi$.
2. Soit $\Sigma_0 \subseteq \Sigma$ telle que $\Sigma_0 \models \varphi$. Alors une valuation modèle de Σ est un modèle de Σ_0, donc $\Sigma_0 \models \varphi$.

fin de la preuve

1.4 Déduction formelle

Le précédent paragrpha a présenté la relation de déduction *sémantique* et il reste à envisager le point de vue *syntaxique* ou *formel*.

Soient les schémas d'axiomes suivants :

(A_1) $\alpha \Rightarrow (\beta \Rightarrow \alpha)$
(A_2) $(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$
(A_3) $(\sim \alpha \Rightarrow \sim \beta) \Rightarrow ((\sim \alpha \Rightarrow \beta) \Rightarrow \alpha)$

Soit la règle suivante :

$$\frac{\alpha, \alpha \Rightarrow \beta}{\beta}$$

appelée détachement ou modus ponens.

Nous utilisons ces axiomes et cette règle dans le cadre de démonstration formelle. Une démonstration est une suite finie de formules

$$\begin{array}{ll} 1 & \varphi_1 \\ 2 & \varphi_2 \\ & \vdots \\ n & \varphi_n \end{array}$$

telle que φ_i est une instance d'axiome ou bien φ_i est obtenue par application de la règle de détachement avec φ_j et φ_k tel que $j, k < i$ et φ_k est $\varphi_j \Rightarrow \varphi_i$.
Une démonstration sous hypothèse $\mathcal{H}$ est une suite finie de formules telle que φ_i est une instance d'axiomes ou bien φ_i est une formule de $\mathcal{H}$ ou bien φ_i est obtenue par application de la règle de détachement avec φ_j

et φ_l tels que j, k<i et φ_k est $\varphi_j \Rightarrow \varphi_i$.

Une formule φ est un thèorème, s'il existe une démonstration contenant φ : $\vdash \varphi$.
Une formule φ est un théorème sous hypothèse $\mathcal{H}$ s'il existe une démonstration sous hypothèse $\mathcal{H}$, contenant φ : on notera $\mathcal{H} \vdash \varphi$.
Nous pouvons aussi noter $\mathcal{H} \longrightarrow \varphi$. Cette notation $\longrightarrow$ se retrouve dans le calcul des séquents de GENTZEN [3] et montre, dans ce cas, l'apparente équivalence de $\vdash$ et $\models$. Cette équivalence fera l'objet d'un théorème.

Exemple 1.3

1. $(\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha)) \Rightarrow ((\alpha \Rightarrow (\alpha \Rightarrow \alpha)) \Rightarrow (\alpha \Rightarrow \alpha))$
2. $(\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha))$
3. $((\alpha \Rightarrow (\alpha \Rightarrow \alpha)) \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$
4. $\alpha \Rightarrow (\alpha \Rightarrow \alpha)$
5. $\alpha \Rightarrow \alpha$.

$\vdash \alpha \Rightarrow \alpha$

Il y a une relation entre l'implication et la déduction, de même nature que la déduction sémantique.

► **Théorème 1.3** Théorème de la déduction
$\{\varphi\} \vdash \psi$ ssi $\vdash \varphi \Rightarrow \psi$.

PREUVE:
1. Supposons que $\vdash \varphi \Rightarrow \psi$. Il existe une démonstration à partir d'aucune hypothèse : φ
   (1) φ_1
   (2) φ_2
   ...
   (n) φ_n
   (n+1) $\varphi \Rightarrow \psi$
   (n+2) φ
   (n+3) ψ

   Si nous ajoutons une hypothèse φ, $\varphi \Rightarrow \psi$ reste un théorème. La partie interne des points est la démonstration de $\varphi \Rightarrow \psi$. On complète par φ, puisque c'est une hypothèse. Par application du modus ponens, on déduit ψ. Donc $\{\varphi\} \vdash \psi$.
2. Supposons que $\{\varphi\} \vdash \psi$.
   Nous utilisons une démonstration par récurrence sur la longueur des démonstrations.

Cas 1
$\varphi \vdash \psi$ est de longueur 1.

Sous-Cas 1. 1
ψ est φ
Dans ce cas, l'exemple démontre $\vdash \varphi \Rightarrow \varphi$.

Sous-Cas 1. 2
ψ est une instance d'axiome logique. Dans ce cas, ψ ne dépend pas de φ.

a. ψ
b. $\psi \Rightarrow (\varphi \Rightarrow \psi)$ (axiome)
c. $\varphi \Rightarrow \psi$ DETACHEMENT sur 2 et 3.
On en déduit $\vdash \varphi \Rightarrow \psi$.

◇

Cas 2
Si $\{\varphi\} \vdash \psi$, alors $\vdash \varphi \Rightarrow \psi$, pour toute démonstration de longueur $<$ n.

Soit $\{\varphi\} \vdash \psi$ un théorème dont la démonstration est de longueur n. φ
(1) φ_1
$\vdots$(n) φ_n et φ_n est ψ

Sous-Cas 2. 1
ψ est une hypothèse. ψ est φ. Or, $\vdash \varphi \Rightarrow \varphi$, par l'exemple. $\diamond$

Sous-Cas 2. 2
ψ est une instance d'axiome.
On construit une démonstration :
(1) ψ
(2) $\psi \Rightarrow (\varphi \Rightarrow \psi)$
(3) $\varphi \Rightarrow \psi$

Donc, $\vdash \varphi \Rightarrow \psi$.

Sous-Cas 2. 3
:
φ

(1) φ_1
...
(i) φ_i
...
(j) φ_j où φ_j est $\varphi_i \Rightarrow \psi$
...
(n) ψ

$\varphi \vdash \varphi_i$, donc $\vdash \varphi \Rightarrow \varphi_i$ car $i < n$
$\varphi \vdash \varphi_i \Rightarrow \psi$, donc $\vdash \varphi \Rightarrow (\varphi_i \Rightarrow \psi)$ car $j < n$.

On construit une démonstration :

(1) φ_1
$\vdots$(k) $\varphi \Rightarrow \varphi_i$
$\vdots$
(e) $\varphi \Rightarrow (\varphi_i \Rightarrow \psi)$
(e+1) $(\varphi \Rightarrow (\varphi_i \Rightarrow \psi)) \Rightarrow ((\varphi \Rightarrow \varphi_i) \Rightarrow (\varphi \Rightarrow \psi))$
(e+2) $((\varphi \Rightarrow \varphi_i) \Rightarrow (\varphi \Rightarrow \psi))$ Détachement de e et e+1
(e+3) $\varphi \Rightarrow \psi$ id sur k et e+2.

Donc, $\vdash \varphi \Rightarrow \psi$.

$\diamond$
$\diamond$

$\square$

On déduit une proposition plus générale dont la preuve est déduite de la preuve précédente.
Proposition 1.2

Soit $\Sigma \subseteq \mathcal{P}rop(\mathcal{P})$.
$\Sigma \cup \{\varphi\} \vdash \psi$ ssi $\Sigma \vdash \varphi \Rightarrow \psi$.

Exercice 5 *Rédiger la preuve de la proposition précédente.*

Nous abordons dans la partie suivante l'équivalence des deux notations $\vdash$ et $\models$.

1.5 Cohérence et complétude

Ce paragraphe montre que les deux déductions précédemment présentées sont équivalentes. Il est clair que cette équivalence est presque évidente puisque ces deux notations ont les mêmes propriétés. L'équivalence n'est pas toujours simple à montrer pour des systèmes formels en général et elle n'est pas toujours vraie. Nous introduisons les termes de complétude et de cohérence (ou encore correction) de notre système formel.

► **Théorème 1.4** Cohérence
Si $\vdash \varphi$, alors, $\models \varphi$.

► **Théorème 1.5** Complétude
Si $\models \varphi$, alors, $\vdash \varphi$.

PREUVE: Supposons que $\vdash \varphi$. On utilise une démonstration par induction sur la longueur des démonstrations.

Cas 1

$\vdash \varphi$ et la démonstration de longueur 1.
(1) φ : φ est un axiome. Il faut donc montrer que tout valuation est un modèle de φ.

Sous-Cas 1. 1

: $\alpha \Rightarrow (\beta \Rightarrow \alpha)$

$$\llbracket \alpha \Rightarrow (\beta \Rightarrow \alpha) \rrbracket = \llbracket \alpha \rrbracket \; imp \; \llbracket \beta \Rightarrow \alpha \rrbracket = \llbracket \alpha \rrbracket \; imp \; (\llbracket \beta \rrbracket \; imp \; \llbracket \alpha \rrbracket) = \overline{\llbracket \alpha \rrbracket} + (\overline{\llbracket \beta \rrbracket} + \llbracket \alpha \rrbracket) = 1$$

$$\llbracket (\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma)) \rrbracket = \llbracket \alpha \rrbracket . \llbracket \beta \rrbracket . \overline{\llbracket \gamma \rrbracket} + \llbracket \alpha \rrbracket . \overline{\llbracket \beta \rrbracket} + \overline{\llbracket \alpha \rrbracket} + \llbracket \gamma \rrbracket = 1$$

$$\begin{aligned} & \llbracket (\sim \alpha \Rightarrow \sim \beta) \rrbracket \Rightarrow ((\sim \alpha \Rightarrow \beta) \Rightarrow \alpha) \rrbracket \\ &= \overline{\llbracket \sim \alpha \Rightarrow \sim \beta \rrbracket} + \llbracket (\sim \alpha \Rightarrow \beta) \Rightarrow \alpha \rrbracket \\ &= \overline{\overline{\llbracket \sim \alpha \rrbracket} + \llbracket \sim \beta \rrbracket} + \overline{\llbracket (\sim \alpha \Rightarrow \beta) \rrbracket} + \llbracket \alpha \rrbracket \\ &= \overline{\llbracket \alpha \rrbracket} . \llbracket \beta \rrbracket + \overline{\llbracket \alpha \rrbracket} . \overline{\llbracket \beta \rrbracket} + \llbracket \alpha \rrbracket \\ &= (\llbracket \beta \rrbracket + \overline{\llbracket \beta \rrbracket}) . \overline{\llbracket \alpha \rrbracket} + \llbracket \alpha \rrbracket \\ &= \overline{\llbracket \alpha \rrbracket} + \llbracket \alpha \rrbracket = 1. \end{aligned}$$

◇
◇

Cas 2

:

$\vdash \varphi$ et la démonstration est de longueur n.
Supposons que, pour tout φ dont la démonstration est de longueur < n, $\models \varphi$.
(1) φ_1
⋮
(n) φ_n est φ

Sous-Cas 2. 1

φ est un axiome et dans ce cas $\llbracket \varphi \rrbracket = 1$.

Sous-Cas 2. 2

φ est déduit de φ_i et φ_j, $i,j < n$ par la règle de détachement.Dans ce cas, $\vdash \varphi_i$ est $\vdash \varphi_j$ où $\vdash \varphi_j$ est $\varphi_i \Rightarrow \varphi$. Par hypothèse, on déduit que $\models \varphi_i$ et $\models \varphi_i \Rightarrow \varphi$.
Or $\models \varphi_i \Rightarrow \varphi$ ssi $\varphi_i \models \varphi$
Soit δ un modèle quelconque. δ est un modèle de φ_i, donc un modèle de φ.
◇
◇

□

La propriété réciproque est plus délicate. Elle est vérifiée dans le cadre du calcul propositionnel mais n'est pas toujours vraie, si on considère des systèmes plus généraux. Nous préparons la démonstration de la réciproque par le lemme suivant :

Proposition 1.3

Soit φ une formule à variables $P_1 \dots P_n$.
Soit une valuation $\varepsilon = (\varepsilon_1 \dots \varepsilon_n)$ à laquelle on associe la suite $(P'_1 \dots P'_n)$ telle que :

- P'_i est P_i, si $\varepsilon_i = 1$.
- P'_i est $\sim P_i$, si $\varepsilon_i = 0$.

et la forme $\varphi' = \varphi$, si ε valide φ et $\sim \varphi$.
Alors $P'_1, \dots, P'_n \vdash \varphi'$.

PREUVE: On utilise une récurrence sur la complexité de φ.

Cas 1
: φ est P

Sous-Cas 1. 1
$\varepsilon(P) = 0$, $P' =\sim P$, $\varphi' =\sim \varphi$.
Dans ce cas, $\varphi' \vdash \varphi'$ et $P' \vdash \varphi'$

Sous-Cas 1. 2
$\varepsilon(P) = 1$, $P' = P$, $\varphi' = \varphi$.
$\varphi' \vdash \varphi'$ et $P' \vdash \varphi'$.

◇

Cas 2
: φ est $\sim \psi$

Sous-Cas 2. 1
: ψ vaut 1 est φ vaut 0.
$\psi' = \psi$, $\varphi' =\sim \varphi$
$P'_1 \dots P'_n \ \vdash \ \psi'$ (hypothèse de récurrence)
$P'_1 \dots P'_n \ \vdash \ \psi$
$P'_1 \dots P'_n \ \vdash \ \sim\sim \psi$ (théorème à montrer)
$P'_1 \dots P'_n \ \vdash \ \sim \varphi$
$P'_1 \dots P'_p n \ \vdash \ \varphi'$

◇

Sous-Cas 2. 2
: ψ vaut 0 et φ vaut 1.
$\psi' =\sim \psi$ et $\varphi' = \varphi$
$P'_1 \dots P'_n \ \vdash \ \psi'$
$P'_1 \dots P'_n \ \vdash \ \sim \psi$
$P'_1 \dots P'_n \ \vdash \ \varphi$
$P'_1 \dots P'_n \ \vdash \ \varphi'$

◇

◇

Cas 3
: φ est $\alpha \Rightarrow \beta$

Sous-Cas 3. 1
: α vaut 0 et φ vaut 1. $\varphi = \varphi'$ et $\alpha' =\sim \alpha$
$P'_1 \dots P'_n \ \vdash \ \alpha'$ (hypothèse de récurrence)
$P'_1 \dots P'_n \ \vdash \ \sim \alpha$
$P'_1 \dots P'_n \ \vdash \ \alpha \Rightarrow \beta$ (théorème à montrer)
$P'_1 \dots P'_n \ \vdash \ \varphi$
$P'_1 \dots P'_n \ \vdash \ \varphi'$

◇

Sous-Cas 3. 2
: β vaut 1 et φ vaut 1.
$P'_1 \dots P'_n \ \vdash \ \beta'$
$P'_1 \dots P'_n \ \vdash \ \beta$
$P'_1 \dots P'_n \ \vdash \ \alpha \Rightarrow \beta$ (introduction)
$P'_1 \dots P'_n \ \vdash \ \varphi$

$P'_1 \dots P'_n \ \vdash \ \varphi'$ $\diamond$

Sous-Cas 3. 3

: α vaut 1, β vaut 0 et φ vaut 0.
$\alpha' = \alpha$, $\beta' = \sim \beta$, $\varphi' = \sim \varphi$.
$P'_1 \dots P'_n \ \vdash \ \alpha'$
$P'_1 \dots P'_n \ \vdash \ \beta'$
$P'_1 \dots P'_n \ \vdash \ \alpha$
$P'_1 \dots P'_n \ \vdash \sim \beta$
$P'_1 \dots P'_n \ \vdash \sim (\alpha \Rightarrow \beta)$
$P'_1 \dots P'_n \ \vdash \sim \varphi$
$P'_1 \dots P'_n \ \vdash \ \varphi'$ $\diamond$

$\diamond$

□

Proposition 1.4

Si $\models \varphi$, alors, $\vdash \ \varphi$.

PREUVE:
Notons $P_1 \dots P_n$ les variables de φ telle que $\models \varphi$.
Soit δ une valuation quelconque.
$P'_1 \dots P'_n \ \vdash \ \varphi$
où $P'_1 \dots P'_n$ est une suite associée à δ.
$P'_1 \dots P'_n \ \vdash \ \varphi$
et $P'_1 \dots P'_{n-1} \sim P'_n \ \vdash \ \varphi$.
Donc on peut supprimer P'_n. D'où $\vdash \ \varphi$. CQFD.

□

► **Théorème 1.6**

Soit φ une formule de $\mathcal{P}rop(\mathcal{P})$. $\vdash \ \varphi$ ssi $\models \varphi$.

On peut se poser la question de la déduction automatique d'une formule du calcul des propositions.

► **Théorème 1.7**

Le problème de la détermination du statut d'une formule φ de $\mathcal{P}rop(\mathcal{P})$ par rapport à la propriété de tautologie est décidable.

Cela signifie qu'il existe un algorithme permettant de valider ou d'invalider une propriété exprimée sous la forme du calcul des propositions. Par exemple, on peut tester toutes les valuations possibles d'une formule propositionnelle avec n variables proporitionnelles : il y a 2^n valuations à tester. Nous abordons dans la section suivante une autre façon de raisonner sur les formules propositionnelles, le calcul des séquents.

1.6 Calcul des séquents

GENTZEN[4] introduit le calcul des séquents. Son clcul est noté LK (pour calcul logique classique). Un séquent est formé de deux ensembles finis de formules Γ et Δ séparés par le symbole $\rightarrow$. L'intuition de l'écriture $\Gamma \rightarrow \Delta$ est la suivante :

— $\Gamma = \{\phi_1, \dots, \phi_n\}$
— $\Delta = \{\psi_1, \dots, \psi_p\}$
— $\phi_1 \wedge \dots \wedge \phi_n \Rightarrow \psi_1 \vee \dots \vee \psi_p$

— La partie gauche constitue les hypothèses supposées vraies et la partie droite les conclusions possibles mais pas nécessairement toutes les conclusions.

☼**Définition 1.4**

Un séquent est une paire (Γ, Δ) d'ensembles finis de formules. On note aussi un séquent sous la forme $\Gamma \rightarrow \Delta$ ou encore $\Gamma \vdash \Delta$.

Si l'ensemble des formules de Γ est $\{A_1, \ldots, A_n\}$ et si l'ensemble des formules de Δ est $\{B_1, \ldots, B_p\}$, alors on écrira le séquent $\Gamma \rightarrow \Delta$ sous la forme $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_p$. L'interprétation d'une telle forme est la suivante :*Si les formules $A_1, \ldots, A_n$ sont toutes vraies, alors l'une au moins des formules $B_1, \ldots, B_p$ est vraie.*. Pour simplifier certaines écritures, on écrira $\Gamma, A \rightarrow \Delta$ à la place de $\Gamma \cup \{A\} \rightarrow \Delta$. Certaines formes particulières sont possibles : $\varnothing \rightarrow \Delta$ ou $\Gamma \rightarrow \varnothing$.

☼**Définition 1.5**

Soit une valuation δ de $\mathcal{P}$ dans $\mathcal{B}$ associant à tout symbole de $\mathcal{P}$ une valeur booléenne de $\{0, 1\}$. On étend l'interprétation de cette valuation aux séquents comme suit :

si $\rrbracket A \llbracket(\delta) = 0$ pour une formule de Γ, ou si $\rrbracket B \llbracket(\delta) = 1$ pour une formule B de Δ, $\llbracket \Gamma \rightarrow \Delta \rrbracket(\delta) = 1$ et vaut 0 sinon.

Les conséquences de cette définition sont les suivantes :

— $\llbracket \rightarrow \rrbracket(\delta) = 0$: les deux ensembles de formules sont vides.
— $\llbracket \rightarrow B \rrbracket(\delta) = \llbracket B \rrbracket(\delta)$: la formule B est valide pour la valuation δ.
— $\llbracket A \rightarrow A \rrbracket(\delta) = 1$: si la formule A est valide pour la valuation δ, alors $\neg A$ n'est pas valide pour δ et réciproquement ; en particulier, $A \rightarrow A$ est valide pour toute valuation δ.
— $\llbracket false \rightarrow \rrbracket(\delta) = 1$ et $\llbracket \rightarrow true \rrbracket(\delta) = 1$: une simple analyse des cas selon la valuation et la propriété des deux constantes permet de s'assurer de la correctiond e cette affirmation.
— Si $\llbracket \Gamma \rightarrow \Delta \rrbracket(\delta) = 1$, alors $\llbracket \Gamma \rightarrow \Delta, B \rrbracket(\delta) = 1$, pour toute formule B et pour toute valuation δ.
—
— Si $\llbracket \Gamma \rightarrow \Delta \rrbracket(\delta) = 1$, alors $\llbracket \Gamma, A \rightarrow \Delta \rrbracket(\delta) = 1$, pour toute formule A et pour toute valuation δ.

Le symbole $\rightarrow$ est un *métasymbole* et on ne pourra pas écrire des expressions de la forme suivante : $A \rightarrow (B \rightarrow C)$ mais $A \rightarrow (B \Rightarrow C)$. Nous dérivons des règles de *calcul* pour dériver des séquents. Nous avons des règles d'introduction et d'élimination pour chacun des connecteurs logiques. Nous décomposons les règles en figures opératoires (signes logiques, connecteurs logiques) et les figures structurelles de déduction. Les règles sont rassemblées à la figure 1.1.

☼**Définition 1.6**

Une preuve est un arbre étiquetté par des séquents satisfaisant les règles de construction suivantes :

— Si un nœud n est étiquetté par un séquent $\Gamma \rightarrow \Delta$ et si ce nœud est une feuille, alors ce séquent doit être un axiome.
— Si un nœud n étiquetté par un séquent $\Gamma \rightarrow \Delta$ admet des enfants n_1 :$\Gamma_1 \rightarrow \Delta_1$ et n_2 :$\Gamma_2 \rightarrow \Delta_2$, alors les séquents des enfants doivent être des prémisses d'une règle du calcul des séquents dont la conclusion est le séquent étiquettant n.

Nous donnons quelques exemples de preuve dans ce calcul. Une preuve consiste donc à construire un arbre correct par rapport aux règles ci-dessus.

Exemple 1.4 Preuve de $\rightarrow P \Rightarrow (Q \Rightarrow P$

Règles du calcul des séquents pour le calcul propositionnel

Axiomes

- Pour toute formule A, $A \rightarrow A$ est un axiome.
- $false \rightarrow$ est un axiome.
- $\rightarrow true$ est un axiome.

Règles

- **Règles structurelles** :
  - Si Γ_1, Γ_2, Δ_1, Δ_2 sont des ensembles finis de formules tels que $\Gamma_1 \subseteq \Gamma_2$, $\Delta_1 \subseteq \Delta_2$, alors
  $$\frac{\Gamma_1 \rightarrow \Delta_1}{\Gamma_2 \rightarrow \Delta_2}$$
  - atténuations :
    - $\dfrac{\Gamma \rightarrow \Delta}{\phi, \Gamma \rightarrow \Delta}$
    - $\dfrac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \psi}$
  - contraction :
    - $\dfrac{\phi, \phi, \Gamma \rightarrow \Delta}{\phi, \Gamma \rightarrow \Delta}$
    - $\dfrac{\Gamma \rightarrow \Delta, \psi, \psi}{\Gamma \rightarrow \Delta, \psi}$
  - permutation :
    - $\dfrac{\Gamma_1, \phi, \psi, \Gamma_2 \rightarrow \Delta}{\Gamma_1, \psi, \phi, \Gamma_2 \rightarrow \Delta}$
    - $\dfrac{\Gamma \rightarrow \Delta_1, \phi, \psi, \Delta_2}{\Gamma \rightarrow \Delta_1, \psi, \phi, \Delta_2}$
  - **Coupure**
  $$\frac{\Gamma \rightarrow \Delta, \phi, \quad \phi, \Lambda \rightarrow \Pi}{\Gamma, \Lambda \rightarrow \Delta, \Pi}$$
- **Négation**
$$\frac{\Gamma \rightarrow \Delta, B}{\Gamma, \neg B \rightarrow \Delta} \qquad \frac{\Gamma, A \rightarrow \Delta}{\Gamma \rightarrow \Delta, \neg A}$$
- **Conjonction**
$$\frac{\Gamma, A, B \rightarrow \Delta}{\Gamma, A \wedge B \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow \Delta, A \quad \Gamma \rightarrow \Delta, B}{\Gamma \rightarrow \Delta, A \wedge B}$$
- **Disjonction**
$$\frac{\Gamma, A \rightarrow \Delta \quad \Gamma, B \rightarrow \Delta}{\Gamma, A \vee B \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow \Delta, A, B}{\Gamma \rightarrow \Delta, A \vee B}$$
- **Implication**
$$\frac{\Gamma \rightarrow \Delta, A \quad \Gamma, B \rightarrow \Delta}{\Gamma, A \Rightarrow B \rightarrow \Delta} \qquad \frac{\Gamma, A \rightarrow \Delta, B}{\Gamma \rightarrow \Delta, A \Rightarrow B}$$

FIGURE 1.1 – LK Règles du calcul des séquents pour le calcul propositionnel

$$
\begin{array}{c}
1 : P \to P \\
2 : P, Q \to P \\
3 : P \to Q \Rightarrow P \\
4 : \to P \Rightarrow (Q \Rightarrow P)
\end{array}
$$

Exemple 1.5 Preuve de $\to \neg(P \wedge Q) \Rightarrow (\neg Q \vee \neg P)$

$$
\begin{array}{c}
1 : P \to P \qquad 2 : Q \to Q \\
1.1 : P, Q \to P \quad 2.2 : P, Q \to Q \\
3 : P, Q \to P \wedge Q \\
4 : Q \to P \wedge Q, \neg P \\
5 : \to P \wedge Q, \neg P, \neg Q \\
6 : \neg(P \wedge Q) \to \neg P, \neg Q \\
6 : \neg(P \wedge Q) \to \neg P \vee \neg Q \\
7 : \to \neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q
\end{array}
$$

Les axiomes et les règles constituent un système correct par rapport à la sémantique des valuations.

► **Théorème 1.8** Correction du calcul des séquents

Si une formule A est un théorème pour le calcul des séquents, alors la formule A est une tautologie.

La justification est déduite des propriétés données ci-dessus pour les axiomes et les règles.

1.7 Conclusion

Deux méthodes de preuve coÃ¯ncident dans le calcul des propositions. D'une part, une preuve peut être menée en rapport avec les modèles et dans ce cas les propositions ont des valeurs de vérité qui précisent effectivement la vérité courante de ces dernières. D'autre part, la preuve d'une formule est une construction mécanique et presque automatique d'une suite d'énoncés déduits les uns des autres. Ce point de vue correspond très nettement à une certaine mécanisation du raisonnement car le principe caché est celui de la substitution de propositions dans les schémas d'axiomes. Dans une certaine mesure, la déduction formelle est plus opérationnelle ou plus calculatoire que la déduction sémantique. Mais, une preuve permet de trouver des contre-exemples comme dans le cas de la méthode des tableaux [8]. Enfin, le calcul des propositions est décidable et peut être étendu par des modalités pour construire des logiques modales et temporelles. Nous donnerons en annexe une présentation rapide de ces cadres.

CHAPITRE 2

CALCUL DES PRÉDICATS

Sommaire

2.1 Préliminaires

Nous avons vu les pductionropriétés de la relation " $\longrightarrow$ " pour le cas des formules propositionnelles. Nous allons introduire la notion de variable d'individu et la notion de quantificateurs. Les classes suivantes sont définies :

1. Symboles logiques : $\sim$, $\Rightarrow$, $\forall$,
2. Symboles de variables : $\mathcal{V} = \{x, y, z, \ldots\}$
3. Symboles de constantes : $\mathcal{C} = \{a, b, c, \ldots\}$
4. Symboles de fonctions : $\mathcal{F} = \{f, g, h, \ldots\}$
5. Symboles de relations : $\mathcal{R} = \{P, Q, R, \ldots\}$
6. Symboles logiques auxiliaires : $\wedge$, $\vee$, $\equiv$, $\exists$, $\ldots$

Exemple 2.6

1. $\forall x(\exists y\ (P(x,y) \wedge R(x,y)) \ \Rightarrow\ Q(z,x))$
2. $\forall x((>(x,0)) \ \Rightarrow \exists y\ (>(y,0) \ \Rightarrow\ (-(x,y) < 0)))$

Remarque

la quantification porte sur les variables de constantes. Il n'y a pas de quantification sur les fonctions.

Tout triplet $\mathcal{L} = (\mathcal{C}; \mathcal{R}; \mathcal{F})$ est appelé un langage du premier ordre. Deux catégories d'objets sont à définir :

$\longrightarrow$ les termes de $\mathcal{L}$
$\longrightarrow$ les formules de $\mathcal{L}$
Les termes de $\mathcal{L}$, notés $\mathcal{T}[\mathcal{L}]$, sont les éléments de la $\mathcal{F}\cup\mathcal{C}$-algèbre $M(\mathcal{F}, \mathcal{V})$.
Les formules de $\mathcal{L}$ sont obtenues à partir des formules atomiques de $\mathcal{L}$, noté $\mathcal{A}[\mathcal{L}]$:
$\mathcal{A}[\mathcal{L}] = \{R(t_1, \ldots, t_n)/t_1, \ldots, t_n \in M(\mathcal{F}, \mathcal{V})\}$ et $R \in \mathcal{R}\}$
Les formules atomiques permettent de construire les formules de $\mathcal{L}$. Il faut noter qu'il y a une formule particulière atomique construite à l'aide de la relation d'égalité ; dans ce cas, on parle de calcul des prédicats avec égalité.
Une formule de f de $\mathcal{L}$, ou un élément de $\mathcal{F}(\mathcal{L})$, est :

1. une formule atomique de $\mathcal{A}[\mathcal{L}]$
2. la négation d'une formule : $\sim f$ où $f \in \mathcal{F}(\mathcal{L})$
3. la forme implicative $f_1 \Rightarrow f_2$ où $f_1, f_2 \in \mathcal{F}(\mathcal{L})$
4. la forme quantifiée $\forall x(f)$ où $x \in \mathcal{V}$ et $f \in \mathcal{F}(\mathcal{L})$
5. le résultat de l'application finie des règles 1,2,3,4 et uniquement celles-ci.

L'occurence d'une variable x d'une formule f est liée, si elle se trouve dans la partie d'un quantificateur $\forall x$, sinon on dit qu'elle est libre.

Exemple 2.7

$\forall x(\exists y(P(f(x,y)) \wedge Q(x,z))) \wedge R(x,z)$
— les occurences de z sont libres
— les 2 premières occurences de x sont liées
— la dernière occurence de x est libre
— les occurences de y sont liées

Une variable peut avoir des occurences libres et des occurences liées. Une variable est libre, s'il y a au moins une occurence libre de celle-ci dans la formule considérée. On note $\mathcal{V}(\varphi)$, l'ensemble des variables libres de φ.

Exercice 6 *Donner une définition structurelle de la notion de variable libre dans une formule.*

Soit x une variable, t un terme et φ une formule. On dit que t est libre pour x dans φ, quand aucune occurence libre de x dans φ n'est dans le champ d'un quantificateur liant une variable de t.

Exemple 2.8

$\varphi \ = \ \forall y \ (\exists z \ (R(x, y, z)))$

(1) $t = f(x, u)$: t est libre pour x dans φ
(2) $t = f(x, y)$: t n'est pas libre pour x dans φ car y est lié par $\forall$.

Une formule φ est ouverte, s'il y a au moins une occurence libre d'une variable. Sinon elle est close. Nous noterons $\varphi(x_1, x_2, \ldots, x_n)$ une formule ouverte avec $x_1, x_2, \ldots, x_n$ les variables ayant des occurences libres. Une formule libre peut être close par quantification $\forall$:

$\forall x_1(\forall x_2(\ldots(\forall x_n(\varphi(x_1, \ldots, x_n))\ldots))$, c'est la cloture universelle de φ. La formule φ peut être close par $\exists$, c'est la cloture existencielle.

Une formule φ est dite sous forme prénexe, quand elle s'écrit
$Q_1x_1(Q_2x_2(\ldots(Q_nx_n(\varphi(x_1, \ldots, x_n, y_1, \ldots, y_p))$ où $Q_i \in \{\forall, \exists\}$ et $\varphi(x_1, \ldots, y_p)$ sans quantificateur.

Soit t un terme, φ une formule et x une variable de φ. $\varphi[t/x]$ ou $\varphi_x[t]$ est la formule obtenue en remplaçant les occurences libres de x par t où t est libre pour x dans φ. Avant de substituer une variable par un terme, on prendra soin de vérifier cette condition d'application.

Exemple 2.9

Soit $\varphi = \ \exists y.(x = 2y)$
- $\varphi_x[y{+}1] \ = \ \exists y.\ (y{+}1 = 2y)$ n'a pas de sens car $y{+}1$ n'est pas libre pour y dans ϕ.
- $\varphi_x[z{+}5] \ = \ \exists y.\ (z{+}5 = 2y)$.

La substitution est donc à manipuler avec soin !

2.2 Interpretation des formules

Une interprétation $\mathcal{I}$ de $\mathcal{L}$ ou une réalisation de $\mathcal{L}$ est la donnée :

1. d'un ensemble non-vide $|\mathcal{I}|$, appelé domaine de $\mathcal{I}$.
2. pour chaque symbole de fonction f de $\mathcal{F}$, d'une fonction ou application $f_{\mathcal{I}} : |\mathcal{I}|^n \longrightarrow |\mathcal{I}|$.
3. pour chaque symbole de relation R de $\mathcal{R}$, d'une relation $r_{\mathcal{I}}$ sur $|\mathcal{I}|$.

Une interprétation $\mathcal{I}$ de $\mathcal{L}$ est une $\mathcal{F}$-algèbre où les relations sont des fonctions à valeurs dans B où le domaine contient B. Les logiciens distinguent les constantes des autres fonctions. Le symbole d'égalité est interprété par l'égalité. Les constantes seront interprétées par des éléments de $|\mathcal{I}|$. Le langage étendu pour une interprétation sera obtenu en ajoutant des symboles de constantes pour les valeurs des éléments de $|\mathcal{I}|$.
Soit une formule close φ et $\mathcal{I}$ une interprétation.

1. φ est atomique :
   φ s'écrit $R(t_1, \ldots, t_n)$ où $R \in \mathcal{R}$ et $t_1 \ldots t_n$ des termes clos (sans variables).
   $\mathcal{I} \models \varphi$ ssi $(t_{1\mathcal{I}}, \ldots, t_{n\mathcal{I}}) \in R_{\mathcal{I}}$
2. φ est $\sim \Psi$:
   $\mathcal{I} \models \varphi$ ssi $non(\mathcal{I} \models \Psi)$
3. φ est $\alpha \Rightarrow \beta$:
   $\mathcal{I} \models \varphi$ ssi $(\mathcal{I} \models \beta \ ou \ \mathcal{I} \models \sim \alpha)$
4. φ est $\forall x(\psi(x))$:
   $\mathcal{I} \models \varphi$ ssi , pour tout i de $|\mathcal{I}|, \mathcal{I} \models \psi(\bar{i}/x)$.

Le point (4) suppose que les quantificateurs lient des variables libres ayant des occurences sinon l'interprétation est lue même celle sans le quantificateur. "$\mathcal{I} \models \varphi$" se lit "I satisfait φ"

Soit φ une formule ouverte. Soit δ une application de $\mathcal{V}$ dans $|\mathcal{I}|$. δ est une valuation.
On dit qu'une formule ouverte $\varphi(x_1, \ldots, x_n)$ est satisfaite dans une interprétation $\mathcal{I}$, si elle est satisfaite pour toutes les valuations possibles :

$\mathcal{I} \models \varphi(x_1 \ldots x_n)$ ssi $\mathcal{I} \models \forall x_1(\forall x_2 \ldots (\forall x_n \varphi(x_1 \ldots x_n)) \ldots)$.

On note $\mathcal{I}, \delta \models \varphi(x_1 \ldots x_n)$ la satisfaction de φ dans $\mathcal{I}$ pour δ.

☺ **Propriété 2.3**
Soit φ une formule close.

1. φ est satisfaite dans $\mathcal{I}$ ssi $\sim \varphi$ n'est pas satisfaite dans $\mathcal{I}$.
2. Soit $\mathcal{I}$ une interprétation. φ est soit satisfaite, soit non satisfaite.
3. Si $\mathcal{I}$ satisfait φ et $\varphi \Rightarrow \psi$, alors $\mathcal{I}$ satisfait ψ.

Une formule φ est satisfaisable, s'il existe une interprétation $\mathcal{I}$ telle que $\mathcal{I} \models \varphi$.
Une formule φ est valide, si elle est satisfaite dans toutes les interprétations : on note $\models \varphi$.
Deux formules φ et ψ sont équivalentes, si, pour toute interprétation $\mathcal{I}$, $\mathcal{I} \models \varphi$ ssi $\mathcal{I} \models \psi$.

Exemple 2.10

1. $\forall x \varphi(x, x_1, \ldots, x_n)$ équivaut à $\forall y\ \varphi(y, x_1, \ldots, x_n)$.
2. $\forall x\ \varphi(x)$ équivaut à $\sim \exists x \sim \varphi(x)$

Une interprétation $\mathcal{I}$ satisfaisant une formule φ est un modèle de φ. Une formule φ qui admet un modèle est non contradictoire. Une formule valide est une thèse.
Soit Σ un ensemble de formules. Un modèle pour Σ est une interprétation $\mathcal{I}$ satisfaisant chaque formule de Σ :

1. Soit φ et ψ deux formules closes. ψ se déduit sémantiquement de φ, si tout modèle de φ est un modèle de ψ : $\varphi \models \psi$.
2. Soit Σ un ensemble de formules closes et ψ une formule close. $\Sigma \models \psi$, si tout modèle de Σ est un modèle de ψ.

► **Théorème 2.9** Théorème de la déduction sémantique
Soient φ, ψ des formules closes. Soit Σ un ensemble de formules closes.

1. $\varphi \models \psi$ ssi $\models \varphi \Rightarrow \psi$
2. $\Sigma \cup \{\varphi\} \models \psi$ ssi $\Sigma \models \varphi \Rightarrow \psi$

Exercice 7 *Démontrer les 2 résultats du théorème.*

Un ensemble Σ de formules closes est consistant, si Σ a un modèle.
Proposition 2.5

Soit Σ ensemble de formules closes et φ closes. $non(\Sigma \models \varphi)$ ssi $\Sigma \cup \{\sim \varphi\}$ est consistant.

Preuve $\Sigma \cup \sim \varphi$ consistant ssi il existe $\mathcal{I}$ modèle de Σ et de $\sim \varphi$ ssi il existe $\mathcal{I}$ modèle de Σ et non modèle de Σ ssi $non(\Sigma \models \varphi)$. *fin de la preuve*

2.3 Axiomatique et déduction formelle

Nous étendons le système propositionnel comme suit :

1. $\varphi \Rightarrow (\psi \Rightarrow \varphi)$
2. $(\varphi \Rightarrow (\psi \Rightarrow \delta)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \delta))$
3. $(\sim \varphi \Rightarrow \sim \psi) \Rightarrow ((\sim \varphi \Rightarrow \psi) \Rightarrow \varphi)$
4. $\forall x(\varphi(x)) \Rightarrow \varphi(t)$ où t est libre pour x dans φ.
5. $\forall x(\varphi \Rightarrow \psi(x)) \Rightarrow (\varphi \Rightarrow \forall x\psi(x))$ où x n'a pas d'occurence libre dans φ.
6. $\forall x(x = x)$
7. $(x = y) \Rightarrow (\varphi(x/z) \Rightarrow \varphi(y/z))$, où ϕ est une formule atomique.

DETACHEMENT : $\dfrac{\varphi, \varphi \Rightarrow \psi}{\psi}$

GENERALISATION : $\dfrac{\varphi}{\forall x(\varphi)}$

Exercice 8 *Démontrer :*

1. $\vdash t = t$

2. $\vdash x = y \Rightarrow y = x$

Proposition 2.6

Soit φ une tautologie propositionnelle.
Si φ' est une substitution des variables propositionnelles par des formules closes, alors φ' est démontrable.

Preuve La preuve de φ' n'utilise que les règles propositionnelles.
Les notions de démonstration sont étendues pour prendre en compte les nouveaux axiomes et la nouvelle règle. *fin de la preuve*

► **Théorème 2.10**
Soit φ démontrable ($\vdash \varphi$). Alors $\models \varphi$.

Preuve Par induction sur la longueur de la démonstration de φ en montrant que les axiomes sont valides. *fin de la preuve*

► **Théorème 2.11** Théorème de la Déduction
Soit φ une formule close. $\{\varphi\} \vdash \psi$ ssi $\vdash \varphi \Rightarrow \psi$

Le résultat est faux, en général, si φ n'est pas close. Un ensemble de formules closes Σ est consistant, s'il n'existe pas ψ telle que $\Sigma \vdash$ et $\Sigma \vdash \sim \psi$.

► **Théorème 2.12** Théorème de finitude
Σ est consistant ssi toute partie finie de Σ est consistante.

► **Théorème 2.13** (final)
Soit Σ un ensemble de formules closes et φ une formule.
$\Sigma \vdash \varphi$ ssi $\Sigma \models \varphi$

Règles du calcul des séquents pour le calcul des prédicats du premier ordrel

Axiomes

- Pour toute formule A, $A \to A$ est un axiome.
- $false \to$ est un axiome.
- $\to true$ est un axiome.

Règles

- **Règles structurelles** :
  - Si Γ_1, Γ_2, Δ_1, Δ_2 sont des ensembles finis de formules tels que $\Gamma_1 \subseteq \Gamma_2$, $\Delta_1 \subseteq \Delta_2$, alors
  $$\frac{\Gamma_1 \to \Delta_1}{\Gamma_2 \to \Delta_2}$$
  - atténuations :
    - $\dfrac{\Gamma \to \Delta}{\phi, \Gamma \to \Delta}$
    - $\dfrac{\Gamma \to \Delta}{\Gamma \to \Delta, \psi}$
  - contraction :
    - $\dfrac{\phi, \phi, \Gamma \to \Delta}{\phi, \Gamma \to \Delta}$
    - $\dfrac{\Gamma \to \Delta, \psi, \psi}{\Gamma \to \Delta, \psi}$
  - permutation :
    - $\dfrac{\Gamma_1, \phi, \psi, \Gamma_2 \to \Delta}{\Gamma_1, \psi, \phi, \Gamma_2 \to \Delta}$
    - $\dfrac{\Gamma \to \Delta_1, \phi, \psi, \Delta_2}{\Gamma \to \Delta_1, \psi, \phi, \Delta_2}$
  - **Coupure**
  $$\frac{\Gamma \to \Delta, \phi, \quad \phi, \Lambda \to \Pi}{\Gamma, \Lambda \to \Delta, \Pi}$$
- **Négation**
$$\frac{\Gamma \to \Delta, B}{\Gamma, \neg B \to \Delta} \qquad \frac{\Gamma, A \to \Delta}{\Gamma \to \Delta, \neg A}$$
- **Conjonction**
$$\frac{\Gamma, A, B \to \Delta}{\Gamma, A \wedge B \to \Delta} \qquad \frac{\Gamma \to \Delta, A \quad \Gamma \to \Delta, B}{\Gamma \to \Delta, A \wedge B}$$
- **Disjonction**
$$\frac{\Gamma, A \to \Delta \quad \Gamma, B \to \Delta}{\Gamma, A \vee B \to \Delta} \qquad \frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \vee B}$$
- **Implication**
$$\frac{\Gamma \to \Delta, A \quad \Gamma, B \to \Delta}{\Gamma, A \Rightarrow B \to \Delta} \qquad \frac{\Gamma, A \to \Delta, B}{\Gamma \to \Delta, A \Rightarrow B}$$
- **Quantificateur universel** : x n'a pas d'occurence libre dans Γ et r n'est pas liée pour x dans $A(x)$:
$$\frac{\Gamma \to A(x)}{\Gamma \to \forall x.A(x)} \qquad \frac{}{\Gamma, \forall x.A \to A(r)}$$
- **Quantificateur existenciel** : x n'a pas d'occurence libre dans Γ, x n'a pas d'occurence libre dans C et r n'est pas liée pour x dans $A(x)$:
$$\frac{}{\Gamma, A(r) \to \exists r.A(r)} \qquad \frac{\Gamma, A(x) \to C}{\Gamma, \exists x.A(x) \to C}$$

FIGURE 2.1 – LK Règles du calcul des séquents pour le premier ordre

2.4 Calcul de Gentzen

Le symbole $\to$ est un *métasymbole* et on ne pourra pas écrire des expressions de la forme suivante : $A \to (B \to C)$ mais $A \to (B \Rightarrow C)$. Nous dérivons des règles de *calcul* pour dériver des séquents. Nous avons des règles d'introduction et d'élimination pour chacun des connecteurs logiques. Nous décomposons les règles en figures opératoires (signes logiques, connecteurs logiques) et les figures structurelles de déduction. Les règles sont rassemblées à la figure 2.1.

☼**Définition 2.7**

Une preuve est un arbre étiquetté par des séquents satisfaisant les règles de construction suivantes :

— Si un nœud n est étiquetté par un séquent $\Gamma \rightarrow \Delta$ et si ce nœud est une feuille, alors ce séquent doit être un axiome.

— Si un nœud n étiquetté par un séquent $\Gamma \rightarrow \Delta$ admet des enfants n_1 :$\Gamma_1 \rightarrow \Delta_1$ et n_2 :$\Gamma_2 \rightarrow \Delta_2$, alors les séquents des enfants doivent être des prémisses d'une règle du calcul des séquents dont la conclusion est le séquent étiquettant n.

Les axiomes et les règles constituent un système correct par rapport à la sémantique des valuations.

► **Théorème 2.14** Correction du calcul des séquents

Si une formule A est un théorème pour le calcul des séquents, alors la formule A est une tautologie.

La justification est déduite des propriétés données ci-dessus pour les axiomes et les règles.

2.5 Théorie et Incomplétude

Dans la partie précédente, nous avons présenté le résultat sur la complétude du calcul des prédicats avec égalité. En nous fondant sur un langage du premier ordre et sur une axiomatique du premier ordre, on peut définir la notion de théorie du premier ordre.. Une théorie du premier ordre est une extension du calcul des prédicats par des axiomes caractérisant des fonctions ou des opérations.

Exemple 2.11

L'arithmétique de PEANO est une théorie du premier ordre définie comme suit : on considère les axiomes déjà connus du premier ordre et on ajoute les axiomes suivants :

— $\forall x.(0 \neq S(x))$

— $\forall x, y.(S(x) = S(y)) \Rightarrow (x = y)$

— $\forall x.(x+0 = x)$

— $\forall x, y.(x+S(y) = S(x+y))$

— $\forall x.(x.0 = 0)$

— $\forall x, y.(x.S(y) = x.y+x)$

— $\phi(0) \wedge (\forall x.(\phi(x) \Rightarrow \phi(S(x))) \Rightarrow (\forall x.\phi(x)))$

Le dernier axiome est l'axiome d'induction. Certains modèles satisfaisant ces axiomes sont dits standards et d'autres sont dits non standards. Les modèles standards sont les modèles classiques et en particulier l'ensemble des naturels avec les opérations classiques est un modèle standard.

2.6 Conclusion

Le calcul des prédicats est indécidable. Cela signifie que, bien qu'il soit complet, il n'existe pas de procedure de décision associé à celui-ci. L'ajout de variables et de quantification conduit à cette indécidabilité mais la logique temporelle qui peut être considérée comme uine extension du calcul des propositions est décidable. En fait, la logique temporelle comprend certaines formes de quantificatioin qui assurent la décidabilité. Enfin, la méthode des tableaux [9] peuut être étendue au calcul des prédicats et son intérêt réside dans la découverte de contre-exem[ples. Le théorème de complétude de GÖDEL et le théorème d'incomplétude de GÖDEL sont les deux résultats à retenir. Pour le cas du théorème d'incomplétude , la conséquence est assez importante puisqu'elle affirme l'inexistence de système formel complet pour toute théorie de l'arithmétique formelle . certaines restrictions de l'arithmétique sont complète comme l'arithmétique de Presburger qui n'a que le + et les suc. L'ajout du * conduit à un système incomplet.

CHAPITRE 3

MÉTHODE DE RÉSOLUTION

Sommaire

Nous avons exposé un certain nombre de techniques et de systèmes logiques. Nous nous préoccupons dans ce paragraphe de la méthode de résolution. Afin de simplifier l'approche, nous nous intéressons au cas propositionnel dans un premier temps puis au cas du calcul des prédicats du 1er ordre.
La méthode de résolution est due à ROBINSON (1965). Cette méthode est basée sur une seule règle appelée règle de résolution et appliquée sur des propositions ou formules en forme normale conjonctive.

3.1 Calcul Propositionnel

Une formule φ du calcul propositionnel est en forme normale conjonctive, si φ s'écrit sous la forme $\varphi_1 \wedge \varphi_2 \wedge \varphi_n$ où chaque φ_i est une disjonction finie de variables propositionnelles ou de négation de variables propositionnelles :

φ_i est $B_{i,1} \vee \ldots \vee B_i, k_i$ et $B_{i,j} \in P, \sim P$.
Chaque $B_{i,j}$ est aussi appelé un littéral.
Une formule φ_i est une clause. Une clause sera notée C_i.

Une formule φ sous forme clausale pourra être représentée par un ensemble de clauses : $\varphi \ = \ C_1 \wedge \ldots \wedge C_n$ ou $\varphi \ = \{C_1, \ldots, C_n\}$.

Exemple 3.12

1. $\{P, R \vee S, \sim P \vee V, \sim Q \vee \sim P\}$
2. $\{R \vee S \vee T, T \vee V, V \vee T\}$

Une clause C sera représentée par un ensemble de littéraux de cette clause.

$C = B_1 \vee \ldots \vee B_p$ sera représentée par $\{B_1, \ldots, B_p\}$ et on évitera de confondre le $\vee$ et le $\wedge$.
Par exemple, $\{P, R \vee S, \sim P \vee V, \sim Q \vee \sim P\}$ sera représentée par $\{P, \{R, S\}, \{\sim P, V\}, \{\sim Q, \sim P\}\}$.

Lemme 3.3.1. 1

Soient α, β, γ des formules du calcul propositionnel.

1. $\vdash ((\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma)) \Rightarrow ((\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma) \wedge (\alpha \vee \beta))$
2. $\vdash ((\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma) \wedge (\alpha \vee \beta)) \Rightarrow ((\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma))$

PREUVE:
Utilisons la déduction naturelle.

$$\cfrac{\cfrac{\cfrac{\cfrac{1 \longrightarrow \alpha}{1 \longrightarrow \alpha \vee \gamma} \quad \cfrac{2 \longrightarrow \beta}{2 \longrightarrow \beta \vee \sim \gamma} \quad \cfrac{1 \longrightarrow \alpha}{1 \longrightarrow \alpha \vee \beta}}{1,2 \longrightarrow (\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma) \wedge (\alpha \vee \beta)}}{1,2 \longrightarrow (\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma)}}{\longrightarrow (\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma) \wedge (\alpha \vee \beta) \Rightarrow (\alpha \vee \gamma) \wedge (\beta \vee \sim \gamma)}$$

□

☼**Définition 3.8**

Soient deux clauses C_1 et C_2. Une clause C est une résolvante de C_1 et C_2, s'il existe un littéral L tel que $L \in C_1$ et $\overline{L} \in C_2$ et $C = (C_1 - L) \cup (C_2 - \overline{L})$.
On note $\overline{L}$ le conjugué de L.

► **Théorème 3.15**
Soit S un ensemble de clauses.
Soit C une résolvante de clauses de S.
S est logiquement équivalent à $S \cup C$.

On pourait écrire aussi : $S = C_1, \ldots, C_n$
$\vdash C_1 \wedge \ldots \wedge C_n \Leftrightarrow C_1 \wedge \ldots \wedge C_n \wedge C$ où C est une résolvante de S.

PREUVE:Déduite du lemme précédent. □

Exemple 3.13

Soit $S = \{\{P, Q\}, \{\sim P, \sim Q\}, \{\sim P, R\}\}$
$\{Q, \sim Q\}, \{P, \sim P\}, \{Q, R\}$ sont des résolvantes de S

La méthode de résolution consiste à ajouter une résolvante à l'ensemble des clauses. Ajouter une résolvante est un pas de résolution. Le processus s'arrête, si la clause vide est ajoutée.
Cela signifie que l'ensemble des clauses n'est pas satisfaisable.

Exemple 3.14

Soit l'ensemble de clauses suivantes :

$S_0 = \{\{P, Q\}, \{P, \sim Q\}, \{\sim P, Q], \{\sim P, \sim Q\}, \{R, S\}\}$
$C_0 = \{P\}$
$S_1 = S_0 \cup \{\{P\}\}$
$C_1 = \{Q\}$
$S_2 = S_1 \cup \{\{Q\}\}$
$C_2 = \{\sim P\}$
$S_3 = S_2 \cup \{\{\sim P\}\}$
$C_3 = \{\}$
$S_4 = S_3 \cup \{\{\}\}$

► **Théorème 3.16** Correction et Complétude de la méthode de Résolution

Soit S un ensemble fini de clauses. S n'est pas satisfaisable ssi il existe une suite finie $S_0, \ldots, S_m$ d'ensembles de clauses telles que $S_0 = S$ et $S_{i+1} = S_i \cup C_i$ où C_i est une résolvante de S_i et $\| \in S_m$.

preuve

Soit une suite fine $S_0 \ldots S_m$ telle que $\| \in S_m$ Alors S_i est satisfaisable ssi S_{i+1} l'est. Donc S_0 n'est pas satisfaisable.
Réciproquement, supposons que S soit non satisfaisable. setcountercas0
Cas 4
[$\| \in S$] Dans ce cas il suffit de prendre m = 0.
Cas 5
[$\| \notin S$ et $\{P\} \in S$ et $\{\sim P\} \in S$] Par résolution, $\| \in S \cup \{\|\}$
◇
Cas 6
[$\| \notin S$ et $\{P, \ldots\} \in S$ et $\{\sim P, \ldots\} \in S$] Par résolution, on ajoute la résolvante de ces deux clauses à S. On note que la résolvante a déjà été calculée pour ces deux clauses. Si on suppose que toutes les clauses de numéros maxima ont été résoulues avec toutes les

autres, on applique une hypothèse de récurrence pour ce critère et on aboutit à notre résultat. $\diamond$

Cas 7
[$\| \notin S$ et aucune situation comme ci-dessus]

Dans ce cas, les variables propositionnelles figurent dans toutes les clauses mais sous une seule forme littérale. Soient $l_1, l_2 \ldots l_p$ les littéraux apparaissant dans la formule. Pour satisfaire cette formule, il suffit de considérer la valuation suivante :
Si l_i est $\sim P_i$, alors $v(P_i) = 0$
Si l_i est P_i, alors $v(P_i) = 1$.

Donc v satisfait S. Ce qui est absurde.
On en déduit que, si S est non satisfaisable, la règle de résolution est applicable et ce cas ne peut être applicable ici. Par induction, on aboutit à la clause vide. La taille minimal des clauses décroit strictement. $\diamond$

fin de la preuve

3.2 Forme Clausale

☼**Définition 3.9**

Un littéral est une formule atomique ou la négation d'une formule littérale.
Une clause est une disjonction finie de littéraux.
Une formule φ est dite sous forme clausale, si φ est de la forme $\forall x_1 \, \forall x_2 \ldots \, \forall x_n [C_1 \wedge \ldots \wedge C_p]$
Une formule φ est dite sous forme normale prénexe, si φ est de la forme $Q_1 x_1 \ldots \, Q_p x_p \, [C_1 \wedge \ldots \wedge C_q]$ où Q_i est $\forall$ ou $\exists$ et C_j est une clause.

Lemme 3.3.2. 2

Toute formule φ peut être transformée en une formuleψ équivalente en forme normale prénexe.

Preuve Il suffit de raisonner par induction sur la complexité de la formule et d'appliquer les transformations suivantes :

1. $(Qx.F(x)) \vee G \longrightarrow Qx(F(x) \vee G)$
2. $(Qx.F(x)) \wedge G \longrightarrow Qx(F(x) \wedge G)$
   si x n'a pas d'occurence libre dans G
3. $\sim (\forall x.F(x)) \longrightarrow \exists x. \sim F(x)$
4. $\sim (\exists x.F(x)) \longrightarrow \forall x. \sim F(x)$

fin de la preuve

Lemme 3.3.2. 3

Soit φ une formule en forme normale prénexe.
φ peut être transformée en une formule sous forme clausale, en éliminant les quantifications existencielles par introduction de fonctions de Skolem, on note φ^S cette formule.

Preuve Soit φ sous forme normale prénexe.
$Q_1 x_1 \ldots Q_n x_n \psi$
fin de la preuve

Cas 8

[: $Q_1 = \exists$] On remplace x_i dans ψ par une constante et on supprime $Q_i x_i$. $\diamond$

Cas 9

Soit $Q_i,\ i \in 2,\ldots,n$ et $Q_i = \exists$ et i le premier.
On intoduit la fonction $f(x_1,\ldots,x_{i-1}) = x_i$ et on supprime $Q_i x_i$ et on remplace x_i par le terme.
Après examen de gauche à droite de la formule, on obtient une formule universelle. $\diamond$

Exemple 3.15
$\forall x\ \exists y\ \exists z\ \ ((\sim P(x,y) \wedge Q(x,z)) \vee R(x,y,z))$
$\equiv$

$$\forall x\ \exists y\ \exists z \begin{pmatrix} (\sim P(x,y) \wedge R(x,y,z)) \\ \vee \\ (Q(x,y) \wedge R(x,y,z)) \end{pmatrix}$$

D'où :

$$\forall x. \begin{pmatrix} \sim P(x,f(x)) \wedge R(x,f(x),g(x)) \\ \vee \\ Q(x,f(x)) \wedge R(x,f(x),g(x)) \end{pmatrix}$$

► Théorème 3.17
Soit S un ensemble de clauses représentant une formule φ.
φ est non satisfaisable ssi S n'est pas satisfaisable.

Preuve Soit φ en forme prénexe normale :
φ est $Q_1 X_1 \ldots Q_n x_n\ \psi(x_1,\ldots,x_n)$.
Notons i le premier tel que Q_i est $\exists$.
La transformation conduit à la formule $\forall x_i \ldots \forall x_{i-1}\ Q_{i+1} X_{i+1} \ldots Q_n x_n . \psi(x_1,\ldots,x_{i-1}, f(x_1 - x_{i-1})\ldots)$

fin de la preuve

Supposons que φ soit non satisfaisable et que la formule transformée φ_1 le soit. Alors, il existe un modèle $\mathcal{M}_1$ pour φ_1. Pour tout $x_1 \ldots x_{i-1}$, il existe au moins un élément validant φ_1 ie $f(x_1 \ldots x_{i-1})$, pour $\mathcal{M}_1$. Donc φ est vraie pour $\mathcal{M}_1$, ce qui contredit l'hypothèse.

Réciproquement, si φ_1 est non satisfaisable, et, si φ est satisfaisable, on étend l'interprétation de φ à la fonction f.

fin de la preuve

☼Définition 3.10

Soit $\mathcal{L}$ un langage du premier ordre. On appelle interprétation de Herbrand ou modèle de Herbrand une interprétation ou un modèle $\mathcal{H}$ tel que :

1. le domaine de $\mathcal{H}$ est l'ensemble des termes de $\mathcal{L}$: $|\mathcal{H}| = T(\mathcal{L})$
2. chaque terme $f(t_1,\ldots,t_n)$ est interprété par $fT_1 \ldots t_n$.
3. l'interprétation des relations est libre.

Les relations ont une interprétation libre ce qui va être utile pour construire des modèles particuliers.

► **Théorème 3.18**
Soit Σ un ensemble de formules sans quantificateurs.
Σ admet un modèle ssi Σ admet un modèle de Herbrand.

Preuve Supposons que $\mathcal{M}$ soit un modèle de Σ et supposons que $\Sigma = \varphi_1, \ldots, \varphi_n, \ldots$
Soit $\mathcal{H}$ le modèle tel que $r_{\mathcal{H}}(t_1, ldots, t_n)$ vaut $r_{\mathcal{M}}(\delta(t_1), ldots, \delta(t_n)$ où δ est une valuation fixée choisie.
Nous montrons que $\mathcal{H}$ est un modèle de Σ.
Cas 1
soit φ une formule atomique de Σ.
φ s'écrit $r(t_1, ldots, t_n)$. Soit δ' une valuation de $\mathcal{H}$

La valuation δ peut être étendue à tous les termes de $T(\mathcal{L})$. Soit un terme t, la valeur de t en δ est parfaitement définie.
L'interprétation de φ en δ' est définie par :

$r(t_1, ldots, t_n)_{\mathcal{H},\delta'} = r_{\mathcal{H}}(\delta'(t_1), \ldots, \delta'(t_n)) = r_{\mathcal{M}}(delta"(t_1), \ldots, \delta"(t_n))$
or $\mathcal{M}$ est un modèle de $r(t_1, \ldots, t_n)$.
Donc $r(t_1, \ldots, t_n)_{\mathcal{H},\delta'}$ est vraie pour tout δ' de V dans $T(\mathcal{L})$. $\diamond$

Cas 2

φ est une formule de Σ du type $\sim \psi(t_1, \ldots, t_n)$
Soit δ' une valuation de $\vee$ dans $T(\mathcal{L})$. Alors $\sim \psi(t_1, \ldots, t_n)_{\mathcal{H},\delta'}$ vaut $non \quad \psi(t_1, ldots, t_n)_{\mathcal{H},\delta'}$. Par induction, cela signifie que $\mathcal{H}$ est un modèle de φ. $\diamond$

Cas 3

φ est sous la forme $\varphi_1 \wedge \varphi_2$.
Dans ce cas, $\mathcal{H}$ est un modèle de φ ssi $\mathcal{H}$ est un modèle de φ_1 et φ_2 ssi $\mathcal{M}$ est un modèle de φ_1 et φ_2. $\diamond$

fin de la preuve

Remarque

1. Nous avons défini une interprétation de Herbrand pour $\mathcal{M}$ en fixant δ mais il est clair que chaque δ définit une interprétation de Herbrand associée à $\mathcal{M}$.
2. Soit ε un ensemble de formules en formes prénexes universelles. ε admet un modèle ssi ε admet un modèle de Herbrand.
3. Soit ε un ensemble de formules prénexes closes existencielles. ε admet un modèle ssi tout modèle de Herbrand est un modèle de ε.

► **Théorème 3.19**
Un ensemble ε de clauses est non satisfaisable ssi ε est faux pour toutes les interprétations d'Herbrand de ε

Preuve Supposons que ε est faux pour toutes les interprétations d'Herbrand de ε. Supposons que ε est satisfaisable.
Soit $\mathcal{M}$ un modèle pour ε. On peut construire un modèle de Herbrand à partir de $\mathcal{M}$ tel que ε soit satisfait par ce modèle, ce qui est contraire à l'hypothèse.
fin de la preuve

Ainsi, on réduit le champp d'exploration de la non-satisfaisabilité d'un ensemble de clauses aux modèles de Herbrand.

☼**Définition 3.11**

(a) Soit ε un ensemble de clauses.
Soit $\mathcal{A}$ l'ensemble des atomes clos de ε.
Un arbre sémantique pour ε est un arbre dont chaque lien entre deux noeuds est pourvu d'un ensemble fini d'atomes clos ou de négations d'atomes clos de $\mathcal{A}$ comme suit :

1. Soit N un noeud quelconque. N a un nombre fini de fils $L_1 \dots L_p$. Si Q_i est la conjonction des littéraux associés à L_i, alors $Q_1 \vee \dots \vee Q_p$ est une formule propositionnelle valide.
2. Pour chaque noeud N, l'union des ensembles associés au chemin de la racine à N ne contient pas de littéraux complémentaires. On notera $I(N)$ cet ensemble.

(b) Un arbre sémantique pour ε est complet, si, pour tout noeud terminal N, $I(N)$ contient A_i ou $\sim A_i$, pour tout atome fermé.

Exemple 3.16

$\mathcal{S} = P(x), Q(f(x))$
$\mathcal{A} = P(a), P(f(a)), \dots, Q(a), Q(f(a)), \dots$

Exemple 3.17

$\mathcal{A} = P, Q, R$

Un noeud N est un noeud de réfutation, si $I(N)$ réfute une instance d'une cause dans Σ mais $I(N')$ ne réfute aucune instance de clause dans Σ où N' est un ancêtre de N.
Un arbre sémantique T est fermé ssi chaque branche de T se termine à un noeud de réfutation.

Exemple 3.18

$\Sigma = P(x), \sim P(x) \vee Q(f(x)), \sim Q(f(a))$
$\mathcal{A} = P(a), Q(a), P(f(a)), Q(f(a)), \dots$
? est un arbre fermé

► **Théorème 3.20** Théorème de Herbrand
Un ensemble de clauses ε est contradictoire ou non satisfaisable ssi, à tout arbre sémantique complet de ε, correspond un arbre sémantique fini fermé.

PREUVE:Supposons que ε soit contradictoire. Soit T un arbre sémantique complet de ε. Soit B une branche, on note J_B l'ensemble des littéraux de B. I_B est une interprétation de ε qui est contradictoire. Puisque ε est contradictoire, I_B réfute une instance close de C, notée C', dans ε. Puisque c'est fini, il y a un noeud de réfutation sur le chemin N_B. Puisque chaque branche de T a un noeud de réfutation, il y a un arbre sémantique fermé pour ε.
□

Le théorème de Herbrand est appliqué dans la méthode de Davis et Putnam qui est l'ancêtre de la résolution.

3.3 Méthode de Résolution de Robinson

Le principe de résolution du calcul propositionnel peut être étendu à la logique du 1er ordre.

☼**Définition 3.12** S

oient C_1 et C_2 deux clauses sans variables communes. Soient l_1 et l_2 deux littéraux de C_1 et C_2, respectivement. Si l_1 et $\sim l_2$ ont un pgu, σ, alors la clause $(\sigma(C_1)-\sigma(l_1))\cup(\sigma(C_2)-\sigma(l_2))$ est résolvante de C_1 et C_2.
On notera $C_1, C_2 \models$ résolvante (C_1, C_2, σ)

► **Théorème 3.21** Complétude et correction de la méthode de résolution

La méthode de résolution est correcte et complète :

1. Soit C un ensemble de clauses.

   Soit C_1 et C_2 deux clauses de C et C_3 est une résolvante de C_1 et C_2.
   Alors $C \models C_3$
2. C est contradictoire ssi la clause vide peut être obtenue par application de la règle de résolution.

Exemple 3.19 Exemple d'utilisation

$\varphi_1 \ \equiv\ \forall x\ (C(x) \Rightarrow (W(x) \wedge R(x)))$
$\varphi_2 \ \equiv\ \exists x\ (C(x) \wedge O(x))$
$\varphi_3 \ \equiv\ \exists x\ (O(x) \wedge R(x))$

Montrons que $\varphi_1, \varphi_2 \models \varphi_3$. Pour cela, on transforme $\varphi_1 \wedge \varphi_2 \wedge \sim \varphi_3$ comme suit :

1. $\sim C(x) \vee W(x)$
2. $\sim C(x) \vee R(x)$
3. $C(a)$
4. $O(a)$
5. $\sim O(x) \vee \sim R(x)$
   —————————- résolution sur (3) et (2)
6. $R(a)\ \sigma = x \longrightarrow a$
   —————————- résolution sur (4) et (5)
7. $\sim R(a)$
   —————————- résolution sur (6) et (7)
8. $[\,]$

Nous examinons la programmation logique et ses fondements. En fait, nous allons énoncer le principe de SLD résolution sur les clauses de HORN.

☼**Définition 3.13**

Une clause de HORN C est une des expressions suivantes :

1. $P(t_1, \ldots, t_n) \longrightarrow Q_1(t_1^1, \ldots, t_{n_1}^1), \ldots, Q_p(t_1^p, \ldots, t_{n_p}^p)$
2. $P(t_1, \ldots, t_n) \longrightarrow$
3. $\longrightarrow Q_1(t_1^1, \ldots, t_n^1), \ldots, Q_p(t_1^p, \ldots, t_1^p, t_{n_p}^p)$

où $R, Q_1, \ldots, Q_p$ sont des symboles de relation et $t_1, \ldots, t_n, t_k^j$ sont des termes.

Les clauses de HORN de type (1) et (2) sont des clauses définies. La clause (3) est une clause but (ou négative). Une expression de la forme $\alpha \longrightarrow \beta_1, \ldots, \beta_p$ s'interprète par "$\beta_1 \wedge \ldots \wedge \beta_p \Rightarrow \alpha$". Le symbole $\longrightarrow$ est ici le symbole de réécriture et on le remplace aussi par :-. Une clause de HORN est universellement close.

☼**Définition 3.14**

Soit $\mathcal{P}$ un ensemble de clauses de HORN. Notons $\mathcal{D}$ l'ensemble des clauses définies et $\mathcal{G}$ l'ensemble des clauses but de $\mathcal{P}$.
On note $\mathcal{G} = G_1, \ldots, G_p$. Une dérivation SLD est une suite finie de clauses négatives $N_0, N_1, \ldots, N_t$ satisfaisant :

1. $N_0 = G_i$ où $i \in \{1, \ldots, p\}$
2. Soit $k \in \{1, \ldots, t{-}1\}$ et N_k la clause "$\longrightarrow Q_1, Q_2, \ldots, Q_n$". Il existe une clause définie $C_k (\equiv P \longrightarrow R_1, \ldots, R_m)$ dans $\mathcal{D}$ telle que
   (a) P et Q_l sont unifiables, $l \in \{1, \ldots, n\}$
   (b) soit σ_k un pgu de Q_l et $\rho_k(P)$ où ρ_k renomme les variables de P, afin de séparer les variables de Q_l et celles de P.
   si m>0, alors N_{k+1} est la clause "$\longrightarrow \sigma_k(Q_1, \ldots, Q_{l-1}, \rho_k(R_1), \ldots, \rho_k(R_m), Q_{l+1}, \ldots, Q_n)$"
   si m=0, alors N_{k+1} est la clause "$\longrightarrow \sigma_k(Q_1, \ldots, Q_{l-1}, Q_{l+1}, \ldots, Q_n)$"

Exemple 3.20

$A(x, o, x) \longrightarrow$
$A(x, s(y), s(z)) \longrightarrow A(x, y, z) \longrightarrow A(s(o), u, s^2(o))$
$N_0 \equiv\sim A(s(o), u, s^2(o))$
$\sigma_0 = (x \longrightarrow s(o), u \longrightarrow s(y), z \longrightarrow s(o))$
$l_o = Id$

$N_1 \equiv\sim A(s(o), y, s(o))$
$\sigma_i = (x \longrightarrow s(o), y \longrightarrow o)$
$N_2 \equiv \llbracket$

Appliquons les deux substitutions :

$x \longrightarrow s(o)$
$u \longrightarrow s(y) \longrightarrow s(o)$
$z \longrightarrow s(o)$

L'exécution d'un programme prolog revient à appliquer la méthode de résolution jusqu'à trouver la clause vide. Cependant, la stratégie de Prolog est linéaire, sur les symboles de tête, en profondeur d'abord, avec retour arrière sur les échecs et exhaustive. L'exploration de l'arbre pose des problèmes de non terminaison.

BIBLIOGRAPHIE

[1] J. Van Benthem. *The logic of time*. Reidel, Dordrecht, 1983.

[2] J. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Internat. Congr. Logic, Method and Philos. Sci.*, pages 1–12, Stanford, 1960. Stanford University Press.

[3] G. Gentzen. *Untersuchungen Uber das Logische Schliessen ou Recherches sur la déduction loqique*. Presses Universitaires de France, 1955. Traduction de Feys et Ladrière.

[4] G. Gentzen. *Untersuchungen Uber das Logische Schliessen ou Recherches sur la déduction loqique*. Presses Universitaires de France, 1955. Traduction de Feys et Ladrière.

[5] J. A. W. Kamp. *Tense logic and the theory of linear order*. PhD thesis, UCLA, 1968.

[6] A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.

[7] N. Rescher and A. Urquart. *Temporal Logic*. Springer-Verlag, Berlin, 1971.

[8] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.

[9] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.