CHAPITRE 1

LOGIQUE (COURS - PARIS6 - IRÈNE GUESSARIAN)

Nous définissons le calcul propositionnel et le calcul des prédicats, leur syntaxe et leur sémantique, et des systèmes de preuve valides et complets pour chacun d'eux. Nous illustrons le calcul des prédicats en montrant que les modèles de Herbrand suffisent à caractériser la satisfaisabilité des clauses de Horn ; c'est la base de la sémantique de langages tels que PROLOG et DATALOG.

1.1 Remarques sur le raisonnement mathématique

Une *proposition* est une assertion qui est soit vraie, soit fausse, mais pas les deux : par exemple "2+2=5" est une proposition fausse, "$p \implies p$" est une proposition vraie. Par contre, une *formule* exprime une propriété d'un objet ou une relation entre objets, et elle prend une valeur vrai ou faux après attribution d'une valeur aux objets : par exemple $2 + 2 = x$ prend la valeur vrai si on attribue la valeur 4 à x, et prend la valeur faux pour toute autre évaluation de x, "$p \Longrightarrow q$" peut prendre les valeurs vrai ou faux selon les valeurs attribuées à p et q. Un *théorème* est une formule qui est toujours vraie.

Soient p et q deux propositions concernant les mêmes objets : on dit que p implique q, et on note "$p \Longrightarrow q$" si toutes les fois que p est vrai, q est aussi vrai ; $p \Longrightarrow q$ est un théorème, dont p est l'hypothèse et q la conclusion ; la réciproque de $p \Longrightarrow q$ est $q \Longrightarrow p$, qui n'est pas nécessairement un théorème (cf. exercice 1.1).

a) Si $p \Longrightarrow q$ et l'implication contraposée $\neg q \Longrightarrow \neg p$ sont bien deux manières différentes d'affirmer la même chose, l'implication réciproque $q \implies p$ affirme en général une chose totalement différente.

b) Si $\neg q \Longrightarrow \neg p$ est vraie, et q est vraie, p n'est pas nécessairement vraie.

c) Si $p \Longrightarrow q$ est faux, cela n'implique en général pas que la réciproque $q \Longrightarrow p$ soit vraie. A rapprocher du fait que $A \not\supseteq B$ n'implique en général pas que $A \subseteq B$.

EXERCICE 1.1 1) Soient

$$p = \text{“il pleut”}$$
$$q = \text{“il y a des nuages”}$$

Ecrire l'implication $p \implies q$ ainsi que sa contraposée, sa réciproque et la contraposée de sa réciproque. Quelles implications sont vraies ?

2) On considère les formules

$$p = \big(\forall x \in A\ ,\ \exists y \in B\ ,\ P(x,y)\big)$$
$$\text{et} \qquad q = \big(\exists y \in B\ ,\ \forall x \in A\ ,\ P(x,y)\big)$$

où :

- A est l'ensemble des hommes,
- B est l'ensemble des femmes,
- $P(x, y)$ signifie "y aime x".

Que peut-on dire de $p \Longrightarrow q$? De sa réciproque ? $\diamondsuit$

Les assertions en langage courant A *si* B et A *seulement si* B correspondent respectivement à l'implication $B \Longrightarrow A$ et à sa réciproque $A \Longrightarrow B$; par exemple, on dira "il y a des nuages *si* il pleut" ("il pleut" $\Longrightarrow$ "il y a des nuages") et "il pleut *seulement si* il y a des nuages" ("il pleut" $\Longrightarrow$ "il y a des nuages").

1.2 Calcul propositionnel

Dans toute la suite de ce chapitre, la logique et son langage seront l'objet de notre étude ; toutefois la logique est omniprésente en mathématique en tant qu'outil de démonstration, et donc en tant qu'élément du méta-langage ; nous essaierons donc dans le présent chapitre de distinguer les symboles du langage logique considéré en tant qu'objet de l'étude des symboles du langage logique considéré en tant qu'outil du méta-langage. Nous noterons $\supset$ l'implication, considérée comme symbole formel du langage logique, et $\Longrightarrow$ l'implication, considérée comme outil du méta-langage logique.

L'un des buts fondamentaux de la logique est de faire des démonstrations correctes. Pour atteindre ce but, les concepts de conséquences sont essentiels : quand peut-on affirmer, sans erreur possible, qu'une formule est conséquence d'un ensemble de formules ? Nous définirons dans la Section 1.2.2 une notion de conséquence sémantique, puis dans la Section 1.2.3 une notion de prouvabilité ou conséquence syntaxique, dont on peut montrer qu'elles coïncident.

1.2.1 La syntaxe : les formules

Soit $P = \{p, p', q, q', \ldots\}$ un ensemble de symboles propositionnels et soient les symboles $\supset$ et $\neg$ ainsi que les parenthèses ouvrante et fermante.

Définition 1.1 *Une formule propositionnelle est une suite de symboles pris dans* $P \cup \{\supset, \neg, (,)\}$ *et construite selon les règles suivantes :*

1) *tout symbole propositionnel de* P *est une formule,*
2) *si* F *est une formule,* $\neg F$ *est une formule,*
3) *si* F *et* F' *sont deux formules,* $(F \supset F')$ *est une formule,*
4) *toute formule est obtenue par application répétée, un nombre fini de fois, des étapes 1) à 3) ci-dessus.*

Remarquons que la définition 1.1 est un exemple de définition récursive ou inductive d'ensembles.

Les formules sont des suites de symboles. Elles sont dénuées de sens pour l'instant. L'attribution d'un sens, c'est-à-dire une valeur "vrai" ou "faux" à une formule, constitue la sémantique de la formule et fera l'objet de la Section 1.2.2.

Soit σ une application de l'ensemble des symboles propositionnels P dans l'ensemble des formules, appelée substitution. La formule $\sigma(F)$ obtenue par substitution dans la formule F est définie par

- si $F = p$ avec $p \in P$, $\sigma(F) = \sigma(p)$
- si $F = \neg F'$, $\sigma(F) = \neg\sigma(F')$
- si $F = (F_1 \supset F_2)$, $\sigma(F) = (\sigma(F_1) \supset \sigma(F_2))$

Exemple 1.2 Soit la substitution σ définie par $\sigma(p) = q$ et $\sigma(q) = (p \supset q)$. Alors $\sigma(p \supset q) = (q \supset (p \supset q))$.

Définition 1.3 *Un séquent est un couple $(\mathcal{F}, F)$ où $\mathcal{F}$ est un ensemble fini de formules et F une formule.*

L'intuition est qu'un séquent est censé formaliser la notion de conséquence logique : si toutes les prémisses du séquent, c'est-à-dire les formules de $\mathcal{F}$ sont vraies, alors la conclusion du séquent, c'est-à-dire la formule F, est vraie.

1.2.2 La sémantique : interprétation d'une formule

Soit une formule F et I une application de l'ensemble des symboles propositionnels dans l'algèbre de Boole $\mathbb{B} = \{1, 0\}$, munie de ses opérations $+$, "." dénotant le produit, et $\bar{\ }$; les constantes booléennes "vrai", "faux", identifiées ici à 1 et 0 sont aussi notées parfois *tt*, *ff* ou V, F.

On définit la *valeur de vérité* $I(F)$ de F pour I par

- si $F = p \in P$, $I(F) = I(p) \in \mathbb{B}$;
- si $F = \neg F'$, $I(F) = \overline{I(F')}$;
- si $F = (F_1 \supset F_2)$, $I(F) = \overline{I(F_1)} + I(F_2)$.

Si $I(F) = 1$, on dit que F est vraie dans I. I est appelée une *interprétation*, et on dit aussi que $I(F)$ est l'interprétation de la formule F.

Exercice 1.2 Définissons $F \wedge F' \stackrel{\text{def}}{=} \neg(F \supset \neg F')$ et $F \vee F' \stackrel{\text{def}}{=} \neg F \supset F'$.

1) Ecrire les tables donnant les valeurs de vérité de $\wedge, \vee, \supset$. En déduire que

$$I(F \wedge F') = I(F).I(F') \quad \text{et} \quad I(F \vee F') = I(F) + I(F')\,.$$

2) Montrer que

$$\begin{aligned} I\big(F_n \supset (F_{n-1} \supset (\cdots(F_1 \supset F)\cdots))\big) &= \overline{I(F_n)} + \overline{I(F_{n-1})} + \cdots + \overline{I(F_1)} + I(F) \\ &= I\big(\neg F_n \vee (\neg F_{n-1} \vee (\cdots \vee (\neg F_1 \vee F)\cdots))\big) \end{aligned}$$

En déduire que

$$I\big(F_n \supset (F_{n-1} \supset (\cdots(F_1 \supset F)\cdots))\big) = I\big((F_n \wedge (F_{n-1} \wedge (\cdots \wedge (F_2 \wedge F_1))\cdots)) \supset F\big). \qquad \diamondsuit$$

Définition 1.4 *Soit F une formule. On dit que*

- *F est valide, ou que F est une tautologie, si pour toute I, $I(F) = 1$,*
- *F est satisfaisable s'il existe une I telle que $I(F) = 1$,*

– *F est insatisfaisable si pour toute I, $I(F) = 0$.*

EXEMPLE 1.5 $p \wedge \neg p$ est insatisfaisable ; $p \vee \neg p$ est une tautologie ; $p \wedge (p \supset q)$ est satisfaisable mais n'est pas valide.

EXERCICE 1.3 Vérifier que F est insatisfaisable si et seulement si $\neg F$ est valide. ♢

Définition 1.6 *Un séquent $(\mathcal{F}, G)$ est vrai dans I si*

$$\big(\text{pour toute } F \in \mathcal{F}, I(F) = 1\big) \Longrightarrow I(G) = 1,$$

ou en d'autres termes si

$$\Big(F \in \mathcal{F} \Longrightarrow I(F) = 1\Big) \Longrightarrow I(G) = 1.$$

Un séquent $(\mathcal{F}, G)$ est valide si : pour toute I, (pour toute $F \in \mathcal{F}, I(F) = 1) \Longrightarrow I(G) = 1$. On écrira $\mathcal{F} \models G$ pour exprimer que le séquent $(\mathcal{F}, G)$ est valide.

Cette définition formalise la notion de *conséquence sémantique*.

EXERCICE 1.4 1) Montrer qu'une formule G est vraie dans I (resp. valide) si le séquent $(\emptyset, G)$ est vrai dans I (resp. valide).

2) Les séquents suivants sont-ils valides ?

– $(\emptyset, (p \supset q))$,
– $(\{p, (p \supset q)\}, q)$. ♢

Proposition 1.7 *Si σ est une substitution, et si $\mathcal{F} \models G$ est un séquent valide, alors $\sigma(\mathcal{F}) \models \sigma(G)$ est un séquent valide.*

Démonstration. Si I est une interprétation, on définit l'interprétation I_σ par $I_\sigma(p) = I(\sigma(p))$. On en déduit que $I_\sigma(F) = I(\sigma(F))$.

Soit une interprétation I telle que pour toute $F' \in \sigma(\mathcal{F}), I(F') = 1$. On a donc pour toute $F \in \mathcal{F}, I(\sigma(F)) = I_\sigma(F) = 1$. Comme $\mathcal{F} \models G$, $I_\sigma(G) = I(\sigma(G)) = 1$. □

Proposition 1.8 *(Théorème de la déduction) $\{F_n, \ldots, F_1\} \models F$ si et seulement si*

$$\emptyset \models F_n \supset (F_{n-1} \supset (\cdots (F_1 \supset F) \cdots)).$$

Démonstration. Il suffit de montrer que $\mathcal{F} \cup \{F\} \models G$ si et seulement si $\mathcal{F} \models (F \supset G)$, et le résultat s'ensuivra par récurrence sur n. Or, on a $I(F \supset G) = 1$ si et seulement si $I(F) = 1 \Longrightarrow I(G) = 1$.

Soit une interprétation I. On a

$$(\text{pour toute } F' \in \mathcal{F}, I(F') = 1) \text{ et } I(F) = 1 \Longrightarrow I(G) = 1$$

si et seulement si (pour toute $F' \in \mathcal{F}, I(F') = 1) \Longrightarrow I(F \supset G) = 1$. □

On écrit en général $\mathcal{F}, F \models G$ au lieu de $\mathcal{F} \cup \{F\} \models G$.

Exercice 1.5 On peut associer une formule $\phi((\mathcal{F},G))$ à un séquent $S=(\mathcal{F},G)$ en posant

- si $S=(\emptyset,G)$, alors $\phi(S)=G$,
- si $S=(\{F\}\cup\mathcal{F},G)$, et $\phi((\mathcal{F},G))=F'$, alors $\phi(S)=(F\supset F')$.

Montrer que le séquent $S=(\mathcal{F},G)$ est vrai dans I si et seulement si $\phi(S)$ est vraie dans I. ♢

Proposition 1.9 *1) $\mathcal{F}\models F$ si et seulement si $\mathcal{F}\models\neg\neg F$; $\mathcal{F},F\models G$ si et seulement si $\mathcal{F},\neg\neg F\models G$;*

2) si $\mathcal{F}\models\neg(F\supset F')$ alors $\mathcal{F}\models F$ et $\mathcal{F}\models\neg F'$;

3) si $\mathcal{F},(F\supset F')\models G$ alors $\mathcal{F},F'\models G$, , $\mathcal{F},\neg G\models\neg F'$, et $\mathcal{F},\neg G\models F$;

4) si $\mathcal{F},\neg(F\supset F')\models G$ alors $\mathcal{F},\neg G,F\models F'$.

Démonstration. 1) Il est évident que $I(F)=I(\neg\neg F)$.

2) Soit I telle que pour toute $G\in\mathcal{F}$, $I(G)=1$ alors $I(F\supset F')=0$ et donc $I(F)=1$ et $I(F')=0$, donc $I(\neg F')=1$.

3) Soit I telle que pour toute $H\in\mathcal{F}$, $I(H)=1$. Si $I(F')=1$ alors $I(F\supset F')=1$ et donc $I(G)=1$. Si $I(G)=0$, alors $I(F\supset F')=0$ d'où $I(F')=0$, et $I(F)=1$.

4) Soit I telle que pour toute $H\in\mathcal{F}$, $I(H)=1$. Si $I(G)=0$ et $I(F)=1$ on a aussi $I(F')=1$ car sinon $I(F\supset F')=0; I(\neg(F\supset F'))=1$ et $I(G)=1$; contradiction. □

1.2.3 Démonstrations logiques

Définition 1.10 *Un séquent $(\mathcal{F},F)$ est dit prouvable, ce qui est noté $\mathcal{F}\vdash F$, s'il est construit en utilisant un nombre fini de fois les règles suivantes*

- *utilisation d'une hypothèse : $F\in\mathcal{F}\Longrightarrow\mathcal{F}\vdash F$*
- *augmentation des hypothèses : si $G\notin\mathcal{F}$ et $\mathcal{F}\vdash F$ alors $\mathcal{F}\cup\{G\}\vdash F$*
- *règle de détachement (ou modus ponens) : si $\mathcal{F}\vdash(F\supset F')$ et si $\mathcal{F}\vdash F$ alors $\mathcal{F}\vdash F'$*
- *règle de synthèse (ou retrait d'une hypothèse) : si $\mathcal{F},F\vdash F'$ alors $\mathcal{F}\vdash(F\supset F')$*
- *règle de la double négation : $\mathcal{F}\vdash F$ si et seulement si $\mathcal{F}\vdash\neg\neg F$*
- *règle du raisonnement par l'absurde : si $\mathcal{F},F\vdash F'$ et $\mathcal{F},F\vdash\neg F'$, alors $\mathcal{F}\vdash\neg F$.*

Cette définition formalise la notion de *conséquence logique*.

Z Les séquents prouvables sont caractérisés uniquement par des manipulations de suites de symboles, par opposition aux séquents valides qui sont caractérisés par leur interprétation.

Une *démonstration* d'un séquent prouvable $\mathcal{F}\vdash F$ est une suite finie de séquents prouvables $\mathcal{F}_i\vdash F_i$, dont le dernier est $\mathcal{F}\vdash F$, telle que tout séquent de cette suite est obtenu en appliquant les règles ci-dessus à des séquents qui le précèdent dans la suite. Le premier séquent d'une démonstration est donc nécessairement obtenu par la règle d'utilisation d'une hypothèse.

Exemple 1.11 Les accolades sont omises pour les séquents donnés explicitement.

1)	$p \vdash p$	(utilisation d'une hypothèse)
	$\emptyset \vdash (p \supset p)$	(synthèse)
2)	$p, q \vdash p$	(hypothèse)
	$p \vdash (q \supset p)$	(synthèse)
	$\emptyset \vdash \big(p \supset (q \supset p)\big)$	(synthèse)
3)	1. $(p \supset q), \neg q, p \vdash \neg q$	(hypothèse)
	2. $(p \supset q), \neg q, p \vdash p$	(hypothèse)
	3. $(p \supset q), \neg q, p \vdash p \supset q$	(hypothèse)
	4. $(p \supset q), \neg q, p \vdash q$	(modus ponens sur 2 et 3)
	5. $(p \supset q), \neg q \vdash \neg p$	(contradiction en 1 et 4)
	6. $(p \supset q) \vdash (\neg q \supset \neg p)$	(synthèse)
4)	$p, \neg p, \neg q \vdash p$	(hypothèse)
	$p, \neg p, \neg q \vdash \neg p$	(hypothèse)
	$p, \neg p \vdash \neg\neg q$	(contradiction)
	$p, \neg p \vdash q$	(double négation)
	$p \vdash (\neg p \supset q)$	(synthèse)
5)	$p \vdash p$	(hypothèse)
	$\emptyset \vdash (p \supset p)$	(synthèse)
	$p \vdash (p \supset p)$	(augmentation)
	$\emptyset \vdash p \supset (p \supset p)$	(synthèse)

Théorème 1.12 *1. Tout séquent prouvable est valide.*

2. Tout séquent valide est prouvable.

L'assertion 1. du théorème 1.12 est le théorème qui montre la cohérence (ou validité, ou correction) de notre système de démonstrations logiques ; elle est facile à vérifier. L'assertion 2. du théorème 1.12 est le théorème qui montre la complétude du système de démonstrations logiques. La complétude est plus difficile à démontrer.

Le théorème 1.12 nous fournit une première méthode pour montrer qu'une formule F est valide : il suffit de montrer que le séquent $\vdash F$ est prouvable.

Corollaire 1.13 *F est valide si et seulement si $\vdash F$ est prouvable.*

1.2.4 Autres connecteurs logiques

Nous pouvons aussi utiliser en logique propositionnelle les connecteurs $\wedge$ (et) et $\vee$ (ou).

Les formules se définissent alors par la règle supplémentaire : si F et F' sont des formules alors $(F \wedge F')$ et $(F \vee F')$ sont des formules.

L'interprétation de ces formules est définie en ajoutant, cf. exercice 1.2,

$I(F \wedge F') = I(F)\ .\ I(F')$

$I(F \vee F') = I(F) + I(F')$

de sorte que

$I(F \wedge F') = I(\neg(F \supset \neg F'))$

$I(F \vee F') = I(\neg F \supset F')$.

De même, nous étendons les définitions des séquents prouvables en ajoutant les règles :

- Si $\mathcal{F} \vdash F$ et $\mathcal{F} \vdash F'$ alors $\mathcal{F} \vdash (F \wedge F')$
- Si $\mathcal{F} \vdash (F \wedge F')$ alors $\mathcal{F} \vdash F$
- Si $\mathcal{F} \vdash (F \wedge F')$ alors $\mathcal{F} \vdash F'$
- Si $\mathcal{F}, G \vdash F$ et $\mathcal{F}, \neg G \vdash F'$ alors $\mathcal{F} \vdash (F \vee F')$
- Si $\mathcal{F}, F \vdash G$ et $\mathcal{F}, F' \vdash G$ alors $\mathcal{F}, (F \vee F') \vdash G$

De même, on peut introduire le symbole d'équivalence $\equiv$ dont l'interprétation est donnée par :

$$I(F \equiv F') = 1 \text{ si et seulement si } I(F) = I(F').$$

De sorte que $I(F \equiv F') = I\big((F \supset F') \wedge (F' \supset F)\big)$.

Les règles de démonstration associées au symbole d'équivalence $\equiv$ sont :

- Si $\mathcal{F} \vdash (F \equiv F')$ alors $\mathcal{F} \vdash (F \supset F')$ et $\mathcal{F} \vdash (F' \supset F)$,
- Si $\mathcal{F} \vdash (F \supset F')$ et $\mathcal{F} \vdash (F' \supset F)$ alors $\mathcal{F} \vdash (F \equiv F')$,

ou encore :

$$\mathcal{F} \vdash (F \equiv F') \qquad \text{si et seulement si} \qquad \mathcal{F} \vdash ((F \supset F') \wedge (F' \supset F))\,.$$

L'usage “méta-logique” du symbole $\Longleftrightarrow$ peut alors se formaliser par :

$F \Longleftrightarrow F'$ si et seulement si $(F \equiv F')$ est une formule valide, ou en d'autres termes si et seulement si $\vdash (F \equiv F')$.

Les opérations $\vee, \wedge$ ont des propriétés d'associativité, de commutativité, et de distributivité semblables à celles des algèbres de Boole :

1) Distributivité de $\wedge$ par rapport à $\vee$: $F \wedge (G \vee H) \iff (F \wedge G) \vee (F \wedge H)$.
2) Distributivité de $\vee$ par rapport à $\wedge$: $F \vee (G \wedge H) \iff (F \vee G) \wedge (F \vee H)$.
3) Associativité de $\wedge$: $F \wedge (G \wedge H) \iff (F \wedge G) \wedge H$.
4) Associativité de $\vee$: $F \vee (G \vee H) \iff (F \vee G) \vee H$.
5) Commutativité de $\wedge$: $F \wedge G \iff G \wedge F$.
6) Commutativité de $\vee$: $F \vee G \iff G \vee F$.

L'associativité nous permet d'omettre les parenthèses.

Les équivalences suivantes sont utiles à connaître.

$$\begin{aligned}
F \supset G &\iff \neg F \vee G \\
F \supset G &\iff \neg G \supset \neg F \\
\neg(F \supset G) &\iff F \wedge \neg G \\
F \equiv G &\iff (F \supset G) \wedge (G \supset F) \\
F \equiv G &\iff (F \wedge G) \vee (\neg G \wedge \neg F)
\end{aligned}$$

1.2.5 Tables de vérité

On peut prendre une notation abrégée, et représenter $\vee, \wedge, \neg$ respectivement par $+$, la concaténation (notée sans aucun signe comme la multiplication) et ; on a ainsi une notation algébrique, par exemple $(p \vee q) \wedge r$ s'écrit $(p+q)r$.

Une table de vérité pour une formule ayant n variables propositionnelles a une ligne pour chaque combinaison de valeurs de vérité des variables ; il y aura donc 2^n lignes s'il y a n variables propositionnelles. Par exemple, les connecteurs de base ont les tables suivantes :

x	y	$x+y$	xy	$\overline{x}$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

On évalue souvent une formule logique en commençant par ses sous-formules et en remontant, en définissant une colonne pour chaque sous-formule. Par exemple, la table de $\overline{x}y + \overline{y}$ est donnée par

x	y	$\overline{x}$	$\overline{x}y$	$\overline{y}$	$\overline{x}y+\overline{y}$
0	0	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	1	0	0	0	0

Avec la notation algébrique, les tautologies fondamentales du calcul propositionnel prennent la formes de propriétés algébriques :

1) commutativité : $(p+q) \equiv (q+p)$ et $pq \equiv qp$

2) associativité : $(p+q)+r \equiv p+(q+r)$ et $(pq)r \equiv p(qr)$

3) distributivité : $p(q+r) \equiv pq+pr$ et $p+qr \equiv (p+q)(p+r)$

4) idempotence : $pp \equiv p$ et $p+p \equiv p$

5) lois de Morgan ; $\overline{pq} \equiv \overline{p}+\overline{q}$ et $\overline{p+q} \equiv \overline{p}\,\overline{q}$

Les lois de distributivité ne se comportent pas comme ce qu'on attendrait de la part d'opérations notées comme addition et multiplication, mais 1,2 ont bien le comportement usuel de l'addition et la multiplication. Les lois de Morgan se généralisent aussi au cas infini : nous verrons que $\neg\forall x p(x) \equiv \exists x \neg p(x)$ et $\neg\exists x p(x) \equiv \forall x \neg p(x)$; or on peut considérer $\exists$ (resp. $\forall$) comme la disjonction (resp. conjonction) infinie.

1.2.6 Mise sous forme normale conjonctive

Un littéral est soit une variable propositionnelle, soit la négation d'une variable propositionnelle ; une *clause* est une disjonction de littéraux. Toute formule est équivalente à une *formule en forme normale conjonctive* ou *FNC*, c'est-à-dire une *conjonction de clauses.* Les règles suivantes permettent de trouver la FNC d'une formule :

$$\begin{array}{rcl} \neg\neg F & \Longleftrightarrow & F \\ \neg(F \vee G) & \Longleftrightarrow & (\neg F) \wedge (\neg G) \\ \neg(F \wedge G) & \Longleftrightarrow & (\neg F) \vee (\neg G) \\ F \vee (G \wedge H) & \Longleftrightarrow & (F \vee G) \wedge (F \vee H) \\ (G \wedge H) \vee F & \Longleftrightarrow & (G \vee F) \wedge (H \vee G) \end{array}$$

EXEMPLE 1.14 1. Si p et q sont des variables propositionnelles, la FNC de $p \vee q$ est $p \vee q$, la FNC de $p \vee \neg q$ est $p \vee \neg q$.

2. Si p , q et r sont des variables propositionnelles, la FNC de $\neg(p \wedge \neg(q \vee r))$ est $\neg p \vee q \vee r))$; la FNC de $\neg(p \wedge \neg(q \wedge r))$ est $(\neg p \vee q) \wedge (\neg p \vee q)$.

1.2.7 Systèmes déductifs : l'exemple du système de Hilbert-Ackermann

Pour définir les séquents prouvables nous avons défini des règles de manipulation de suites de symboles. Il existe d'autres systèmes de règles permettant d'obtenir le même résultat.

Auparavant nous disons qu'une formule F est *prouvable* si et seulement si $\emptyset \vdash F$ est un séquent prouvable. D'après les théorèmes précédents une formule est prouvable si et seulement si elle est valide.

Nous allons donner maintenant un exemple d'une autre façon de faire des démonstrations de formules ne contenant que les symboles propositionnels et les symboles $\vee$ et $\neg$ et qu'on appelle le *système de Hilbert–Ackermann*. $p \supset q$ est une abréviation de $\neg p \vee q$. On notera aussi de façon abrégée $\neg p$ par $\bar{p}$. Les formules que nous pourrons démontrer avec les règles seront appelées "théorèmes logiques" (pour les distinguer des formules prouvables).

Soient p, q, r trois symboles propositionnels quelconques.

(i) Les quatre formules suivantes, appelées *axiomes*, sont des théorèmes logiques.
  - $(p \vee p) \supset p$, (élimination du ou)
  - $p \supset (p \vee q)$, (introduction du ou)
  - $(p \vee q) \supset (q \vee p)$, (commutativité du ou)
  - $(p \supset q) \supset ((r \vee p) \supset (r \vee q))$, (pre-transitivité)

(ii) Si σ est une substitution et si F est un théorème logique, alors $\sigma(F)$ est un théorème logique.

(iii) Si F et $(F \supset F')$ sont des théorèmes logiques, alors F' est un théorème logique.

(ii) et (iii) sont les *règles d'inférence* ; (iii) est appelée la règle de *modus ponens*.

Une *déduction* de la formule F à partir de l'ensemble de formules $\mathcal{F}$ est une suite finie de formules $F_1, F_2, \ldots, F_n$ telles que

- F_n est identique à F
- pour tout $i \leq n$,
  - soit F_i est l'un des axiomes (i),
  - soit $F_i \in \mathcal{F}$,
  - soit F_i peut être déduite des F_j précédents par une application d'une des règles (ii) ou (iii).

La formule F est un théorème logique si et seulement s'il y a une déduction de F à partir de l'ensemble vide de formules $\mathcal{F} = \emptyset$.

Proposition 1.15 *On a les règles suivantes :*

(i) Si $F \vee F$ est un théorème, alors F est un théorème,

(ii) Si F est un théorème, alors $F \vee G$ est un théorème,

(iii) Si $F \vee G$ est un théorème, alors $G \vee F$ est un théorème,

(iv) Si $F \supset G$ est un théorème et H une formule, alors $(H \vee F) \supset (H \vee G)$ est un théorème,

(v) Si $F \supset G$ et $G \supset H$ sont des théorèmes, alors $F \supset H$ est un théorème.

Démonstration. Ces règles se démontrent immédiatement à partir des axiomes et de la règle de modus ponens. Par exemple, par substitution dans l'axiome d'élimination du ou on obtient le théorème $(F \vee F) \supset F$, et si $F \vee F$ est un théorème, alors le modus ponens implique que F est un théorème. Les règles (ii) à (iv) se prouvent de même. Pour la règle (v), on écrit la suite de théorèmes :

$(G \supset H) \supset ((F \vee G) \supset (F \vee H))$	(pre-transitivité+règle (ii))
$(G \supset H) \supset ((\bar{F} \vee G) \supset (\bar{F} \vee H))$	(règle (ii))
$(G \supset H) \supset ((F \supset G) \supset (F \supset H))$	(abréviation)
$(F \supset G) \supset (F \supset H)$	(modus ponens avec $(G \supset H)$)
$(F \supset H)$	(modus ponens avec $(F \supset G)$) □

Exemple 1.16 Montrons que $p \supset p$ est un théorème. On a

$p \supset p \vee p$	(introduction du ou +règle (ii))
$p \vee p \supset p$	(élimination du ou)
$p \supset p$	(règle (v))

Rappelons que par le calcul des séquents, cette preuve se réduit à :

$p \vdash p$	(utilisation d'une hypothèse)
$\emptyset \vdash (p \supset p)$	(synthèse)

Exemple 1.17 Par exemple, la preuve de l'axiome d'élimination du ou à l'aide du calcul des séquents procède comme suit :

$p \vdash p$	(utilisation d'une hypothèse)
$p \vdash p$	(utilisation d'une hypothèse)
$p \vee p \vdash p$	(règle $\mathcal{F}, F \vdash G$ et $\mathcal{F}, F' \vdash G$ alors $\mathcal{F}, (F \vee F') \vdash G$)
$\emptyset \vdash ((p \vee p) \supset p))$	(synthèse)

On peut, au vu de ces deux exemples, commencer à soupçonner que l'utilisation du calcul des séquents sera plus simple que celle du système de Hilbert-Ackermann (qui comporte moins de règles).

Il est facile de voir que tout théorème logique est une formule valide (et donc une formule prouvable) : on montre que chaque axiome est une formule valide, et que si l'on applique les règles (ii) et (iii) à des formules valides, on obtient des formules valides, et on en déduit par induction sur la longueur de la déduction que tout théorème est une formule valide ; ceci montre la cohérence du système pour déduire les théorèmes logiques. Inversement, nous pouvons montrer la complétude du système, *i.e.* que toute formule valide est un théorème logique. La complétude est plus difficile à démontrer.

EXERCICE 1.6 Montrer les théorèmes suivants :

1) (a) $\bar{p} \vee p$ et (b) $p \vee \bar{p}$
2) (a) $p \supset \bar{\bar{p}}$ et (b) $\bar{\bar{p}} \supset p$
3) $(p \supset q) \supset (\bar{q} \supset \bar{p})$
4) Si $p \supset q$ et $q \supset p$ sont des théorèmes, et $F(p)$ une formule ayant p comme sous-formule, alors $F(p) \supset F(q)$ et $F(q) \supset F(p)$ sont des théorèmes. ♢

EXERCICE 1.7 Montrer que si $F \supset (G \supset H)$ est un théorème, alors $G \supset (F \supset H)$ et $F \wedge G \supset H$ sont des théorèmes. ♢

En utilisant d'autres systèmes de règles, on pourrait définir d'autres ensembles de formules prouvables qui peuvent coïncider ou non avec les formules valides.

Les systèmes déductifs sont des systèmes de règles qui permettent de définir des ensembles de formules **inclus** dans l'ensemble des formules valides.

1.2.8 Les arbres sémantiques

Nous avons vu dans le corollaire 1.13 une première méthode pour vérifier qu'une formule F est valide : il suffit de vérifier que le séquent $\vdash F$ est prouvable. Une autre méthode pour montrer qu'une formule G est valide consiste à 1. calculer $F = \neg G$, 2. mettre cette formule F sous FNC et 3. utiliser la proposition suivante

Proposition 1.18 *Si F est en FNC, alors F est insatisfaisable si et seulement si F admet un arbre sémantique fermé.*

Un *arbre sémantique* est associé à une énumération $\{p_k\}$ des variables propositionnelles. Un *arbre sémantique* est un arbre binaire dont les arêtes sont étiquetées par des littéraux, et pour chaque nœud de niveau i, l'arête gauche partant de ce nœud est étiquetée par $\neg p_i$ et l'arête droite partant de ce nœud est étiquetée par p_i.

A chaque nœud n d'un arbre sémantique est associée une interprétation partielle I_n des variables propositionnelles : si n est un nœud de niveau i, c'est l'interprétation partielle définie en prenant la valeur 1 pour tous les littéraux qui étiquettent les arêtes menant de la racine au nœud n.

EXEMPLE 1.19 Considérons l'ensemble de clauses $\mathcal{F} = \{p \quad , \quad q \quad , \quad \neg p \vee q \quad , \quad \neg p \vee \neg q\}$. On trouvera ci-dessous un arbre sémantique associé à l'énumération $\{p, q\}$ et un arbre sémantique associé à l'énumération $\{q, p\}$. On remarquera que l'arbre sémantique peut avoir moins de nœuds fermés qu'il y a de clauses dans $\mathcal{F}$, lorsqu'un sous-ensemble strict de clauses de $\mathcal{F}$ est déjà insatisfaisable.

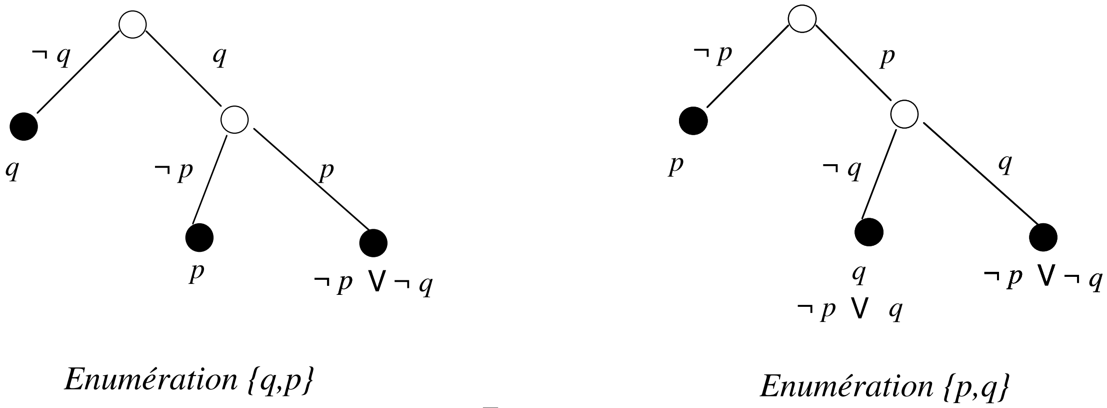

FIGURE 1.1

Définition 1.20 *Soit A un arbre sémantique et $\mathcal{F}$ un ensemble fini de clauses.*

(i) Un nœud n de l' arbre sémantique A réfute la clause C de $\mathcal{F}$ si $I_n(C) = 0$.

(ii) Un nœud n est un nœud d'échec pour $\mathcal{F}$ si :

- *il existe une clause C de $\mathcal{F}$ qui est réfutée par n,*
- *pour tout nœud n' situé sur le chemin menant de la racine de l'arbre à n, il n'existe aucune clause C' de $\mathcal{F}$ qui soit réfutée par n.*

(iii) Un arbre sémantique A est fermé pour $\mathcal{F}$ si, sur toute branche de A, il existe un nœud d'échec pour $\mathcal{F}$.

Exemple 1.21 Les nœuds fermés sont représentés par un point noir et les autres nœuds par un point blanc. Pour un arbre fermé, les nœuds de niveau supérieur à un nœud d'échec et se trouvant sur la même branche ne sont pas dessinés. Nous écrivons sous chaque nœud d'échec la (les) clause(s) qu'il réfute.

Considérons la formule $G = \big(p \supset (q \supset r)\big) \supset \Big(p \supset \big((p \supset q) \supset r\big)\Big)$. La FNC de $F = \neg G$ est l'ensemble $\mathcal{F} = \{p \supset (q \supset r) \quad , \quad p \supset q \quad , \quad p \quad , \quad \neg r\}$, qu'on peut aussi écrire sous la forme de clauses : $\mathcal{F} = \{\neg p \vee \neg q \vee r) \quad , \quad \neg p \vee q \quad , \quad p \quad , \quad \neg r\}$. On trouvera ci-dessous un arbre sémantique associé à l'énumération $\{r, p, q\}$, puis un arbre sémantique associé à l'énumération $\{q, p, r\}$.

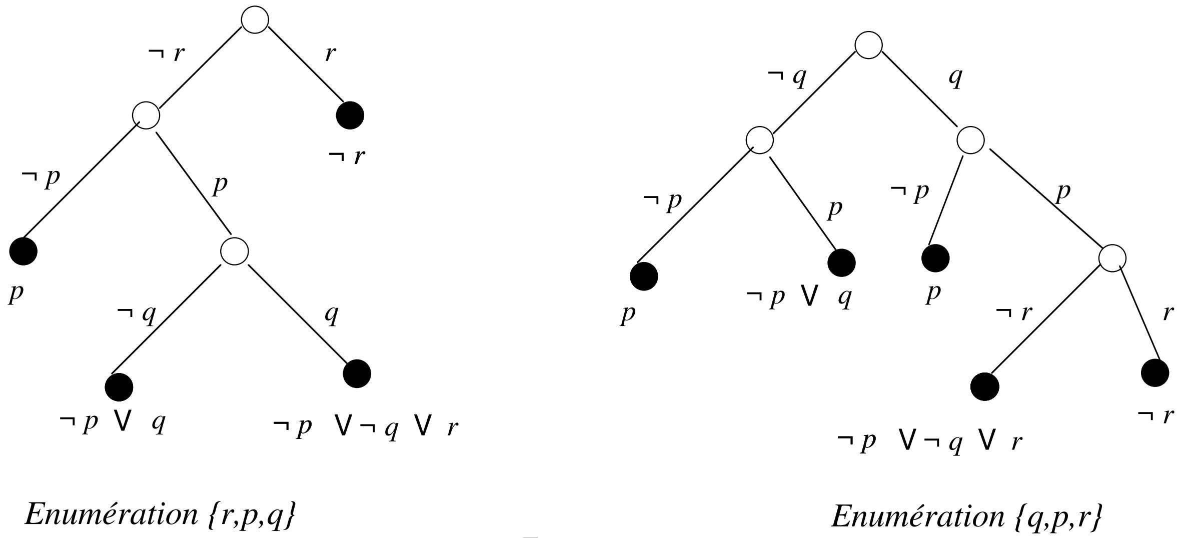

Figure 1.2

Les arbres sémantiques de la figure 1.2 ne sont pas fermés pour l'ensemble de clauses $\mathcal{F}' = \{\neg p \vee q \vee r) \quad , \quad \neg p \vee q \quad , \quad p \quad , \quad \neg r\}$ qui est satisfaisable.

Remarque 1.22 Pour améliorer l'efficacité de la représentation, on peut aussi utiliser un ordre d'énumération différent pour les différents sous-arbres d'un arbre sémantique ; on peut ainsi obtenir des arbres plus petits.

1.2.9 Résolution pour le calcul propositionnel : méthode des coupures

L'idée de la méthode de résolution est la suivante : rappelons que $P \models A$ si et seulement si $P \cup \{\neg A\}$ est insatisfaisable. On montre le

Théorème 1.23 *$P \cup \{\neg A\}$ est insatisfaisable si et seulement si on peut déduire par résolution la clause vide à partir de $P \cup \{\neg A\}$.*

La résolution se réduit ici à la méthode des coupures, que nous illustrons sur deux exemples avant de la définir formellement.

Z Dans cette section, nous emploierons une terminologie proche de PROLOG. Une clause $p_1 \vee \ldots \vee p_n \vee \neg p'_1 \vee \ldots \vee \neg p'_m$ sera notée $p_1, \ldots, p_n \longleftarrow \neg p'_1, \ldots, \neg p'_m$. Un ensemble de clauses positives (ayant un seul littéral positif) sera appelé un programme. Une interprétation I des variables propositionnelles sera aussi appelée *valuation* et notée v.

Exemple 1.24 1) Soit le programme P_0

1 : $arc(a, aa) \longleftarrow$

2 : $arc(aa, aab) \longleftarrow$

3 : $chem(a, aa) \longleftarrow arc(a, aa)$

4 : $chem(aa, aab) \longleftarrow arc(aa, aab)$

5 : $chem(a, aab) \longleftarrow chem(a, aa), chem(aa, aab)$

Pour montrer que $P_0 \models chem(a, aab)$, on montre que l'on peut déduire la clause vide $\Box$ de $P_0 \cup \{\neg chem(a, aab)\}$. On remarque que $\neg chem(a, aab)$ peut s'écrire

6 $\longleftarrow chem(a, aab)$

et on vérifie que

7 : $1 + 3 \Longrightarrow \quad chem(a, aa) \longleftarrow$

8 : $2 + 4 \Longrightarrow \quad chem(aa, aab) \longleftarrow$

9 : $5 + 6 \Longrightarrow \quad \longleftarrow chem(a, aa), chem(aa, aab)$

10 : $7 + 9 \Longrightarrow \quad \longleftarrow chem(aa, aab)$

10 : $8 + 10 \Longrightarrow \quad \longleftarrow \Box$

2) Soit le programme $P_1 = \{A \longleftarrow \quad , \quad C \longleftarrow A, B \quad , \quad B \longleftarrow A\}$.

Pour montrer que $P_1 \models C$, on montre que l'on peut déduire la clause vide $\Box$ de $P_1 \cup \{\neg C\}$, ce qui se fait comme suit.

De $\longleftarrow C$ et $C \longleftarrow A, B$ on tire $\longleftarrow A, B$;

de $\longleftarrow A, B$ et $B \longleftarrow A$ on tire $\longleftarrow A$;

de $\longleftarrow A$ et $A \longleftarrow$ on tire la clause vide $\Box$;

Enonçons maintenant formellement la règle de coupure.

Définition 1.25 *Soient* C : $A, A_1, \ldots, A_n \longleftarrow B_1, \ldots, B_p$ *et* C' : $A'_1, \ldots, A'_{n'} \longleftarrow A, B'_1, \ldots, B'_{p'}$ *deux clauses où le même atome* A *se trouve une fois à gauche de la flèche et une fois à droite, alors on peut en déduire la clause* R : $A_1, \ldots, A_n, A'_1, \ldots, A'_{n'} \longleftarrow B_1, \ldots, B_p, B'_1, \ldots, B'_{p'}$ *appelée résolvent de* C *et* C'*, en mettant ensemble d'une part les hypothèses de* C *et* C'*, d'autre part les conclusions de* C *et* C'*, et en supprimant l'atome commun* A *des deux côtés, i.e.*

$$\frac{A, A_1, \ldots, A_n \longleftarrow B_1, \ldots, B_p \qquad A'_1, \ldots, A'_{n'} \longleftarrow A, B'_1, \ldots, B'_{p'}}{A_1, \ldots, A_n, A'_1, \ldots, A'_{n'} \longleftarrow B_1, \ldots, B_p, B'_1, \ldots, B'_{p'}}$$

Notation : $R = res(C, C')$.

On peut démontrer que la règle de coupure est valide (ou consistante, i.e. on ne peut faire que des déductions correctes à l'aide de cette règle) et complète (i.e.toutes les déductions correctes peuvent être faites à l'aide de cette règle). Pour la validité, c'est une conséquence du lemme suivant.

Lemme 1.26 *Soient* C : $A, A_1, \ldots, A_n \longleftarrow B_1, \ldots, B_p$ *et* C' : $A'_1, \ldots, A'_{n'} \longleftarrow A, B'_1, \ldots, B'_{p'}$ *deux clauses,* R : $A_1, \ldots, A_n, A'_1, \ldots, A'_{n'} \longleftarrow B_1, \ldots, B_p, B'_1, \ldots, B'_{p'}$ *leur résolvent, et* I *une valuation.*

1) si $I(C) = I(C') = 1$ *alors* $I(R) = 1$,

2) si $I(R) = 0$ *alors* $I(C) = 0$ *ou* $I(C') = 0$.

Démonstration. Montrons le 2) et le 1) s'en déduit aussitôt. Supposons $I(R) = 0$, alors

– $\forall i \in \{1, \ldots, n\} \quad I(A_i) = 0$, et $\forall i \in \{1, \ldots, n'\} \quad I(A'_i) = 0$,

– $\forall j \in \{1, \ldots, p\} \quad I(B_j) = 1$, et $\forall j \in \{1, \ldots, p'\} \quad I(B'_j) = 1$.

On distingue les 2 cas possibles pour $I(A)$

– soit $I(A) = 1$, et alors $I(C') = 0$,

– soit $I(A) = 0$, et alors $I(C) = 0$.

D'où le 2) ; le 1) est la contraposée, donc en découle aussitôt. □

Remarque 1.27 Les réciproques sont fausses, par exemple soit :

$$\frac{A \longleftarrow B \quad B \longleftarrow A}{A \longleftarrow A}$$

P' : $A \longleftarrow A$ est le résolvant de $P = \{A \longleftarrow B \quad , \quad B \longleftarrow A\}$, P' est valide mais pas P, c'est-à-dire que, pour toute valuation $I(A \longleftarrow A) = 1$, alors qu'il existe des valuations qui ne satisfont pas $I(A \longleftarrow B) = I(B \longleftarrow A) = 1$.

Z On ne peut pas couper un atome à droite et à gauche dans la même règle, ou deux fois. Par exemple, la coupure

$$\frac{A \longleftarrow A, B \quad D \longleftarrow A}{D \longleftarrow A, B}$$

est correcte, alors que la coupure

$$\frac{A \longleftarrow A, B \quad D \longleftarrow A}{D \longleftarrow B}$$

n'est pas correcte : il suffit de prendre $I(A) = I(D) = 0$ et $I(B) = 1$ pour s'en convaincre.

Corollaire 1.28 *Soit $P = \{C_1, C_2, C_3, \ldots, C_n\}$ un ensemble de clauses, $R = Res(C_1, C_2)$ le résolvent de C_1 et C_2 et soit $P' = \{R, C_1, C_2, C_3, \ldots, C_n\}$ l'ensemble de clauses déduit de P par une coupure portant sur C_1 et C_2. Alors*

1) si P est valide alors P' est aussi valide,

2) si P' est insatisfaisable alors P est aussi insatisfaisable.

Démonstration. Montrons le 2). Si P' est insatisfaisable, alors pour toute valuation I,

(i) soit $\exists i \in \{3, \ldots, n\}$ tel que $I(C_i) = 0$,

(ii) soit $I(R) = 0$.

Dans le cas (i), I ne satisfait pas P trivialement, et dans le cas (ii) par le lemme précédent, soit $I(C_1) = 0$ soit $I(C_2) = 0$ et I ne satisfait pas P non plus.

Montrons le 1). Si toute valuation I satisfait P, alors $\forall i = 1, \ldots, n$, $I(C_i) = 1$, de plus par le lemme précédent, $I(R) = 1$: donc $I(R) = I(C_3) = I(C_4) = \cdots = I(C_n) = 1$, et donc toute valuation I satisfait P' qui est aussi valide. □

On peut déduire du corollaire 1.28 la validité de la méthode de coupure pour le calcul propositionnel.

Définition 1.29 *Soit $P = \{C_1, C_2, C_3, \ldots, C_p\}$ un ensemble de clauses et soit C une clause, on dit que P se dérive en C par résolution si et seulement si il existe une suite de clauses $C'_0, C'_1, \ldots, C'_n = C$ telle que $\forall i = 0, \ldots, n$,*

- *soit $C'_i \in P$*
- *soit il existe j et k tels que $j < i$, $k < i$, et $C'_i = Res(C'_j, C'_k)$ se déduit de C'_j et C'_k par une seule coupure (on dira une étape de résolution).*

Notation : $P \vdash^* C$ ou bien si on veut préciser le nombre d'étapes de résolution, et si k est ce nombre –k est donc le nombre de C'_i qui sont des résolvants obtenus par coupure dans la suite $C'_0, C'_1, \ldots, C'_n$ – on écrit $P \vdash^k C$.

Remarquons qu'une clause intermédiaire (et en particulier une clause de P peut être utilisée plusieurs fois).

Une dérivation de la clause vide à partir de P est appelée une réfutation (par coupure ou résolution) de P.

Exemple 1.30 Construisons une dérivation par résolution de r à partir de $P = \{(\neg p \vee \neg q \vee r) \quad , \quad \neg p \vee q \quad , \quad p\}$. P se récrit en $P = \{r \longleftarrow p, q \quad , \quad q \longleftarrow p \quad , \quad p \longleftarrow\}$. On a les coupures

$$\dfrac{\dfrac{r \longleftarrow p, q \qquad q \longleftarrow p}{r \longleftarrow p} \qquad p \longleftarrow}{r}$$

On peut donc réfuter $P \cup \{\neg r\}$: il suffit d'ajouter une dernière coupure ($\{r \longleftarrow \quad , \quad \longleftarrow r\}$ qui donne la clause vide) à la dérivation précédente.

Exemple 1.31 L'ensemble de clauses $P = \{(p \vee q \vee r) \quad , \quad \neg q \vee r \quad , \quad \neg p \vee r \quad , \quad \neg r \vee q \quad , \quad \neg r \vee \neg q\}$ est insatisfaisable. On peut en effet déduire par réfutation la clause vide avec la suite de clauses : $C'_0 = \neg q \vee r \quad , \quad C'_1 = \neg r \vee \neg q \quad , \quad C'_2 = \neg q \quad , \quad C'_3 = \neg r$ (réutiliser C'_0) $C'_4 = \neg p \vee r \quad , \quad C'_5 = \neg p \quad , \quad C'_6 = p \vee q \vee r \quad , \quad C'_7 = q \vee r \quad , \quad C'_8 = q$ (réutiliser C'_6

avec C'_3), $C'_9 = \square$ (réutiliser C'_6 avec C'_2). On remarquera l'utilisation répétée de certaines clauses.

Théorème 1.32 *(Validité) Si on peut déduire la clause vide de P par résolution, i.e. si $P \vdash^* \square$, alors P est insatisfaisable.*

Démonstration. Par induction sur le nombre d'étapes de coupures, i.e. par induction sur k tel que $P \vdash^k \square$.

- Si $k = 1$, alors $P = \{A \longleftarrow \quad , \quad \longleftarrow A\}$ et P est clairement insatisfaisable ;
- Si l'hypothèse d'induction est vraie pour $k-1$ étapes de coupures, et si $P \vdash^k \square$, alors $P \vdash P_1 \vdash^{k-1} \square$: par l'hypothèse d'induction, P_1 est insatisfaisable, mais comme P_1 s'obtient à partir de P par une seule coupure, par le corollaire 1.28 2) , P est aussi insatisfaisable. $\square$

Le théorème de complétude est un peu plus complexe : nous en donnons ici une preuve simple mais non constructive.

Théorème 1.33 *(Complétude) Si P est insatisfaisable, alors on peut déduire la clause vide de P par résolution, i.e. $P \vdash^* \square$.*

C'est une conséquence immédiate du

Lemme 1.34 *Soit P insatisfaisable et $C : L_1 \vee \cdots \vee L_n$ une clause $C \in P$. Alors il existe $D : L_{i_1} \vee \cdots \vee L_{i_p} \subsetneq C$, avec $\{i_1, \ldots, i_p\} \subsetneq \{1, \ldots, n\}$, $p < n$, (i.e. D est une clause strictement incluse dans C) telle que l'on puisse dériver D par résolution à partir de P.*

Démonstration. Par induction sur le nombre n d'atomes dans P.

- Si $n = 1$, alors $P = \{A \longleftarrow \quad , \quad \longleftarrow A\}$ et clairement $P \vdash \square \subsetneq C$ pour C l'une des 2 clauses de P.
- Supposons le résultat vrai si P' comporte au plus $n-1$ atomes et soit P comportant n atomes. $C : L \vee L_2 \vee \cdots \vee L_k$ une clause $C \in P$ où l'on distingue un littéral L. On construit P' ayant un atome de moins que P (l'atome correspondant au littéral L) comme suit :
  - on supprime toutes les clauses de P contenant $\neg L$,
  - on enlève L dans toutes les clauses de P contenant L,
  - toutes les clauses de P ne contenant ni $\neg L$, ni L sont inchangées.

Alors

- la clause C' correspondant à $C : L \vee L_2 \vee \cdots \vee L_k$ dans P' est $C' : L_2 \vee \cdots \vee L_k$
- P' a au plus $n-1$ atomes,
- P' est insatisfaisable (soit il existe dans P une clause qui ne contient que L et alors P' contient la clause vide, soit sinon, si une valuation I' satisfaisait P', alors la valuation I égale à I' sur les atomes autres que L et telle que $I(L) = 0$ satisfairait P, une contradiction).

Par l'hypothèse d'induction, il existe donc une dérivation δ' par résolution $\delta'\colon P' \vdash^q D' \subsetneq C'$; à chaque étape de δ' on remet L dans chacune des clauses de la dérivation δ' provenant d'une clause de P où L figurait, on obtient ainsi une dérivation par résolution δ de P, $\delta\colon P \vdash^q D \subsetneq C$ avec soit $D = D'$ soit $D = D' \vee L$, i.e. dans les 2 cas $D \subsetneq C$. $\square$

Le théorème 1.33 découle immédiatement du lemme 1.34 par induction sur la longueur de la clause C ; on choisit une clause C dans P, on en déduit par résolution une clause $D \subsetneq C$; on remarque que $P \cup \{D\}$ est aussi insatisfaisable, et on itère le processus avec $P \cup \{D\}$ et la clause D qui est strictement plus petite que C ; on finira donc par obtenir la clause vide.

On peut déduire des théorèmes 1.32 et 1.33 la méthode de réfutation par coupure pour vérifier qu'une formule F est conséquence d'un ensemble fini de formules P.

1) Mettre P sous la forme d'un ensemble de clauses S, obtenu en mettant chacune des formules de P sous FNC.

2) Mettre $\neg F$ sous la forme d'un ensemble de clauses S',

3) Essayer de déduire la clause vide par coupure à partir de $S \cup S'$. Si c'est possible, $S \cup S'$ n'est pas satisfaisable et donc $P \cup \{\neg F\}$ non plus, et F est conséquence de P. Sinon, si l'ensemble des clauses obtenues par coupure à partir de $S \cup S'$ ne contient pas la clause vide, $S \cup S'$ est satisfaisable et F n'est pas conséquence de P.

Remarque 1.35 La méthode de résolution est complète pour réfuter, pas pour prouver : soit par exemple

$$P : \begin{cases} p \longleftarrow \\ q \longleftarrow \end{cases}$$

et soit le but $C = \{p, q \longleftarrow\}$. On ne peut pas déduire C de P par résolution, par contre on peut déduire la clause vide de $P \cup \{\neg C\}$ par résolution.

1.3 Calcul des prédicats du premier ordre

Un “prédicat” est une affirmation qui porte sur des objets et qui peut être vraie ou fausse selon les objets auxquels elle s'applique. Par exemple, “être un nombre pair” est vraie quand elle s'applique à “2” et fausse quand elle s'applique à “3”. Un prédicat peut aussi s'appliquer à plusieurs objets, comme par exemple “être plus petit que”. Cette affirmation est vraie pour le couple (2,3) et fausse pour le couple (3,2).

Le calcul des prédicats permet de construire des énoncés complexes à partir de prédicats, comme par exemple “ tout nombre premier strictement supérieur à 2 est impair”, qui s'écrira formellement

$$\forall x \quad \big((\text{Premier}(x) \wedge x > 2) \Longrightarrow \text{Impair}(x) \big)$$

Ces énoncés complexes sont aussi susceptibles d'être vrais ou faux.

1.3.1 La syntaxe : Formules du premier ordre

Soit $\mathcal{G}$ un ensemble de *symboles de fonctions*. A chaque symbole f de $\mathcal{G}$ est associée une arité (ou rang) $\rho(f) \in \mathbb{N}$. Si $\rho(f) = 0$, alors f est aussi appelé symbole de constante. Soit $C = \{a, b, \ldots, a', b', \ldots, a_1, b_1, \ldots\}$ l'ensemble des symboles de constantes.

Soit $\mathcal{R}$ un ensemble de *symboles de relations*. A chaque symbole R de $\mathcal{R}$ est associée une arité (ou rang) $\rho(R) \in \mathbb{N}$. Si $\rho(R) = 0$, alors R est aussi appelé symbole propositionnel.

Soit $X = \{x, y, \ldots, x', y', \ldots, x_0, y_0, x_1, y_1, \ldots\}$ un ensemble de symboles de variables.

Soit le langage $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. Nous considérons aussi les symboles $\supset, \neg, \wedge, \vee$ de la logique propositionnelle, et deux symboles $\forall$ et $\exists$, appelés quantificateurs universel et existentiel, ainsi que les deux parenthèses et la virgule.

Rappelons que l'ensemble T des *termes sur* $\mathcal{G} \cup X$ est défini inductivement par :

(B) $C \cup X \subseteq T$,

(I) pour tout symbole de fonction f d'arité n dans $\mathcal{G}$, et pour tous $t_1, \ldots, t_n$ dans T, $f\big(t_1, \ldots, t_n\big) \in T$.

Un *terme clos* est un terme sans variable.

Les *formules du premier ordre* pour $\mathcal{L}$ sont définies inductivement par :

- Si R est un symbole de relation d'arité n, et si $t_1, \ldots, t_n \in T$, alors $R(t_1, \ldots, t_n)$ est une formule, appelée *formule atomique*.
- Si F et F' sont des formules, alors $\neg F$, $(F \supset F')$, $(F \wedge F')$, et $(F \vee F')$ sont des formules.
- Si F est une formule et x est une variable, alors $\forall x F$ et $\exists x F$ sont des formules.

Exemple 1.36 1) $F = (\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$.

2) Comme $\mathcal{R}$ peut contenir des symboles relationnels d'arité 0, le calcul propositionnel est un "sous-calcul" du calcul des prédicats. Toute formule propositionnelle est donc une formule du premier ordre, puisque d'une part les symboles propositionnels sont des symboles relationnels d'arité 0, et d'autre part tous les autres symboles du calcul propositionnel sont aussi des symboles du calcul des prédicats.

Définition 1.37 *Une occurrence d'une variable x dans une formule F est un couple (x, n) tel que le n-ième symbole de F est x et que le $(n-1)$-ième n'est ni $\forall$ ni $\exists$.*

Exemple 1.38 $(x, 8)$ et $(x, 17)$ sont les deux occurrences de x dans la formule F ci-dessus, $(x, 7)$ et $(x, 14)$ n'en sont pas : $(x, 7)$ parce que le 7-ième symbole de F n'est pas un x, et $(x, 14)$ parce que le 14-ième symbole de F, qui est bien un x, est quantifié par $\exists$.

Définition 1.39 *Soit F une formule. L'ensemble $SF(F)$ des sous-formules de F est l'ensemble des couples (n, F') avec $n \in \mathbb{N}$ et où*

- *F' est une suite de symboles consécutifs de F qui est elle-même une formule,*
- *n est l'occurrence du premier symbole de F' dans F.*

Exemple 1.40 Les sous-formules de $(\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$ sont $(1, F)$, $(2, \forall x \exists y R(x, y))$, $(4, \exists y R(x, y))$, $(6, R(x, y))$, $(13, \exists x R'(x, y, a))$, et $(15, R'(x, y, a))$.

Les formules peuvent être représentées par des arbres ; par exemple la formule

$$\Big((\forall x \exists y R(x,y)) \supset \exists x R'(x,y,a)\Big)$$

est représentée par l'arbre t de la figure 1.3.

A chaque noeud de t étiqueté par un symbole de relation, un quantificateur, ou l'un des symboles $\supset, \neg, \wedge, \vee$, est associé un sous-arbre t' de t ; chaque sous-arbre t' représente une sous-formule de F. Les sous-formules de $(\forall x \exists y R(x,y) \supset \exists x R'(x,y,a))$ sont représentées dans la figure 1.4.

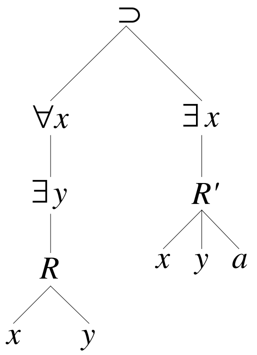

FIGURE 1.3

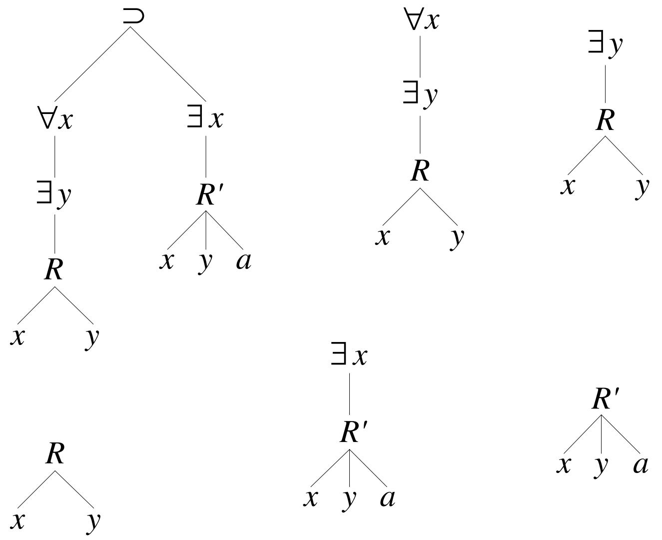

FIGURE 1.4

Une occurrence (x,n) de x dans F apparaît dans une sous-formule (p,F') de F si et seulement si $p \leq n \leq p + |F'|$, où $|F'|$ dénote le nombre de symboles de F'.

EXEMPLE 1.41 $(x,8)$ apparaît dans $(1,F)$, dans $(2, \forall x \exists y R(x,y))$, dans $(4, \exists y R(x,y))$, et dans $(6, R(x,y))$.

$(y,19)$ apparaît dans les sous-formules $(1,F)$, $(13, \exists x R'(x,y,a))$, et $(15, R'(x,y,a))$.

Définition 1.42 *Une occurrence (x,n) de la variable x dans la formule F est une occurrence liée si elle apparaît dans une sous-formule (p, QxF'), avec $Q \in \{\forall, \exists\}$. Sinon c'est une occurrence libre.*

Une variable x est dite libre dans une formule F si elle a au moins une occurrence libre ; x est dite liée dans F si toutes ses occurrences sont liées.

EXEMPLE 1.43 Dans l'exemple 1.36, les occurrences $(x, 8), (x, 17)$, et $(y, 10)$ sont liées ; l'occurrence $(y, 19)$ est libre.

EXERCICE 1.8 Trouver les variables libres et les occurrences libres des variables dans les formules suivantes :

- $\exists x \big(\text{logicien}(x) \wedge \text{malin}(x)\big)$
- $\big(\exists x\, \text{logicien}(x)\big) \wedge \text{malin}(x)$ ♢

Soit $l(F)$ l'ensemble des variables libres de F.

Proposition 1.44

- $l(R(t_1, \ldots, t_n)) = \{u_i \,/\, u_i \in X$ *et* u_i *apparaît dans* $R(t_1, \ldots, t_n)\}$,
- $l(\neg F) = l(F)$,
- $l(F \supset F') = l(F \wedge F') = l(F \vee F') = l(F) \cup l(F')$, *et*
- $l(\forall x F) = l(\exists x F) = l(F) \setminus \{x\}$.

Démonstration. Simple : par induction structurelle sur F. □

1.3.2 Sémantique : interprétation des formules

Soit $\mathcal{R}$ un ensemble de symboles de relations et $\mathcal{G}$ un ensemble symboles de fonctions. Soit $\mathcal{L}$ le langage $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. Le langage $\mathcal{L}$ a beaucoup d'interprétations possibles, chaque interprétation étant adaptée à un domaine de discours. Pour interpréter le langage $\mathcal{L}$ nous devons spécifier le domaine de discours, ainsi que l'interprétation des symboles de prédicats et de fonctions ; pour ce faire nous définissons une $\mathcal{L}$-structure.

Définition 1.45 *Une $\mathcal{L}$-structure est un triplet $S = \langle E, \gamma, h\rangle$ où*

- *E est un ensemble non vide,*
- *γ est une application associant à chaque $R \in \mathcal{R}$ un sous-ensemble $\gamma(R)$, aussi noté R_S, de $E^{\rho(R)}$, et*
- *h est une application associant à chaque $f \in \mathcal{G}$ une fonction $h(f) = f_S$ de $E^{\rho(f)}$ dans E. A chaque constante $a \in C$, h associe un élément $h(a) = a_S$ de E.*

Remarquons ici que E^0 n'a, par définition qu'un seul élément (pour la même raison que $n^0 = 1!$). Donc $\mathcal{P}(E^0)$ n'a que deux parties $\emptyset$ et E^0 et peut être identifié à l'algèbre de Boole à deux éléments.

Une valuation v est une application de l'ensemble des variables dans E. Deux valuations v et v' sont congruentes sur une partie Y de X, ce qui est noté $v \underset{Y}{=} v'$, si pour tout $x \in Y$, $v(x) = v'(x)$.

Définition 1.46 *(i) Si t est un terme et v une valuation, nous définissons $v^*(t) \in E$ par :*

- *Si $t = a \in C$, alors $v^*(t) = a_S$.*
- *Si $t = u \in X$, alors $v^*(t) = v(u)$.*
- *Si $t = f(t_1, \ldots, t_n)$, alors $v^*(t) = f_S(v^*(t_1), \ldots, v^*(t_n))$.*

(ii) Si F est une formule et v une valuation, nous pouvons définir de manière unique la valeur de vérité $\bar{v}(F)$ de F, $\bar{v}(F) \in \mathbb{B}$, par :

– *Si* $F = R(t_1, \dots, t_n)$, *alors*

$$\bar{v}(F) = 1 \text{ si et seulement si } \big(v^*(t_1), \dots, v^*(t_n)\big) \in R_S \ .$$

Remarquons que si R *est d'arité* 0, *alors*

$$\bar{v}(F) = \begin{cases} 1 & \text{si } R_S \neq \emptyset, \\ 0 & \text{si } R_S = \emptyset. \end{cases}$$

– $\bar{v}(\neg F) = \overline{\bar{v}(F)}$.
– $\bar{v}(F \supset F') = 1$ *si et seulement si* $\bar{v}(F) \leq \bar{v}(F')$.
– $\bar{v}(F \wedge F') = 1$ *si et seulement si* $\bar{v}(F) = 1$ *et* $\bar{v}(F') = 1$.
– $\bar{v}(F \vee F') = 1$ *si et seulement si* $\bar{v}(F) = 1$ *ou* $\bar{v}(F') = 1$.
– $\bar{v}(\forall x F) = 1$ *si et seulement si pour toute* v' *telle que* $v' \underset{X-\{x\}}{=} v$, *nous avons* $\bar{v}'(F) = 1$.
– $\bar{v}(\exists x F) = 1$ *si et seulement s'il existe* v' *telle que* $v' \underset{X-\{x\}}{=} v$ *et* $\bar{v}'(F) = 1$.

(iii) Deux formules F *et* F' *sont équivalentes si, pour toute* $\mathcal{L}$*-structure* S *et pour toute valuation* v, $\bar{v}(F) = \bar{v}(F')$. *Nous notons* $F \approx F'$.

Cette sémantique est bien consistante avec la sémantique déjà donnée pour le calcul propositionnel. Nous avons vu dans l'exemple 1.36 que toute formule F du calcul propositionnel est une formule du calcul des prédicats ; si S est une $\mathcal{L}$-structure, la restriction I de S aux symboles propositionnels est une application de ces symboles propositionnels dans l'algèbre de Boole, et donc une interprétation au sens du calcul propositionnel ; et nous avons bien, pour toute formule propositionnelle F, $I(F) = v(F)$ pour toute valuation v à valeurs dans S. En résumé, les interprétations I que nous avons considérées pour la logique propositionnelle peuvent s'obtenir comme restrictions des $\mathcal{L}$-structures.

Proposition 1.47 $\forall x F \approx \neg \exists x \neg F$.

Démonstration.

$$\begin{aligned} \bar{v}(\forall x F) = 0 &\iff \text{il y a une } v' \text{ telle que } v' \underset{X-\{x\}}{=} v \text{ et } \bar{v}'(F) = 0 \\ &\iff \text{il y a une } v' \text{ telle que } v' \underset{X-\{x\}}{=} v \text{ et } \bar{v}'(\neg F) = 1 \\ &\iff \bar{v}(\exists x \neg F) = 1 \\ &\iff \bar{v}(\neg \exists x \neg F) = 0 \end{aligned}$$

□

Proposition 1.48 *Soit* Y *l'ensemble des variables qui ont une occurrence libre dans* F. *Si* $v \underset{Y}{=} v'$, *alors* $\bar{v}(F) = \bar{v}'(F)$.

Remarque 1.49 Si une formule F ne contient aucune occurrence libre (on dit alors que F est une formule *close* ou une *formule de base* ou une *sentence*), alors sa valeur de vérité dans S ne dépend pas de la valuation. En effet pour toutes v, v', nous avons $v \underset{\emptyset}{=} v'$, et donc pour toutes v, v', $\bar{v}(F) = \bar{v}'(F)$. C'est en particulier le cas si F est une formule de la logique propositionnelle.

Exercice 1.9 Dans les syllogismes aristotéliciens, interviennent souvent des propriétés P, Q des individus et des assertions ayant les formes suivantes *(i) Tous les P sont des Q,*

(ii) Certains P sont des Q,

(iii) Aucun P n'est un Q,

(iv) Certains P ne sont pas des Q.

Traduire ces assertions par des formules du calcul des prédicats, en introduisant les prédicats $P(x)$ et $Q(x)$. $\diamondsuit$

EXERCICE 1.10 Montrer que si x n'est pas libre dans F, alors

$$\bar{v}(\forall x F) = \bar{v}(\exists x F) = \bar{v}(F). \qquad \diamondsuit$$

Définition 1.50 *Une formule F est dite*

- *satisfaisable dans S s'il existe une valuation v telle que $\bar{v}(F) = 1$,*
- *satisfaisable s'il existe une structure S et une valuation v telles que $\bar{v}(F) = 1$,*
- *valide (ou vraie) dans S si pour toute v, $\bar{v}(F) = 1$, et*
- *universellement valide si elle est valide dans toutes les $\mathcal{L}$-structures.*

EXEMPLE 1.51 1) $\big((\neg\exists x P(x)) \Longleftrightarrow \forall x(\neg P(x)) \big)$ est universellement valide.

Si x et y sont des nombres réels, et si S est la structure associée à $\mathbb{R}$, alors $x \leq x + y$ est satisfaisable dans S mais elle n'est pas valide dans S.

Soit $F = R(x, z) \wedge Q(x, y, z)$. Considérons la structure $S = (\mathbb{N}, R_{\mathbb{N}}, Q_{\mathbb{N}})$, définie par $R_{\mathbb{N}} = \{(n, m) \,/\, n < m\}$ et $Q_{\mathbb{N}} = \{(n, m, p) \,/\, n + m = p\}$. F est satisfaisable dans S (soit par exemple $v(x) = v(y) = 1$, $v(z) = 2$), mais n'est pas valide dans S (soit par exemple $v(x) = v(z) = 1$, $v(y) = 0$).

2) Soit $\mathcal{R} = \{$mâle, femelle$\}$, constitué de deux prédicats unaires. Alors

$$A = \big(\,(\neg \text{ mâle}(x)) \supset \text{ femelle}(x) \,\big)$$

est satisfaisable mais non valide, alors que

$$C = \Big(\big(\,(\neg\text{mâle}(x)) \supset \text{ femelle}(x) \,\big) \;\vee\; \big(\,\neg\text{mâle}(x) \wedge \neg\text{femelle}(x) \,\big)\Big)$$

est valide.

EXERCICE 1.11 Soit $S = \langle E, \{R, =\}\rangle$ un ensemble muni d'une relation R et d'un prédicat $=$ que nous interprétons par l'égalité. Ecrire une formule qui est valide dans S si et seulement si R est un ordre (resp. un ordre total). $\diamondsuit$

EXERCICE 1.12 1) $\exists y \forall x\, r(x, y) \approx \forall x \exists y\, r(x, y)$ est-elle valide pour un prédicat binaire r ? Même question pour $(\exists y p(y)) \wedge (\exists y q(y)) \approx (\exists y(p(y) \wedge q(y)))$, avec p et q des prédicats unaires. Donner une preuve si la réponse est oui, un contre-exemple si la réponse est non.

2) Montrer que $\exists y \forall x \,\big(p(x) \wedge q(y)\big) \approx \forall x \exists y \,\big(p(x) \wedge q(y)\big)$, pour des prédicats unaires p et q. $\diamondsuit$

Comme pour la logique propositionnelle nous définissons les séquents.

Définition 1.52 *Un séquent $(\mathcal{F}, F)$ est valide dans S, noté $\mathcal{F} \models_S F$, si*

$$\textit{pour toute } v, \quad \Big(\big(\textit{ pour toute } G \textit{ dans } \mathcal{F},\ \bar{v}(G) = 1\big) \Longrightarrow (\bar{v}(F) = 1)\Big).$$

Un séquent est universellement valide, noté $\mathcal{F} \models F$, s'il est valide dans toute structure S.

Proposition 1.53

$$\{F_1, \ldots, F_n\} \underset{S}{\models} F \text{ si et seulement si } \emptyset \underset{S}{\models} (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots)).$$

Proposition 1.54 *Si $\mathcal{F} \underset{S}{\models} F$ et si x n'est libre dans aucune formule de $\mathcal{F}$, alors $\mathcal{F} \underset{S}{\models} \forall x F$.*

Démonstration. Soit v une valuation telle que pour toute $G \in \mathcal{F}$ $\bar{v}(G) = 1$. Soit v' une valuation quelconque telle que $v' \underset{X-\{x\}}{=} v$.
Comme x n'a aucune occurrence libre dans $\mathcal{F}$, pour toute $G \in \mathcal{F}$ nous avons aussi $\bar{v}'(G) = 1$ et donc $\bar{v}'(F) = 1$. Puisque ceci est vrai pour toute $v' \underset{X-\{x\}}{=} v$, nous avons que $\bar{v}(\forall x F) = 1$. □

1.3.3 Substitution et renommage

Soit F une formule et soit x une variable. Soit u un terme. Soit $F[x := u]$ la formule où toutes les occurrences *libres* de x ont été remplacées par u. $F[x := u]$ est appelée une *instance* de F. $F[x := u]$ est appelée une *instance de base* de F s'il n'y a pas de variables dans $F[x := u]$. Si x n'a aucune occurrence libre dans F, alors $F[x := u] = F$.

On dit que u est *substituable à x dans F* si u est un terme clos ou si u est tel que toutes les occurrences de variables dans u sont libres dans $F[x := u]$.

Lorsque u est une variable, on parle de *renommage* au lieu de substitution. Enfin, si x a des occurrences liées dans F, x apparait dans des sous-formules de la forme $F' = \forall x G$ ou $F' = \exists x G$ et renommer x en la variable u consiste à remplacer par exemple la sous-formule $\forall x G$ par $\forall u G[x := u]$, et cela sous-entend que u est substituable à x dans G. Si une même variable x a des occurrences liées dans des sous-formules différentes $\forall x G$ et $\forall x H$ par exemple, il est préférable de renommer x par des variables différentes dans $\forall x G$ et $\forall x H$.

EXEMPLE 1.55 Soit $F = \big(\forall y R(x, y, z)\big) \vee \big(\forall z R'(z)\big)$, où R, R' sont des symboles de relations. u est substituable à x dans F si et seulement si y n'apparaît pas dans u. Par exemple y n'est pas substituable à x dans F car $F[x := y] = \big(\forall y R(y, y, z)\big) \vee \big(\forall z R'(z)\big)$ et l'occurrence $(y, 5)$ devient liée. De même, si f est un symbole de fonction, $f(y, z)$ n'est pas substituable à x dans F, mais $f(x, z)$ est substituable à x dans F.

Désormais lorsque nous écrivons $F[x := u]$, nous sous-entendons que u est substituable à x dans F.

Proposition 1.56 *Soit x une variable et soit u un terme substituable à x dans F. Soit v une valuation. Soit v' définie par :*

$$v'(y) = \begin{cases} v(y) & \text{si } y \neq x, \\ v^*(u) & \text{si } y = x. \end{cases}$$

Alors $\bar{v}'(F) = \bar{v}(F[x := u])$.

Démonstration. Par induction sur la construction de F. □

Proposition 1.57 *Soit x une variable et soit u un terme substituable à x dans F.*

1) $\mathcal{F} \underset{S}{\models} \forall x F \quad \Longrightarrow \quad \mathcal{F} \underset{S}{\models} F[x := u]$;

2) $\mathcal{F} \underset{S}{\models} F[x := u] \quad \Longrightarrow \quad \mathcal{F} \underset{S}{\models} \exists x F$.

1.3.4 Un formulaire

Donnons quelques règles fort utiles de calcul des prédicats : nous abrègerons dans cette section $F \approx G$ en $F \iff G$ pour nous conformer aux notations classiques lorsque la logique est utilisée comme méta-langage et pour augmenter la lisibilité ; nous utiliserons aussi la notation $F \implies G$ pour indiquer que $F \supset G$ est universellement valide, c'est-à-dire que $\emptyset \models F \supset G$.

Proposition 1.58

$$(i) \qquad \forall x\,(p(x) \wedge q(x)) \iff \forall x\, p(x) \wedge \forall x\, q(x)$$

$$(ii) \qquad \exists x\,(p(x) \wedge q(x)) \implies \exists x\, p(x) \wedge \exists x\, q(x)$$

Par dualité entre $\forall$ et $\exists$ nous avons aussi

$$(iii) \qquad \exists x\,(p(x) \vee q(x)) \iff \exists x\, p(x) \vee \exists x\, q(x)$$

$$(iv) \qquad \forall x\, p(x) \vee \forall x\, q(x) \implies \forall x\,(p(x) \vee q(x))$$

Les réciproques des règles (ii) et (iv), à savoir

$$\exists x\, p(x) \wedge \exists x\, q(x) \implies \exists x\,(p(x) \wedge q(x))$$

et

$$\forall x\,(p(x) \vee q(x)) \implies \forall x\, p(x) \vee \forall x\, q(x)\ ,$$

sont fausses (cf. exercice 1.12).

Enfin les règles suivantes, écrites avec les mêmes conventions que ci-dessus, sont utiles pour mettre les formules en forme prénexe, c'est-à-dire avec tous les quantificateurs en tête de formule.

Proposition 1.59 *Soit $* \in \{\vee, \wedge\}$, soit F une formule, soit x une variable, et soit G une formule dans laquelle x n'a aucune occurrence libre. Nous avons :*

(i)	$\neg\forall x\, F \iff \exists x\, \neg F$	$\neg\exists x\, F \iff \forall x\, \neg F$
(ii)	$(\forall x\, F) * G \iff \forall x\,(F * G)$	$(\exists x\, F) * G \iff \exists x\,(F * G)$
(iii)	$G * (\forall x\, F) \iff \forall x\,(G * F)$	$G * (\exists x\, F) \iff \exists x\,(G * F)$
(iv)	$(\forall x\, F) \supset G \iff \exists x\,(F \supset G)$	$(\exists x\, F) \supset G \iff \forall x\,(F \supset G)$
(v)	$G \supset (\forall x\, F) \iff \forall x\,(G \supset F)$	$G \supset (\exists x\, F) \iff \exists x\,(G \supset F)$

Les preuves des deux propositions précédentes sont simples mais fastidieuses.

1.3.5 Formules prénexes

Définition 1.60 *Une formule F est dite prénexe si elle est de la forme $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n F'$ où les Q_i sont des quantificateurs pour $i = 1, 2, \ldots, n$, et où F' est une formule sans quantificateur.*

Théorème 1.61 *Toute formule F est équivalente à une formule prénexe G.*

Exercice 1.13 Trouver une formule prénexe équivalente à

$$\exists x P(x) \wedge \forall x \big(\exists y Q(y) \supset R(x)\big) \qquad \diamondsuit$$

1.4 Théorème de Herbrand et conséquences

1.4.1 Théories et Modèles

Définition 1.62 *Un séquent prouvable est un séquent obtenu par les règles de la logique propositionnelle, auxquelles on ajoute les trois règles suivantes :*

- *Si $\mathcal{F} \vdash \forall x F$ alors : $\mathcal{F} \vdash F[x := t]$ (règle d'instantiation).*
- *Si $\mathcal{F} \vdash F$ et si x n'est pas libre dans $\mathcal{F}$, alors : $\mathcal{F} \vdash \forall x F$ (règle de généralisation universelle).*
- *$\mathcal{F} \vdash \exists x F$ si et seulement si $\mathcal{F} \vdash \neg\forall x \neg F$ (définition de $\exists$).*

La règle de généralisation universelle ne s'applique pas si x est libre dans $\mathcal{F}$. Par exemple, $p(x) \vdash p(x)$ est prouvable, mais $p(x) \vdash \forall x p(x)$ n'est pas prouvable. Cette règle est la formalisation du raisonnement : "si une propriété est vraie pour un objet *arbitraire* x alors elle est vraie pour tout x" ; x est arbitraire signifie qu'aucune hypothèse n'est faite sur x et ce manque de connaissances sur x est formellement exprimé par le fait que x n'est pas libre dans $\mathcal{F}$.

Dans la règle d'instanciation, il ne faut pas oublier que *t doit être substituable à x dans F*. Par exemple, $\forall x \exists y\, p(x,y) \vdash \forall x \exists y\, p(x,y)$ est prouvable, mais $\forall x \exists y\, p(x,y) \vdash \big(\exists y\, p(x,y)\big)[x := y]$ n'est pas prouvable, car y n'est pas substituable à x dans $\exists y\, p(x,y)$, et le séquent $\forall x \exists y\, p(x,y) \vdash \exists y\, p(y,y)$ n'est pas valide.

Théorème 1.63 *(Correction ou Cohérence) Si un séquent est prouvable alors il est universellement valide.*

Démonstration. Comme dans le théorème 1.12 1), il suffit de vérifier par induction sur la longueur des preuves que chaque utilisation d'une des règles de preuve n'engendre que des séquents valides à partir de séquents valides. Nous avons déjà vérifié dans le théorème 1.12 que chacune des règles de construction de preuve de la logique propositionnelle donnée dans la définition 1.10 est valide ; il suffit donc de vérifier que les trois règles données dans la définition 1.62 sont valides.

Par exemple la validité de la règle d'instantiation découle immédiatement de la proposition 1.57 1), la validité de la règle de généralisation découle immédiatement de la proposition 1.54, et la validité de la règle de définition de $\exists$ découle de l'exemple 1.51 1). $\square$

Théorème 1.64 *(Complétude) Si un séquent est universellement valide alors il est prouvable.*

Nous n'allons pas démontrer ce théorème mais seulement donner quelques indications sur la preuve.

Définition 1.65 *Une théorie est un ensemble T de formules tel que pour tout sous-ensemble fini $\mathcal{F}$ de T, si $\mathcal{F} \vdash F$ alors $F \in T$.*

Une théorie T est contradictoire s'il existe une formule F telle que $F \in T$ et $\neg F \in T$.

Exemple 1.66 Soit $\mathcal{F}$ un ensemble fini de formules. L'ensemble

$$Th(\mathcal{F}) = \{F \;/\; \mathcal{F} \vdash F\}$$

est une théorie.

Proposition 1.67 *Une théorie T est contradictoire si et seulement si elle contient toutes les formules.*

Démonstration. Soit G une formule quelconque.

$$\left.\begin{array}{l} F, \neg F, \neg G \vdash F \\ F, \neg F, \neg G \vdash \neg F \end{array}\right\} \quad \text{d'où} \quad F, \neg F \vdash G$$

et donc $G \in T$. □

Proposition 1.68 *$\mathcal{F} \vdash F$ si et seulement si $Th(\mathcal{F} \cup \{\neg F\})$ est contradictoire.*

Démonstration. Si $\mathcal{F} \vdash F$, alors :

$$\left.\begin{array}{rl} & \mathcal{F}, \neg F \vdash F \\ \text{et} & \mathcal{F}, \neg F \vdash \neg F \end{array}\right\} \Longrightarrow \text{la théorie } \mathcal{F} \cup \{\neg F\} \text{ est contradictoire}$$

Si $Th(\mathcal{F} \cup \{\neg F\})$ est contradictoire, $\mathcal{F}, \neg F \vdash F$ et $\mathcal{F}, \neg F \vdash \neg F$, et donc $\mathcal{F} \vdash F$. □

Une $\mathcal{L}$-structure S est un *modèle* d'un ensemble (fini ou infini) $\mathcal{G}$ de formules si pour toute v et pour toute F dans $\mathcal{G}$, nous avons $\bar{v}(F) = 1$.

Nous notons $\emptyset \underset{S}{\models} \mathcal{G}$ le fait que S est un modèle de $\mathcal{G}$.

Un ensemble $\mathcal{G}$ de formules est *satisfaisable* s'*il existe une* $\mathcal{L}$-structure S et s'*il existe une* valuation v telles que pour toute F dans $\mathcal{G}$, nous ayons $\bar{v}(F) = 1$. Un ensemble $\mathcal{G}$ de formules a un *modèle* s'*il existe* une $\mathcal{L}$-structure S telle que *pour toute* valuation v et pour toute F dans $\mathcal{G}$, nous ayons $\bar{v}(F) = 1$. Un ensemble $\mathcal{G}$ de formules qui n'a pas de modèle peut donc être satisfaisable : par exemple, l'ensemble $\mathcal{F}$ de formules défini dans la remarque 1.70 n'a pas de modèle, mais il est satisfaisable.

Si une théorie a un modèle, alors elle n'est pas contradictoire. La réciproque est un des théorèmes fondamentaux de la logique. La preuve de ce théorème est assez longue, nous énoncerons donc le théorème sans le prouver.

Théorème 1.69 *Si $\mathcal{F}$ consiste de formules closes, et si $Th(\mathcal{F})$ n'est pas contradictoire, alors $\mathcal{F}$ a un modèle.*

Remarque 1.70 Le théorème 1.69 est faux si l'on permet des formules non closes. Soit $\mathcal{F} = \{\exists x p(x)\,,\, \neg p(x)\}$; $\mathcal{F}$ est satisfaisable : soit $S = \langle E, \gamma \rangle$ avec $E = \{0, 1\}$, et $\gamma(p) = p_S$ définie par $p_S(0) = 0$, $p_S(1) = 1$, la valuation $v(x) = 0$ est telle que $\bar{v}(F) = 1$ pour toute F dans $\mathcal{F}$. $\mathcal{F}$ n'a pas de modèle : si la structure S' est telle que pour toute valuation v, $\bar{v}\big(\neg p(x)\big) = 1$, alors, pour toute valuation v, $\bar{v}\big(\exists x p(x)\big) = 0$. Néanmoins $Th(\mathcal{F})$ n'est pas contradictoire : sinon, par la proposition 1.68 nous pourrions conclure que $\exists x p(x) \vdash p(x)$, ce qui est faux puisque le séquent $\{\exists x p(x)\,,\, p(x)\}$ n'est pas universellement valide, *e.g.* $\exists x p(x) \underset{S}{\not\models} p(x)$.

Déduisons le théorème de complétude du théorème 1.69. Nous devons montrer que $\mathcal{F} \models F$ implique $\mathcal{F} \vdash F$.

1) Nous considérons d'abord des formules closes : supposons que $\mathcal{F}$ et F consistent de formules closes, que $\mathcal{F} \models F$, et que le séquent $(\mathcal{F}, F)$ n'est pas prouvable. Alors $\mathcal{F} \cup \{\neg F\}$ n'est pas contradictoire et donc a un modèle S par le Théorème 1.69. Toute valuation v vérifie donc pour toute $F_i \in \mathcal{F}$, $\bar{v}(F_i) = 1$ et $\bar{v}(\neg F) = 1$. Mais, comme $\mathcal{F} \models F$, le séquent $(\mathcal{F}, F)$ est valide dans S et donc toute valuation v vérifiant $\bar{v}(F_i) = 1$ pour toute F_i dans $\mathcal{F}$ vérifie aussi $\bar{v}(F) = 1$, en contradiction avec $\bar{v}(\neg F) = 1$.

2) Supposons maintenant que $\mathcal{F} \models F$, et que de plus $\mathcal{F} = \{F_1, \ldots, F_n\}$ et F consistent de formules non nécessairement closes. $\mathcal{F} \models F$ et la proposition 1.53 impliquent que $\emptyset \models (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))$. Soient $x_1, \ldots, x_k$ les variables libres de $(F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))$; par la proposition 1.54, $\emptyset \models \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))$.

$\big(\emptyset\,, \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))\big)$ est donc un séquent universellement valide consistant de formules closes, il est donc prouvable par le cas 1), et $\emptyset \vdash \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))$. Par la règle d'instantiation, $\emptyset \vdash (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots))$. Nous en déduisons que $\{F_1, \ldots, F_n\} \vdash F$, le séquent $(\mathcal{F}, F)$ est donc prouvable. $\square$

Exercice 1.14 Bernard et Christophe sont membres du Club Alpin Français (CAF en abrégé). Tout membre du CAF est skieur ou alpiniste. Les alpinistes n'aiment pas la pluie, et les skieurs aiment la neige. Christophe aime tout ce que Bernard n'aime pas, et n'aime pas tout ce que Bernard aime (c'est-à-dire il y a des choses que Bernard aime et pas Christophe).

1) Exprimer les conditions du CAF par un ensemble $\mathcal{F}$ de formules du calcul des prédicats.

2) Trouver un modèle de $\mathcal{F}$.

3) Peut-on prouver qu'il y a un membre du CAF qui est un alpiniste et pas un skieur (ou vice versa) ? $\diamondsuit$

1.4.2 Problème de la décision et décidabilité

Etant donné une formule F, se pose le problème de savoir si cette formule est universellement valide. Ce problème, appelé *problème de la décision*, est équivalent à celui de savoir si une formule est satisfaisable : en effet, F est universellement valide si et seulement si $\bar{F}$ n'est pas satisfaisable. Il suffit donc de savoir répondre à l'une des deux questions pour avoir la réponse à l'autre. On aimerait disposer d'une méthode effective, par exemple un programme d'ordinateur, auquel on donnerait en entrée une formule arbitraire F, et qui nous donnerait en sortie la réponse "oui, cette formule est une tautologie" ou "non, cette formule n'est pas une tautologie". Par exemple, pour le calcul propositionnel, on a une telle méthode : il suffit de faire la table de vérité de la formule (ou de manière équivalente de la calculer dans toutes les interprétations possibles, ce qui est faisable puisqu'il n'y a qu'un nombre fini d'interprétations à considérer pour chaque formule : s'il y a n variables propositionnelles, on a 2^n interprétations). On dit que le problème de la validité des formules du calcul propositionnel est *décidable*, ou que le problème de la décision pour le calcul propositionnel est *décidable*.

Dès qu'on dépasse le cadre du calcul propositionnel, le problème de savoir si une formule est universellement valide devient très rapidement *indécidable* : la validité d'une formule est en général une notion qui n'est pas effective, *i.e.* il n'y a pas d'algorithme qui réponde par *oui* ou *non* à la question de savoir si une formule F arbitraire est universellement valide. Pour le calcul des prédicats par exemple, on pourrait espérer utiliser le fait que F est universellement valide si et seulement s'il existe une preuve de F à partir des axiomes et des règles (du calcul des séquents, ou du système de Hilbert-Ackermann) : malheureusement, il n'existe pas de méthode générale pour trouver la preuve d'une formule arbitraire, il faut faire preuve d'astuce et de créativité. Nous ne démontrerons pas ici l'indécidabilité de la validité des formules du calcul des prédicats ; nous donnons un exemple qui peut donner l'intuition de la raison pour laquelle les méthodes employées pour le calcul propositionnel ne s'appliqueront pas ici : il existe des formules qui ont des modèles infinis, mais aucun modèle fini, il sera donc difficile de vérifier tous les modèles d'une telle formule.

Exemple 1.71 La formule

$$\forall x \exists y\, R(x,y) \bigwedge \forall x\, \neg R(x,x) \bigwedge \forall x \forall y \forall z \Big(R(x,y) \wedge R(y,z) \supset R(x,z)\Big)$$

admet un modèle infini ($\mathbb{N}$ avec pour $R(x,y)$ la relation $x < y$) mais n'a aucun modèle fini. En effet dans un modèle fini, à cause de $\forall x \exists y\, R(x,y)$, on peut construire un chaîne infinie $a_1, a_2, \ldots, a_n \ldots$ avec pour tout i, $R(a_i, a_{i+1})$, par la finitude du modèle, il existe i, k avec $i < k$ et $a_i = a_k$, et cela implique avec la transitivité de R que $R(a_i, a_i)$ soit vrai, qui contredit $\forall x\, \neg R(x,x)$.

Définition 1.72 *Une théorie T est dite décidable s'il existe un algorithme pour, étant donné une formule arbitraire de T, déterminer si cette formule est prouvable.*

Par exemple, si T_0 (resp. T_1) est l'ensemble des tautologies du calcul propositionnel (resp. du calcul des prédicats), T_0 est décidable et T_1 est indécidable. La plupart des théories sont indécidables.

1.4.3 Modèles de Herbrand

Soit $\mathcal{G}$ un ensemble de fonctions dont l'ensemble de constantes C est non vide. Soit $\mathcal{L}$ le langage $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. L'*univers de Herbrand* de $\mathcal{L}$ est l'ensemble des termes clos (*i.e.* sans variable) construit sur $\mathcal{G}$. L'univers de Herbrand est noté par U_H. U_H est défini inductivement par :

(B) $C \subseteq U_H$,

(I) pour toute f d'arité n dans $\mathcal{G}$, et pour tous $t_1, \ldots, t_n$ dans U_H, $f\big(t_1, \ldots, t_n\big) \in U_H$.

Comme C est non vide, U_H est non vide. (En effet U_H est vide si et seulement si C est vide.)

Une $\mathcal{L}$-structure $S = \langle E, \gamma, h \rangle$ est une *structure de Herbrand* si :

- $E = U_H$, et
- h associe à chaque $f \in \mathcal{G}$ la fonction f_S de $U_H^{\rho(f)}$ dans U_H définie par $f_S(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$. Ceci implique qu'à chaque constante $a \in C$, h associe l'élément a de U_H.

La définition d'une $\mathcal{L}$-structure (définition 1.45) requiert que le domaine de la structure soit non vide. Donc U_H doit être non vide. C'est pourquoi nous supposons C non vide.

La *base de Herbrand* de $\mathcal{L}$ est l'ensemble B_H des formules atomiques closes (ou sans variables), *i.e.* les formules de la forme $R(t_1, \ldots, t_n)$ avec $R \in \mathcal{R}$ et $t_1, \ldots, t_n$ dans U_H. Pour un langage $\mathcal{L}$ donné, il y a un seul univers de Herbrand, mais sur cet univers de Herbrand nous pouvons définir de nombreuses structures de Herbrand ; une structure de Herbrand H est définie par une *interprétation de Herbrand* qui est un sous-ensemble I de la base de

Herbrand B_H ; I spécifie les formules atomiques qui sont vraies dans H. Formellement, I définit $H = \langle U_H, \gamma, h\rangle$ signifie que pour $t_1, \dots, t_n$ dans U_H et $R \in \mathcal{R}$, $(t_1, \dots, t_n)$ est dans R_H si et seulement si $R(t_1, \dots, t_n) \in I$. Une structure de Herbrand sera donc notée par $H = \langle U_H, I, h\rangle$, ou par $H = \langle U_H, I\rangle$, ou simplement par I, puisque U_H et h sont définis de manière unique par le langage.

EXEMPLE 1.73 1) Soit $\mathcal{L}$ le langage $\mathcal{L} = \{a, p, q\}$ où a est un symbole de constante et p, q sont des symboles de relations d'arité 0. Alors l'univers de Herbrand est $\{a\}$, la base de Herbrand est l'ensemble $B_H = \{p, q\}$, et il y a exactement quatre structures de Herbrand définies par $I_0 = \emptyset$, $I_1 = \{p\}$, $I_2 = \{q\}$, $I_3 = \{p, q\}$.

2) Soit $\mathcal{L}$ le langage $\mathcal{L} = \{a, f, p, q\}$ où p, q sont des symboles de relations d'arité 1, a est un symbole de constante, et f est un symbole de fonction unaire. Alors l'univers de Herbrand est

$$U_H = \{a\,,\, f(a)\,,\, f^2(a)\,,\, \dots\,,\, f^n(a)\,, \dots\} = \{f^n(a)\,/\,n \in \mathbb{N}\}\ ,$$

la base de Herbrand est l'ensemble $B_H = \{p(t), q(t)\,/\,t \in U_H\}$, et les structures de Herbrand sont définies par des sous-ensembles I de B_H ; par exemple $I_0 = \emptyset$, $I_1 = B_H$, $I_2 = \{p(a), p(f(a))\}$, $I_3 = \{p(t)\,/\,t \in U_H\}$, etc. définissent des structures de Herbrand H_0, H_1, H_2, H_3, etc.

Proposition 1.74 *Pour toute $\mathcal{L}$-structure $A = \langle E, \gamma, h\rangle$ il y a une unique structure de Herbrand H et une unique application $h^*\colon U_H \longrightarrow E$ telles que*

(i) $h^*\big(f(t_1, \dots, t_n)\big) = f_A(h^*(t_1), \dots, h^*(t_n))$, et

(ii) pour $t_1, \dots, t_n$ dans U_H, $(t_1, \dots, t_n) \in R_H \Longleftrightarrow (h^*(t_1), \dots, h^*(t_n)) \in R_A$.

Définition 1.75 *Soit $\mathcal{F}$ un ensemble de formules du langage $\mathcal{L}$ et soit H une structure de Herbrand pour $\mathcal{L}$. On dit que H est un modèle de Herbrand de $\mathcal{F}$ si et seulement si H est un modèle de $\mathcal{F}$.*

EXEMPLE 1.76 Soit $\mathcal{L} = \{a, f, p, q\}$, soit $\mathcal{F} = \{p(a)\,,\, \forall x\big(p(x) \supset p(f(x))\big)\}$. I_1 et I_3 dans l'exemple 1.73 (2) définissent des modèles de Herbrand de $\mathcal{F}$; I_0 et I_2 dans l'exemple 1.73 (2) ne définissent pas des modèles de Herbrand de $\mathcal{F}$; tout $I = I_3 \cup \{q(f^k(a))\,/\,k \in K \subset \mathbb{N}\}$ définit aussi un modèle de Herbrand de $\mathcal{F}$.

1.4.4 Théorème de Herbrand

Définition 1.77 *Une formule prénexe est universelle si et seulement si elle n'a que des quantificateurs universels.*

Théorème 1.78 *(Théorème de Herbrand) Soit $\mathcal{L}$ un langage avec un ensemble non vide C de constantes, et soit $\mathcal{F}$ un ensemble de formules universelles closes, alors $\mathcal{F}$ a un modèle si et seulement si $\mathcal{F}$ a un modèle de Herbrand.*

Démonstration. La condition "si" est claire. Pour la condition "seulement si", supposons que $\mathcal{F}$ a un modèle $S = \langle E, \gamma, h\rangle$ et construisons un modèle de Herbrand pour $\mathcal{F}$. Soit H la structure de Herbrand définie par l'ensemble I suivant de formules atomiques de la base de Herbrand :

$$I = \{F \in B_H \,/\, \emptyset \underset{S}{\models} F\}$$

Montrons que H est un modèle de Herbrand de $\mathcal{F}$. Comme C est non vide, U_H est non vide, et à chaque valuation $v_H : X \longrightarrow U_H$ nous pouvons associer une unique valuation $v = h^* \circ v_H$,

$$v: X \xrightarrow{v_H} U_H \xrightarrow{h^*} E\,,$$

où h^* est définie dans la proposition 1.74. Par induction structurelle sur les formules, nous pouvons montrer que pour toute formule G sans quantificateur, $\bar{v}_H(G) = \bar{v}(G)$. Le cas de base découle de la proposition 1.74 et l'étape inductive est claire. Nous devons prouver que pour toute $F = \forall x_1 \cdots \forall x_n G$ dans $\mathcal{F}$, où G est une formule sans quantificateur, on a $H \models F$, *i.e.* pour toute valuation $v_H : \{x_1, \ldots, x_n\} \longrightarrow U_H$, $\bar{v}_H(G) = 1$. Si v_H est une valuation, $v = h^* \circ v_H$ est une valuation dans E, et comme $\emptyset \models_S F$, $\bar{v}(G) = 1$. Donc $\bar{v}_H(G) = 1$. $\square$

Exemple 1.79 Le théorème de Herbrand n'est plus vrai si on ne suppose pas que $\mathcal{F}$ est un ensemble de formules universelles. Soit $\mathcal{L} = \{a, R\}$ avec une constante a, un symbole de relation unaire R, et soit $\mathcal{F} = \{R(a)\ ,\ \exists x \neg R(x)\}$. $\mathcal{F}$ a un modèle mais $\mathcal{F}$ n'a pas de modèle de Herbrand. La structure S définie par $E = \{0, 1\}$ avec $a_S = 0$ et $0 \in R_S$, $1 \notin R_S$ est un modèle de $\mathcal{F}$.

$\mathcal{F}$ n'a pas de modèle de Herbrand. Il y a exactement deux structures de Herbrand sur l'univers de Herbrand $U_H = \{a\}$, respectivement définies par $I_0 = \emptyset$ (*i.e.* $R_{I_0} = \emptyset$ est toujours fausse) et $I_1 = \{R(a)\}$ (*i.e.* $R_{I_1} = \{a\}$ est toujours vraie), et aucune n'est un modèle de $\mathcal{F}$.

Remarque 1.80 Le théorème de Herbrand est vrai si $\mathcal{F}$ est un ensemble de formules sans quantificateurs. En effet, si $F(x)$ n'est pas close, S est un modèle de $F(x)$ si et seulement si S est un modèle de $\forall x F(x)$.

En fait, le théorème 1.78 est une forme affaiblie du théorème de Herbrand qui donne le résultat plus général suivant :

Théorème 1.81 *Soit $\mathcal{F}$ un ensemble de formules universelles closes,*

– soit $\mathcal{F}$ a un modèle de Herbrand,

– soit $\mathcal{F}$ n'a pas de modèle, et de plus il y a un ensemble fini d'instances closes de $\mathcal{F}$ dont la conjonction est insatisfaisable.

Nous ne prouverons pas le théorème 1.81 ici.

Le théorème de Herbrand a nombre de conséquences en programmation logique et théorie de la preuve, par exemple :

- Un ensemble satisfaisable $\mathcal{F}$ de formules universelles a un modèle de Herbrand, et donc un modèle fini ou dénombrable.
- Si l'ensemble $\mathcal{F}$ de formules universelles est insatisfaisable, alors le théorème 1.81 permet d'exhiber directement un ensemble fini d'instances closes insatisfaisables. Donc le théorème 1.81 donne une méthode pour construire effectivement soit un modèle de Herbrand pour $\mathcal{F}$ soit un contre-exemple fini montrant qu'il n'existe aucun modèle de $\mathcal{F}$.
- Le théorème de Herbrand implique la complétude de la méthode de résolution ; la méthode de résolution est basée sur l'idée suivante : la formule $F = \exists x G(x)$, où G est

sans quantificateur, est une conséquence de l'ensemble de formules universelles $\mathcal{F}$ si et seulement si $\mathcal{F} \cup \{\neg F\}$ est insatisfaisable. $\mathcal{F} \cup \{\neg F\}$ est un ensemble de formules universelles dont on peut prouver l'insatisfaisabilité en exhibant un ensemble fini d'instances closes insatisfaisables. On peut montrer que le fait d'exhiber les instances closes insatisfaisables donne aussi des valuations$v(x) = t$ telles que $\mathcal{F} \vdash G[x := t]$, et qui sont appelées des substitutions réponses.

– On peut utiliser le théorème de Herbrand pour prouver le Théorème 1.64.

Exercice 1.15 Trouver tous les modèles de Herbrand de

$$\begin{aligned}\mathcal{F} = \{&arc(a, b) \quad , \quad arc(b, c) \quad , \quad \forall x \forall y \big(arc(x, y) \supset chemin(x, y)\big) \quad , \\ &\forall x \forall y \big((arc(x, z) \wedge chemin(z, y)) \supset chemin(x, y)\big)\} \ ,\end{aligned}$$

où le langage $\mathcal{L}$ consiste des constantes a, b, c, et des symboles de relations binaires arc et $chemin$. Avec les notations de PROLOG, $\mathcal{F}$ serait noté par :

$$\begin{array}{lrl} r_1: & arc(a, b) & \longleftarrow \\ r_2: & arc(b, c) & \longleftarrow \\ r_3: & chemin(X, Y) & \longleftarrow arc(X, Y) \\ r_4: & chemin(X, Y) & \longleftarrow arc(X, Z), chemin(Z, Y) \end{array}$$

où les quantifications universelles sont omises et la virgule dénote $\wedge$. $\diamondsuit$

1.4.5 Skolemisation

Nous avons vu dans l'exemple 1.79 que le théorème de Herbrand est faux pour des formules non universelles. Mais on peut construire, pour chaque formule F, une formule universelle F' qui est équisatisfaisable avec F : *i.e.* F est satisfaisable si et seulement si F' est satisfaisable. (Note : F' n'est pas équivalente à F, voir l'exercice 1.17.) Chaque formule

$$F = \forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_p G(x_1, \ldots, x_n, y_1, \ldots, y_p)$$

sera remplacée par

$$F' = \forall x_1 \cdots \forall x_n G(x_1, \ldots, x_n, f_1(x_1, \ldots, x_n), \ldots, f_p(x_1, \ldots, x_n)) \, ,$$

où $f_1, \ldots, f_p$ sont des nouveaux symboles de fonctions, appelés *fonctions de Skolem*. F' est appelée une *Skolemisation* de F.

Théorème 1.82 *Soit F une formule close du langage $\mathcal{L}$; il existe une formule universelle F' dans un langage $\mathcal{L}' = \mathcal{L} \cup \{f_1, \ldots, f_p\}$, où $f_1, \ldots, f_p$ sont des nouveaux symboles de fonctions, telle que F est satisfaisable si et seulement si F' est satisfaisable.*

Démonstration. Par le théorème 1.61 nous pouvons supposer que F est en forme prénexe ; supposons

$$F = \forall x_1 ... \forall x_{n_1} \exists y_1 \forall x_{n_1+1} ... \forall x_{n_2} \exists y_2 \cdots \forall x_{n_{p-1}+1} ... \forall x_{n_p} \exists y_p \forall x_{n_p+1} ... \forall x_n G$$

où G est une formule sans quantificateurs. Ajoutons p nouveaux symboles de fonctions $f_1, \ldots, f_p$ à $\mathcal{L}$; pour $i = 1, \ldots, p$, chaque f_i est d'arité n_i et dépend des x_j telles que $\forall x_j$ apparaît avant $\exists y_i$ dans F. F' est la formule

$$F' = \forall x_1 \cdots \forall x_n G[y_1 := f_1(x_1, \ldots, x_{n_1})] \cdots [y_p := f_p(x_1, \ldots, x_{n_p})].$$

F est satisfaisable si et seulement si F' est satisfaisable. Par induction sur p il suffit de prouver le lemme 1.83. $\square$

Lemme 1.83 *$F = \forall x_1 \cdots \forall x_n \exists y G$ est satisfaisable si et seulement si $F' = \forall x_1 \cdots \forall x_n G[y := f(x_1, \ldots, x_n)]$ est satisfaisable, où f est un nouveau symbole de fonction.*

Exercice 1.16 Prouver le lemme 1.83. $\diamondsuit$

Remarque 1.84 F' ne sera pas équivalente à F. Voir l'exercice 1.17.

Exercice 1.17 Trouver des Skolemisations de $F = (\forall x R(x)) \vee (\exists y R'(y))$. $\diamondsuit$

Exercice 1.18 Trouver des Skolemisations de

$$F = (\forall x \exists y R(x, y)) \vee \neg(\exists x \forall y R'(x, y)).$$ $\diamondsuit$

Pour prouver l'existence d'un modèle pour une formule, la Skolemisation nous permet de supprimer tous les quantificateurs existentiels. Par la Remarque 1.80, les modèles de $F(x_1, \ldots, x_n)$ et les modèles de $\forall x_1 \cdots \forall x_n F(x_1, \ldots, x_n)$ coïncident. Nous pouvons donc supposer que toutes les variables sont universellement quantifiées et omettre les quantificateurs universels de la dénotation de la formule : c'est la notation usuelle pour PROLOG.

1.4.6 Axiomes de Hilbert-Ackermann pour le calcul des prédicats

Comme les axiomes du calcul des séquents, les axiomes du système de Hilbert-Ackermann introduit dans la Section 1.2.7 peuvent être enrichis pour donner un système d'axiomes correct (ou cohérent) et complet pour le calcul des prédicats. Il suffit d'ajouter aux quatre axiomes déjà vus les deux axiomes suivants :

- $\forall x F \supset F[x := t]$ (instantiation),
- $F[x := t] \supset \exists x F$ (existence),

et les règles :

(i) Si σ est une substitution et si F est un théorème logique, alors $\sigma(F)$ est un théorème logique. La substitution σ doit satisfaire les conditions suivantes :

   (i1) on peut substituer une formule G à une variable propositionnelle p, à condition de substituer toutes les occurrences de p dans F, et que G ne contienne aucune variable qui serait liée dans F.

   (i2) on peut substituer un terme t à une variable x, à condition que t soit substituable à x dans F.

   (i3) on peut substituer une formule G à un prédicat n-aire $R(x_1, \ldots, x_n)$, à condition que G ait au moins n variables libres et que, en supposant que $z_1, \ldots, z_n, y_1, \ldots, y_k$ sont les variables libres de G, 1) aucun des y_j ne soit lié dans dans F, et 2) qu'aucune des occurrences de R dans F n'ait une variable ayant une occurrence liée dans G. On doit substituer toutes les occurrences de R dans F, on remplace une occurrence de R de la forme $R(t_1, \ldots, t_n)$ par $G(z_1 := t_1, \ldots, z_n := t_n, y_1, \ldots, y_k)$.

(ii) Si F et $(F \supset F')$ sont des théorèmes logiques, alors F' est un théorème logique. (*modus ponens*)

(iii) règles pour les quantificateurs :

  (iii1) Si $F \supset G$ est un théorème, et x n'est pas libre dans F, alors $F \supset \forall xG$ est un théorème.

  (iii2) Si $G \supset F$ est un théorème, et x n'est pas libre dans F, alors $\exists xG \supset F$ est un théorème.

(iv) on peut renommer les variables liées.

Exemple 1.85 Les conditions imposées pour les substitutions peuvent sembler très contraignantes : elles sont nécessaires comme le montrent les contrexemples suivants.

1) Si t n'est pas substituable à x dans F, l'axiome d'existence n'est pas valide, par exemple si $F(x) = \forall y\ p(x,y)$, $F[x := y] \supset \exists x\ F$ devient $\forall y p(y,y) \supset \exists x \forall y\ p(x,y)$, qui n'est pas valide, il suffit de choisir une structure où p est interprété par l'égalité pour s'en convaincre.

2) Si l'on ne respecte pas les conditions (i2), la substitution n'est pas correcte, par exemple, $p \supset \forall x(p \vee q(x))$ est un théorème, on peut remplacer p par $r(y)$ et on obtient encore un théorème $r(y) \supset \forall x(r(y) \vee q(x))$. Encore faut-il faire la substitution partout, car $r(y) \supset \forall x(p \vee q(x))$ n'est pas un théorème ; enfin, on ne peut pas remplacer p par $r(x)$, car $r(x) \supset \forall x(r(x) \vee q(x))$ n'est pas un théorème.

3) De même, si l'on ne respecte pas les conditions (i3), la substitution n'est pas correcte, par exemple,

1) $r(x) \supset \exists v \forall y(r(v) \vee q(x,y))$ est un théorème. Mais, on ne peut pas remplacer $r(x)$ par $G = q(z,y)$ car $q(z,y) \supset \exists v \forall y(q(v,y) \vee q(x,y))$ n'est plus un théorème, alors qu'on peut tout à fait remplacer $r(x)$ par $q(z,w)$: $q(z,w)[z := x] \supset \exists v \forall y(q(z,w)[z := v] \vee q(x,y))$ est le théorème $q(x,w) \supset \exists v \forall y(q(v,w) \vee q(x,y))$.

2) $q(x,y) \supset \exists x \exists v\ q(x,v)$ est un théorème, mais on ne peut pas remplacer $q(x,y)$ par $G = \exists v\ q'(x,y,v)$ car $G[x := x, y := y] \supset \exists x \exists v\ G[x := x, y := v]$ est la formule $\exists v\ q'(x,y,v) \supset \exists x \exists v \exists v\ q'(x,v,v)$ qui n'est pas un théorème.

Si toutefois on prend la précaution de renommer toutes les variables liées par de noms nouveaux, on évite ces problèmes.

Remarque 1.86 Les substitutions de formules à des prédicats peuvent être utiles dans le cas suivant par exemple : supposons avoir montré à partir d'axiomes des entiers que $\exists z \forall y\ z \leq y$ (il y a un élément minimum (zéro) pour l'ordre sur $\mathbb{N}$), qui se récrit comme formule logique $\exists z \forall y\ r(z,y)$; on veut ensuite remplacer le prédicat r qui représente $\leq$ par sa caractérisation $r(x,y)$ si et seulement si $\exists z\ y = x + z$, soit $\exists z\ p(y, som(x,z))$; un remplacement sans précaution de $r(z,y)$ par $\exists z p(y, som(x,z))$ dans $\exists z \forall y\ r(z,y)$ donnera $\exists z \forall y \exists z\ p(y, som(x,z))[x := z, y := y]$, soit $\exists z \forall y \exists z\ p(y, som(z,z))$, ou encore $\exists z \forall y \exists z\ y = z + z$, qui n'est pas valide du tout ! La condition (i3)2) de substitution est violée ici.

La correction des axiomes et règles du système de Hilbert-Ackermann est immédiate. Par exemple, la correction des deux axiomes résulte de la proposition 1.57. La complétude est un peu plus complexe. Nous ne la traiterons pas ici.

Exemple 1.87 Donnons un exemple de preuve à l'aide du système de Hilbert-Ackermann. Montrons que si F est un théorème, alors $\forall x\, F$ est aussi un théorème. Nous avons la suite de théorèmes :

$$\begin{array}{cl}
F \vee \overline{p \vee \bar{p}} & \text{(proposition 1.15 (ii))} \\
\overline{p \vee \bar{p}} \vee F & \text{(proposition 1.15 (iii))} \\
\overline{p \vee \bar{p}} \vee \forall x\, F & \text{(règle (iii1))} \\
p \vee \bar{p} & \text{(car } p \supset p \text{ , exemple 1.16)} \\
\forall x\, F & \text{(modus ponens)}
\end{array}$$

1.4.7 Les arbres sémantiques

La méthode des arbres sémantiques pour montrer qu'une formule G' est valide peut aussi être définie pour les formules du calcul des prédicats. Elle consiste à remarquer que G' est valide si et seulement si la clôture universelle G de G' est valide et ensuite

1. calculer $F = \neg G$; G est valide si et seulement si F est insatisfaisable.
2. Ensuite, mettre F sous forme clausale :
   - (2.i) mettre cette formule F sous forme prénexe,
   - (2.ii) éliminer les $\exists$ de la forme prénexe (Skolemisation),
   - (2.iii) mettre la formule obtenue en (2.ii) sous forme d'un ensemble de clauses $\mathcal{F}'$. (voir la définition 1.94 pour la définition des clauses)

$\mathcal{F}'$ est insatisfaisable si et seulement si F est insatisfaisable, et

3. utiliser la proposition suivante

Proposition 1.88 *Un ensemble de clauses $\mathcal{F}'$ est insatisfaisable si et seulement si $\mathcal{F}'$ admet un arbre sémantique fermé.*

Il y a toutefois deux modifications à apporter à la notion d'arbre sémantique. Un *arbre sémantique* est maintenant associé à une énumération $\{p_k\}$ des formules atomiques closes. A chaque nœud n d'un arbre sémantique est associée une interprétation de Herbrand partielle I_n : si n est un nœud de niveau i, c'est l'interprétation partielle définie en prenant la valeur 1 pour tous les littéraux qui étiquettent les arêtes menant de la racine au nœud n. Un nœud n de l' arbre sémantique A réfute la clause C de S s'il existe une substitution σ des variables de C par des termes clos telle que, dans I_n, et pour la valuation définie par σ on ait $\bar{\sigma}(C) = 0$.

Exemple 1.89 Considérons l'exemple suivant : soient les assertions

F_1 : certains étudiants assistent à tous les cours, et

F_2 : aucun étudiant n'assiste à un cours inintéressant.

Montrons que F_1 et F_2 impliquent l'assertion

F_3 : tous les cours sont intéressants.

Nous traitons cet exemple par trois méthodes : la méthode de la déduction naturelle en montrant que le séquent $\{F_1, F_2\} \vdash F_3$ est prouvable, puis la méthode des arbres sémantiques et enfin la méthode de résolution.

Formalisons d'abord le problème par des formules logiques :

$F_1 \quad : \quad \exists e\, \forall c\; ass(e, c)$

$F_2 \quad : \quad \forall c \left((\exists e\; ass(e, c)) \supset int(c) \right)$

$F_3 \quad : \quad \forall c\; int(c)$

exemple 1.89

1. Méthode de la déduction naturelle

Nous montrons d'abord deux lemmes :

Lemme 1.90 *Si $\mathcal{F} \vdash \forall x\, \neg F$ alors $\mathcal{F} \vdash \neg\exists x F$.*

Démonstration.

$$\begin{array}{lll}
1) & \mathcal{F} \vdash \forall x\, \neg F & \\
2) & \mathcal{F}, \exists x F \vdash \forall x\, \neg F & \text{(augmentation)} \\
3) & \mathcal{F}, \exists x F \vdash \exists x F & \text{(hypothèse)} \\
4) & \mathcal{F}, \exists x F \vdash \neg\forall x\, \neg F & \text{(définition de } \exists) \\
5) & \mathcal{F} \vdash \neg\exists x F & \text{(par l'absurde)} \quad \square
\end{array}$$

Lemme 1.91 *Si $F \vdash G$ et si x n'est pas libre dans G alors $\exists x F \vdash G$.*

Démonstration.

$$\begin{array}{lll}
1) & F \vdash G & \\
2) & F, \neg G \vdash G & \text{(augmentation)} \\
3) & F, \neg G \vdash \neg G & \text{(hypothèse)} \\
4) & \neg G \vdash \neg F & \text{(par l'absurde)} \\
5) & \neg G \vdash \forall x\, \neg F & \text{(généralisation car } x \text{ non libre dans } \neg G) \\
6) & \neg G \vdash \neg\exists x\, F & \text{(lemme 1.90)} \\
7) & \neg G, \exists x F \vdash \neg\exists x\, F & \text{(augmentation)} \\
8) & \neg G, \exists x F \vdash \exists x\, F & \text{(hypothèse)} \\
9) & \exists x F \vdash \neg\neg G & \text{(par l'absurde)} \\
10) & \exists x F \vdash G & \text{(double négation)} \quad \square
\end{array}$$

Lemme 1.92 $\exists e\, \forall c\; ass(e,c) \quad \vdash \quad \forall c\, \exists e\; ass(e,c)$

Démonstration.

$$\begin{array}{lll}
1) & \forall c\; ass(e,c), \forall e\, \neg ass(e,c) \vdash \forall c\; ass(e,c) & \text{hypothèse} \\
2) & \forall c\; ass(e,c), \forall e\, \neg ass(e,c) \vdash ass(e,c) & \text{instanciation} \\
3) & \forall c\; ass(e,c), \forall e\, \neg ass(e,c) \vdash \neg ass(e,c) & \text{hypothèse + instanciation} \\
4) & \forall c\; ass(e,c) \vdash \neg\forall e\, \neg ass(e,c) & \text{absurde} \\
5) & \forall c\; ass(e,c) \vdash \exists e\; ass(e,c) & \text{définition de } \exists \\
6) & \forall c\; ass(e,c) \vdash \forall c\, \exists e\; ass(e,c) & \text{généralisation} \\
 & & \text{car } c \text{ non libre dans } \forall c\; ass(e,c) \\
7) & \exists e\, \forall c\; ass(e,c) \vdash \forall c\, \exists e\; ass(e,c) & \text{lemme 1.91} \quad \square
\end{array}$$

Nous avons maintenant

$$\begin{array}{ll}
1)\ \exists e\, \forall c\; ass(e,c) \vdash \forall c\, \exists e\; ass(e,c) & \text{lemme 1.92} \\
2)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash \forall c\, \exists e\; ass(e,c) & \text{augmentation} \\
3)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash \exists e\; ass(e,c) & \text{instanciation} \\
4)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big) & \text{hypothèse} \\
5)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash \big(\exists e\; ass(e,c)\big) \supset int(c) & \text{instanciation} \\
6)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash int(c) & \text{modus ponens} \\
7)\ \forall c\, \Big(\big(\exists e\; ass(e,c)\big) \supset int(c)\Big)\ ,\ \exists e\, \forall c\; ass(e,c) \vdash \forall c\; int(c) & \text{généralisation} \\
 & \text{car } c \text{ non libre dans } F_1, F_2
\end{array}$$

La ligne 7 nous dit que $\{F_1, F_2\} \vdash F_3$.

exemple 1.89

2. Méthode des arbres sémantiques

Comme $F_1, F_2, \neg F_3$ sont déjà sous forme prénexe, on Skolemize $F_1, F_2, \neg F_3$ ce qui donne :

$F'_1 \quad : \quad \forall c\, ass(e_0, c)$

$F'_2 \quad : \quad \forall e\, \forall c\, \big(ass(e, c) \supset int(c)\big)$

$F'_3 \quad : \quad \neg int(c_0)$

On met F'_1, F'_2, F'_3 sous forme clausale ce qui donne :

$F'_1 \quad : \quad \forall c\, ass(e_0, c)$

$F''_2 \quad : \quad \forall e\, \forall c\, \big(\neg ass(e, c) \vee int(c)\big)$

$F'_3 \quad : \quad \neg int(c_0)$

On trouvera ci-dessous un arbre sémantique fermé pour l'ensembles de clauses $\mathcal{F} = \{F'_1, F''_2, F'_3\}$, associé à l'énumération $\{int(c_0)\,,\, ass(e, c_0)\}$.

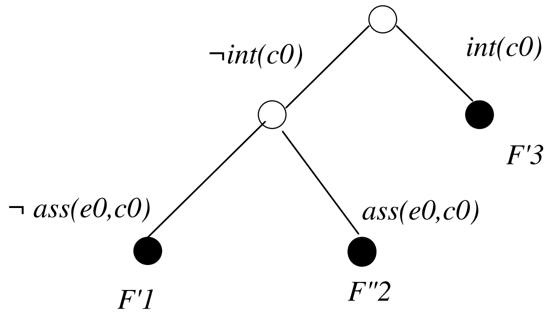

FIGURE 1.5

Soit $\{F_1, F_2, \ldots, F_n\}$ un ensemble de formules closes, la méthode des arbres sémantiques pour montrer que $\{F_1, F_2, \ldots, F_n\} \vdash F$ est valide, consiste donc : à remarquer d'abord que l'on peut remplacer F par sa clôture universelle puisque $\{F_1, F_2, \ldots, F_n\} \vdash F$ est valide si et seulement si $\{F_1, F_2, \ldots, F_n\} \vdash \forall x\, F$ est valide. Ensuite on remplace F par $\neg F$ et on montre que $\{F_1, F_2, \ldots, F_n, \neg F\}$ est insatisfaisable en appliquant les transformation suivantes :

(i) On met $\{F_1, F_2, \ldots, F_n, \neg F\}$ en forme prénexe

(ii) On Skolemize $\{F_1, F_2, \ldots, F_n, \neg F\}$ pour éliminer les quantificateurs existentiels

(iii) On met les formules résultantes sous forme clausale (en mettant les parties sans quantificateurs sous Forme Normale Conjonctive).

Cette méthode peut s'appliquer pour montrer la validité d'une formule quelconque, toujours en remarquant que $\vdash F$ est valide si et seulement si $\vdash \forall x\, F$ est valide, puis en appliquant les transformations (i), (ii), (iii) préconisées ci-dessus à la clôture universelle de F.

REMARQUE 1.93 On peut, pour simplifier la forme clausale, adopter un ordre différent pour les transformations (i), (ii), (iii) et faire des transformations intermédiaires. Par exemple, on peut adopter l'ordre suivant :

1. enlever les $\supset$ en remplaçant $F \supset G$ par $\neg F \vee G$,

2. réduire la portée des symboles de négation au maximum, c'est-à-dire faire “descendre” les négations dans les sous-formules de manière à ce que les seules négations soient de la forme $\neg A$, avec A formule atomique,

3. faire “descendre” les quantificateurs $\exists$ dans les sous-formules le plus possible,

4. Skolemiser,

5. mettre sous forme prénexe,

6. mettre sous FNC.

Par exemple, avec cette ordre, on transformera la formule $\forall x \exists y \neg(p(x) \wedge q(y))$ en $\forall x \exists y(\neg p(x) \vee \neg q(y))$ (par l'étape 2), puis par l'étape 3 en $\forall x(\neg p(x) \vee \exists y \neg q(y))$, et la Skolemisation donne : $\forall x(\neg p(x) \vee \neg q(a))$. Une application directe des transformations (i), (ii), (iii) donnerait ici $\forall x(\neg p(x) \vee \neg q(f(x)))$.

exemple 1.89

3. Méthode de SLD résolution

Nous traitons le même exemple avec la méthode de SLD résolution Un arbre de SLD résolution réfutant $F_1, F_2, \neg F_3$ est dessiné ci-dessous.

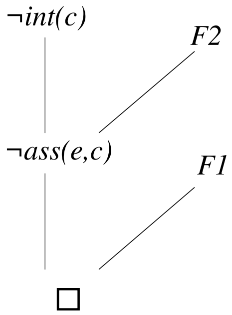

FIGURE 1.6

EXERCICE 1.19 Traiter l'exemple 1.89 avec le système de Hilbert-Ackermann. Indication : on pourra montrer les lemmes

1) si $F \supset (G \supset H)$ est un théorème, alors $G \supset (F \supset H)$ et $F \wedge G \supset H$ sont des théorèmes.

2) si $\forall x(F \supset G)$ est un théorème, alors $\forall x\, F \supset \forall x\, G$ est un théorème. $\diamondsuit$

1.4.8 Clauses de Horn

Les clauses de Horn sont un exemple très utile de formules universelles. PROLOG et la plupart des langages de programmation logiques sont basés sur les clauses de Horn.

Définition 1.94 *(i) Un littéral est une formule atomique ou la négation d'une formule atomique, i.e. une formule de la forme $L = R(t_1, \ldots, t_n)$ (littéral positif), ou de la forme $L = \neg R(t_1, \ldots, t_n)$ (littéral negatif).*

(ii) Une clause est une formule universelle de la forme

$$\forall x_1 \cdots \forall x_p (L_1 \vee \cdots \vee L_n)\,,$$

où les L_i sont des littéraux.

(iii) Une clause de Horn est une formule universelle de la forme

$$\forall x_1 \cdots \forall x_p (L_1 \vee \cdots \vee L_n)\,,$$

où les L_i sont des littéraux, et au plus un d'entre eux est positif.

(iv) Une clause de programme ou clause définie est une clause de Horn avec exactement un littéral positif.

Donc, une clause de Horn a l'une des trois formes suivantes :

(i) $\forall x_1 \cdots \forall x_p A$, (clause positive ou *fait*).

(ii) $\forall x_1 \cdots \forall x_p (\neg A_1 \vee \cdots \vee \neg A_n \vee A)$, où A et les A_i sont des formules atomiques.

(iii) $\forall x_1 \cdots \forall x_p (\neg A_1 \vee \cdots \vee \neg A_n)$, où les A_i sont des formules atomiques (clause négative ou *but*).

Exemple 1.95 Un programme PROLOG est constitué de clauses de Horn de la forme (i) ou (ii), habituellement écrites sous la forme (en omettant les quantificateurs universels et en substituant une virgule à $\wedge$) :

$$(i) \qquad A \longleftarrow$$
$$(ii) \qquad A \longleftarrow A_1, \ldots, A_n$$

Exemple 1.96 L'ensemble $\mathcal{F}$ de formules de l'exercice 1.15 est un ensemble de clauses de programme. $\mathcal{F}$ peut s'écrire $\mathcal{F} = \{arc(a,b)\,, arc(b,c)\,, \forall x \forall y \big(\neg arc(x,y) \vee chemin(x,y)\big) \;,\; \forall x \forall y \big(\neg arc(x,z) \vee \neg chemin(z,y) \vee chemin(x,y)\big)\}$, avec les notations PROLOG, $\mathcal{F}$ est notée par :

$$r_1 : \qquad arc(a,b) \longleftarrow$$
$$r_2 : \qquad arc(b,c) \longleftarrow$$
$$r_3 : \qquad chemin(X,Y) \longleftarrow arc(X,Y)$$
$$r_4 : \qquad chemin(X,Y) \longleftarrow arc(X,Z), chemin(Z,Y)$$

Théorème 1.97 *Un ensemble P de clauses de programme a un plus petit modèle de Herbrand $M = \langle U_H, I_M \rangle$ (i.e. un modèle de Herbrand tel que I_M est contenu dans tout autre modèle de Herbrand I_H).*

Démonstration. Soit $P = \{C_i \,/\, i \in J\}$, où chaque C_i est de la forme (i) ou (ii). Nous prouvons que l'intersection de tous les modèles de Herbrand de P est aussi un modèle de Herbrand de P. Soit M défini par

$$I_M = \bigcap \{I_H \;/\; \emptyset \models_{I_H} C_i \;,\; \text{pour toute } C_i \in P\}\,,$$

i.e. I_M est l'intersection de tous les modèles de Herbrand de P ; alors $\emptyset \models_M C_i$ pour toute $C_i \in P$. Nous vérifions que toutes les clauses C_i de P sont valides dans M.

(i) Si C_i est de la forme $\forall x_1 \cdots \forall x_p A$, alors toutes les instances closes de A sont vraies dans tous les modèles de Herbrand, donc elles sont dans tous les I_H, et aussi dans I_M, et donc elles sont vraies dans M.

(ii) Si C_i est de la forme $\forall x_1 \cdots \forall x_p (\neg A_1 \vee \cdots \vee \neg A_n \vee A)$, posons $C'_i = (\neg A_1 \vee \cdots \vee \neg A_n \vee A)$; soit v une valuation $v\colon \{x_1, \ldots, x_p\} \longrightarrow U_H$, et soit, pour $B \in \{A, A_1, \ldots, A_n\}$, $v^*(B) = B[x_1 := v(x_1)] \cdots [x_p := v(x_p)]$ l'atome clos obtenu en substituant $v(x_i)$ à x_i dans B ; alors

- soit il existe un A_j tel que $v^*(A_j) \notin I_M$, et alors $\bar{v}(\neg A_j) = 1$ et $\bar{v}(C'_i) = 1$.
- soit pour tout A_j, $v^*(A_j) \in I_M$, et alors, pour toute I_H définissant un modèle de Herbrand H de P : $v^*(A_j) \in I_H$, et comme H est un modèle de Herbrand de P, nous avons aussi que $v^*(A) \in I_H$, d'où $v^*(A) \in I_M$, et donc $\bar{v}(A) = 1$ et $\bar{v}(C'_i) = 1$.

Donc, pour toute valuation $v: \{x_1, \ldots, x_p\} \longrightarrow U_H$, $\bar{v}(C'_i) = 1$, et donc $\emptyset \underset{M}{\models} C_i$. □

Exemple 1.98 Le plus petit modèle de Herbrand du programme P de l'Exemple 1.96 est défini par

$$I_M = \{arc(a,b)\,,\ arc(b,c)\,,\ chemin(a,b)\,,\ chemin(b,c), chemin(a,c)\}\,.$$

Remarque 1.99 1) Tout ensemble P de clauses de programme a aussi un plus grand modèle de Herbrand M', qui est défini par la base de Herbrand B_H. Voir aussi l'exercice 1.25 (2).

2) Un ensemble $\mathcal{F}$ de formules universelles qui ne sont pas des clauses de Horn peut avoir plusieurs modèles de Herbrand minimaux incomparables entre eux et donc pas de plus petit modèle de Herbrand. Par exemple, soit $\mathcal{F} = \{\forall x\big(p(x) \vee q(x)\big)\}$, où $\mathcal{L} = \{a, p, q\}$, a un symbole de constante, et p, q des symboles de relations d'arité 1. Alors $U_H = \{a\}$, $B_H = \{p(a)\,,\ q(a)\}$, et $\mathcal{F}$ a trois modèles de Herbrand, respectivement définis par les sous-ensembles $I_1 = \{p(a)\}$, $I_2 = \{q(a)\}$, et $I_3 = B_H$; les modèles I_1 et I_2 sont minimaux (aucun n'est inclus dans l'autre), leur intersection est l'interprétation de Herbrand définie par $I_H = \emptyset$, qui n'est *pas* un modèle de $\mathcal{F}$.

Exercice 1.20 Un ensemble de formules qui ne sont pas universelles peut aussi avoir des modèles de Herbrand minimaux. Trouver les modèles de Herbrand minimaux de l'ensemble de formules $\mathcal{F}$ de l'exercice 1.14. ♢

Exercice 1.21 Trouver le plus petit modèle de Herbrand de l'ensemble de clauses de programme $P = \{\forall x \forall y\big(\neg arc(x,y) \vee chemin(x,y)\big)\,, \forall x \forall y\big(\neg arc(x,z) \vee \neg chemin(z,y) \vee chemin(x,y)\big)\}$. ♢

Exercice 1.22 Trouver le plus petit modèle de Herbrand de l'ensemble de clauses de programme $P = \{i(a)\ ,\ \forall x\big(i(s(x)) \vee \neg i(x)\big)\}$, où $\mathcal{L} = \{a, s, i\}$, a est un symbole de constante, s est un symbole de fonction unaire, et i est un symbole de relation unaire. ♢

Exercice 1.23 Soit P un ensemble de clauses de programme ; P est un ensemble de formules, donc (cf. définition 1.65 et exemple 1.66) $Th(P) = \{F \,/\, P \vdash F\}$ est une théorie.

Montrer que $\{A \in B_H \,/\, A \in Th(P)\}$ définit le plus petit modèle de Herbrand de l'ensemble de clauses de programme P. ♢

Exercice 1.24 Est-ce que tout ensemble $\mathcal{F}$ de clauses de Horn a un plus petit modèle de Herbrand ? ♢

Il y a une preuve constructive de l'existence du plus petit modèle de Herbrand d'un ensemble de clauses de programme, qui est très utile en programmation logique, et qui fait l'objet de l'exercice suivant.

Exercice 1.25 Rappelons qu'un treillis complet est un treillis où tout sous-ensemble a une borne inférieure et une borne supérieure. Si f est une application monotone d'un treillis complet dans lui-même, alors on peut prouver que f a un plus petit point fixe défini par $e = \inf\{x \in E \,/\, f(x) \leq x\}$.

Soit P un ensemble de clauses de programmes. Soit $\mathcal{P}(B_H)$ l'ensemble des parties de la base de Herbrand B_H. Alors $\mathcal{P}(B_H)$ muni de l'inclusion est un treillis complet. Le plus petit élément de $\mathcal{P}(B_H)$ est $\emptyset$, son plus grand élément est B_H, $\sup_i K_i = \cup_i K_i$, $\inf_i K_i = \cap_i K_i$.

L'opérateur de conséquence immédiate $T_P\colon \mathcal{P}(B_H) \longrightarrow \mathcal{P}(B_H)$ est défini par : $T_P(I) = \{A \in B_H |$ il existe $r = (B_1, \ldots, B_n \Longrightarrow B) \in P$, il existe une valuation $s\colon X \longrightarrow U_H$ telle que pour $i = 1, \ldots, n$, $s^*(B_i) = A_i \in I$, $s^*(B) = A\}$. $s^*(B) = B[x_1 := s(x_1)] \cdots [x_p := s(x_p)]$ (resp. $s^*(B_i) = B_i[x_1 := s(x_1)] \cdots [x_p := s(x_p)]$) dénote l'atome clos obtenu en substituant le terme $s(x_k)$ à x_k dans B (resp. B_i), pour toute variable $x_k \in X$.

En d'autres termes, $T_P(I)$ est l'ensemble des formules atomiques A telles que $A_1, \ldots, A_n \Longrightarrow A$ est une instance close d'une clause r de P et de plus $A_1, \ldots, A_n$ sont dans I.

1) Montrer que T_P est monotone.

2) Soit I une interprétation de Herbrand ; montrer que I est un modèle de P si et seulement si $T_P(I) \subset I$.

3) Montrer que le plus petit point fixe de T_P est le plus petit modèle de Herbrand de P.

4) Montrer que T_P est continue (*i.e.* pour toute suite croissante $K_1 \subset K_2 \subset \cdots \subset K_n \subset \cdots$ de $\mathcal{P}(B_H)$, $\sup_i T_P(K_i) = T_P(\sup_i K_i)$).

5) Montrer que le plus petit modèle de Herbrand de P est défini par le sous-ensemble

$$I_M = \sup(\{T_P^n(\emptyset) \,/\, n \in \mathbb{N}\}).$$

6) Montrer que, pour tout $n \in \mathbb{N}$, $T_P^n(B_H)$ est un modèle de Herbrand de P.

Soit $K = \inf(\{T_P^n(B_H) \,/\, n \in \mathbb{N}\})$.

7) K est-il un modèle de P ?

8) K est-il un point fixe de P ? Que pouvez-vous dire du plus grand point fixe de P ? $\diamondsuit$