

Logique et démonstration automatique :
Une introduction à la logique propositionnelle et à la logique du
premier ordre

Stéphane Devismes

Pascal Lafourcade

Michel Lévy

2017/2018

Table des matières

I	Logique propositionnelle	7
1	Logique propositionnelle	9
1.1	Syntaxe	10
1.1.1	Formules strictes	10
1.1.2	Formules à priorité	12
1.2	Sens des formules	13
1.2.1	Sens des connecteurs	13
1.2.2	Valeur d'une formule	14
1.2.3	Définitions et notions élémentaires de logique	14
1.2.4	Compacité	18
1.2.5	Équivalences remarquables	19
1.3	Substitution et remplacement	19
1.3.1	Substitution	20
1.3.2	Remplacement	21
1.4	Formes Normales	22
1.4.1	Transformation en forme normale	23
1.4.2	Transformation en forme normale disjonctive (somme de monômes)	23
1.4.3	Transformation en forme normale conjonctive (produit de clauses)	25
1.5	Algèbre de Boole	25
1.5.1	Définition et notations	26
1.5.2	Propriétés	27
1.5.3	Dualité	30
1.6	Fonctions booléennes	31
1.6.1	Fonctions booléennes et somme de monômes	32
1.6.2	Fonctions booléennes et produit de clauses	33
1.7	L'outil BDDC	34
1.8	Exercices	35
2	Résolution propositionnelle	41
2.1	Résolution	41
2.1.1	Définitions	41
2.1.2	Cohérence	44
2.1.3	Complétude pour la réfutation	45
2.1.4	Réduction d'un ensemble de clauses	47
2.1.5	Clauses minimales pour la déduction	48
2.1.6	Clauses minimales pour la conséquence	48
2.2	Stratégie complète	50
2.2.1	Algorithme de la stratégie complète	50
2.2.2	Arrêt de l'algorithme	52
2.2.3	Le résultat de l'algorithme est équivalent à l'ensemble initial de clauses	53
2.2.4	Le résultat de l'algorithme est l'ensemble des clauses minimales pour la déduction de l'ensemble initial de clauses	53
2.3	Algorithme de Davis-Putnam-Logemann-Loveland	55
2.3.1	Suppression des clauses contenant des littéraux isolés	55

2.3.2	Résolution unitaire	56
2.3.3	Suppression des clauses valides	57
2.3.4	L'algorithme	57
2.3.5	Solveurs SAT	60
2.3.5.1	Heuristique de branchement	60
2.3.5.2	Ajout de clauses	60
2.3.5.3	Analyse des conflits et retour-arrière non chronologique	61
2.3.5.4	Apprentissage	61
2.3.5.5	Redémarrage	61
2.3.5.6	Structures de données paresseuses	61
2.4	Exercices	62
3	Déduction Naturelle	65
3.1	Système formel de la déduction naturelle	65
3.1.1	Règles de conjonction	66
3.1.2	Règles de disjonction	66
3.1.3	Règles de l'implication	66
3.1.4	Deux règles spéciales	67
3.1.5	Preuves en déduction naturelle	67
3.1.5.1	Brouillon de preuve	68
3.1.5.2	Contexte des lignes d'un brouillon de preuve	68
3.1.5.3	Preuves	69
3.2	Tactiques de preuve	71
3.3	Cohérence de la déduction naturelle	73
3.4	Complétude de la déduction naturelle	73
3.5	Outils	75
3.5.1	Logiciel de construction automatique de preuves	76
3.5.2	Dessiner des arbres de preuves	76
3.6	Exercices	77
II	Logique du premier ordre	81
4	Logique du premier ordre	83
4.1	Syntaxe	83
4.1.1	Formules strictes	84
4.1.2	Formules à priorité	85
4.2	Être libre ou lié	87
4.2.1	Occurrences libres et liées	87
4.2.2	Variables libres et liées	88
4.3	Sens des formules	88
4.3.1	Déclaration de symbole	88
4.3.2	Signature	88
4.3.3	Interprétation	90
4.3.4	Sens des formules	90
4.3.4.1	Sens des termes sur une signature	91
4.3.4.2	Sens des formules atomiques sur une signature	91
4.3.4.3	Sens des formules sur une signature	92
4.3.5	Modèle, validité, conséquence, équivalence	93
4.3.6	Instanciation	94
4.3.7	Interprétation finie	94
4.3.7.1	Les entiers et leurs représentations	95
4.3.7.2	Expansion d'une formule	95
4.3.7.3	Interprétation et assignation propositionnelle	95
4.3.7.4	Recherche d'un modèle fini d'une formule fermée	96
4.3.8	Substitution et remplacement	98

4.4	Équivalences remarquables	98
4.4.1	Relation entre \forall et \exists	98
4.4.2	Déplacement des quantificateurs	99
4.4.3	Changement de variables liées	100
4.5	Exercices	102
5	Base de la démonstration automatique	107
5.1	Méthodes de Herbrand	107
5.1.1	Domaine et base de Herbrand	107
5.1.2	Interprétation de Herbrand	108
5.1.3	Théorème de Herbrand	111
5.2	Skolémisation	113
5.2.1	Algorithme de skolémisation d'une formule	113
5.2.1.1	Transformation en formule normale	114
5.2.1.2	Transformation en formule propre	115
5.2.1.3	Élimination des quantificateurs existentiels	115
5.2.1.4	Transformation en fermeture universelle	116
5.2.2	Propriétés de la forme de Skolem	117
5.2.3	Forme clausale	118
5.3	Unification	120
5.3.1	Unificateur	120
5.3.2	Algorithme d'unification	122
5.3.2.1	Les règles de l'algorithme	122
5.3.2.2	Correction de l'algorithme	123
5.3.2.3	Terminaison de l'algorithme	123
5.4	Résolution au premier ordre	124
5.4.1	Trois règles pour la résolution	124
5.4.1.1	Factorisation	124
5.4.1.2	Copie d'une clause	124
5.4.1.3	Résolution binaire	125
5.4.1.4	Preuve par factorisation, copie et résolution binaire	126
5.4.2	Cohérence de la résolution	126
5.4.3	Complétude de la résolution	127
5.5	Exercices	131
6	Déduction naturelle au premier ordre : quantificateurs, copie et égalité	135
6.1	Règles pour la logique du premier ordre	135
6.1.1	Règles des quantificateurs	135
6.1.1.1	Règles pour le quantificateur universel	135
6.1.1.2	Règles pour le quantificateur existentiel	137
6.1.2	Copie	139
6.1.3	Les règles de l'égalité	139
6.2	Tactiques de preuves	140
6.2.1	Raisonnement en avant avec une hypothèse d'existence	140
6.2.2	Raisonnement en arrière pour généraliser	140
6.2.3	Un exemple d'application des tactiques	140
6.3	Cohérence du système	142
6.4	Exercices	144
III	Annexes	147
	Corrigés	149

Introduction

LES mathématiciens ont raisonné *correctement* durant des siècles *sans connaître* la logique formelle. La logique moderne est née de l'ambition de formaliser (mécaniser) le raisonnement mathématique : ce projet a reçu une impulsion décisive quand il est apparu que l'absence de formalisation pouvait conduire à des contradictions. Le développement actuel de la logique continue avec

- le besoin de prouver la correction des programmes (particulièrement quand ces programmes sont utilisés dans des domaines où la sécurité est en jeu)
- l'ambition de représenter en machine l'ensemble des connaissances mathématiques

Cependant la terminologie des mathématiciens est restée différente de celle des logiciens. Pour exprimer que p implique q , le mathématicien dira par exemple :

- p entraîne q
- p est une condition suffisante de q
- pour que q soit vrai, il suffit que p soit vrai
- q est une condition nécessaire de p
- pour que p soit vrai, il faut que q soit vrai

Nous restreindrons notre étude à la logique *classique* (par opposition à la logique intuitionniste). La logique classique est la logique à deux valeurs de vérité. De plus cette logique est celle des circuits combinatoires, ce qui explique sa grande importance pratique.

Plan : Nous présentons dans la première partie la logique propositionnelle. Plus précisément dans un premier chapitre nous donnons les définitions et résultats de base de la logique des prédicats. Dans le second chapitre nous parlons de la résolution propositionnelle en introduisant la résolution binaire, la stratégie complète et l'algorithme DPLL. Enfin nous illustrons une méthode de raisonnement logique en présentant la déduction naturelle. Dans la seconde partie du cours nous revisitons l'ensemble des notions, résultats et techniques présentés dans la première partie pour la logique du premier ordre.

Exercices : Nous proposons à la fin de chaque chapitre une série d'exercices portant sur le contenu du chapitre. Nous indiquons par des étoiles la difficulté des exercices proposés, plus il y a d'étoiles plus l'exercice est difficile. Nous indiquons par \Leftrightarrow les exercices qui complètent les preuves vues durant le cours. Les exercices proposés sont de trois catégories. Nous avons des exercices :

- Basiques qui aident l'étudiant à se familiariser avec le vocabulaire et à manipuler les notions introduites en cours.
- Techniques qui sont des applications directes ou immédiates des résultats vus en cours.
- Réflexifs qui permettent à l'apprenant à raisonner, démontrer, prouver des résultats à partir des techniques et notions apprises grâce aux deux autres types d'exercices.

Objectifs : Les compétences et connaissances que nous souhaitons transmettre sont les suivantes :

- *Comprendre un raisonnement* : être capable de déterminer si un raisonnement logique est correct ou non.
- *Raisonner*, c'est-à-dire, construire un raisonnement correct utilisant les outils de la logique propositionnelle et du premier ordre.
- *Modéliser et formaliser un problème*.
- *Écrire une preuve rigoureuse*.

Première partie

Logique propositionnelle

Chapitre 1

Logique propositionnelle

Sommaire

1.1	Syntaxe	10
1.1.1	Formules strictes	10
1.1.2	Formules à priorité	12
1.2	Sens des formules	13
1.2.1	Sens des connecteurs	13
1.2.2	Valeur d'une formule	14
1.2.3	Définitions et notions élémentaires de logique	14
1.2.4	Compacité	18
1.2.5	Équivalences remarquables	19
1.3	Substitution et remplacement	19
1.3.1	Substitution	20
1.3.2	Remplacement	21
1.4	Formes Normales	22
1.4.1	Transformation en forme normale	23
1.4.2	Transformation en forme normale disjonctive (somme de monômes)	23
1.4.3	Transformation en forme normale conjonctive (produit de clauses)	25
1.5	Algèbre de Boole	25
1.5.1	Définition et notations	26
1.5.2	Propriétés	27
1.5.3	Dualité	30
1.6	Fonctions booléennes	31
1.6.1	Fonctions booléennes et somme de monômes	32
1.6.2	Fonctions booléennes et produit de clauses	33
1.7	L'outil BDDC	34
1.8	Exercices	35

A RISTOTE fut un des premiers à essayer de formaliser le raisonnement en utilisant la logique des syllogismes. La logique sert à préciser ce qu'est un raisonnement correct, indépendamment du domaine d'application. Un raisonnement est un moyen d'obtenir une conclusion à partir d'hypothèses données. Un raisonnement *correct* ne dit rien sur la vérité des hypothèses, il dit seulement qu'à partir de *la vérité des hypothèses, nous pouvons déduire la vérité de la conclusion*. Nous commençons l'étude de la logique par les lois de la logique propositionnelle. La logique propositionnelle est la logique *sans quantificateurs* qui s'intéresse uniquement aux lois gouvernant les opérations logiques suivantes : la négation (\neg), la conjonction, autrement dit le « et » (\wedge), la disjonction, autrement dit le « ou » (\vee), l'implication (\Rightarrow) et l'équivalence (\Leftrightarrow). Ces opérations sont également appelées *connecteurs*. La logique propositionnelle permet de construire des raisonnements à partir de ces connecteurs. Considérons l'exemple suivant qui comporte trois hypothèses :

1. si Pierre est grand, alors Jean n'est pas le fils de Pierre,
2. si Pierre n'est pas grand, alors Jean est le fils de Pierre,

3. si Jean est le fils de Pierre alors Marie est la sœur de Jean.

Nous concluons que Marie est la sœur de Jean ou Pierre est grand.

Afin de pouvoir raisonner nous extrayons la structure logique des hypothèses. Nous désignons les phrases « Pierre est grand », « Jean est le fils de Pierre », « Marie est la sœur de Jean » respectivement par les lettres p, j, m . Les hypothèses peuvent donc s'écrire :

1. $p \Rightarrow \neg j$,
2. $\neg p \Rightarrow j$,
3. $j \Rightarrow m$,

et la conclusion se formalise en $m \vee p$. Nous montrons alors que les hypothèses impliquent la conclusion indépendamment de la nature des énoncés p, j, m . Pour cela nous prouvons que la formule suivante est vraie quelle que soit la vérité des propositions p, j, m .

$$((p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m)) \Rightarrow (m \vee p).$$

Plan : Nous débutons ce chapitre par la *syntaxe des formules logiques*, c'est-à-dire, les règles permettant d'écrire des formules. Une formule peut être vraie ou fausse, ainsi nous devons être capable d'évaluer le *sens d'une formule*. Pour cela nous introduisons le sens de chaque connecteur ainsi que le calcul de la valeur d'une formule qui en dérive. Nous montrons alors un résultat de compacité qui sera principalement utilisé dans la seconde partie de ce livre. Ensuite nous présentons des *équivalences remarquables* utiles pour simplifier les raisonnements logiques. D'autres méthodes permettent de simplifier les raisonnements logiques, par exemple le *remplacement* et la *substitution* de formule. Nous montrons ensuite comment construire les formes normales conjonctives ou disjonctives d'une formule en utilisant les équivalences remarquables. Ces formes normales permettent d'exhiber facilement les modèles ou les contre-modèles d'une formule. Nous montrons ensuite que la logique propositionnelle est une instance d'une *algèbre de Boole*. Nous introduisons la notion de *fonctions booléennes*. Enfin, nous présentons succinctement l'outil BDDC¹ développé par Pascal Raymond. Cet outil permet de manipuler les formules propositionnelles.

1.1 Syntaxe

Avant de raisonner, nous définissons le langage que nous utilisons. Ce langage est celui des formules construites à partir du *vocabulaire* suivant :

- Les constantes : \top et \perp représentant respectivement le *vrai* et le *faux*.
- Les variables : une variable est un identificateur, avec ou sans indice, par exemple x, y_1 .
- Les parenthèses : ouvrante (et fermante).
- Les connecteurs : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ respectivement appelés négation, disjonction (ou), conjonction (et), implication et équivalence.

La syntaxe définit les règles de construction d'une formule de la logique propositionnelle. Nous introduisons deux modes d'écriture d'une formule, l'un strict, l'autre plus souple dans le sens qu'il autorise plusieurs écritures d'une même formule. Cette souplesse est obtenue grâce à l'introduction de priorités entre les connecteurs logiques.

1.1.1 Formules strictes

Ci-dessous, nous donnons les règles de construction d'une formule stricte à partir du vocabulaire donné précédemment.

Définition 1.1.1 (Formule stricte) Une formule stricte est définie de manière inductive comme suit :

- \top et \perp sont des formules strictes.
- Une variable est une formule stricte.
- Si A est une formule stricte alors $\neg A$ est une formule stricte.
- Si A et B sont des formules strictes et si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule stricte.

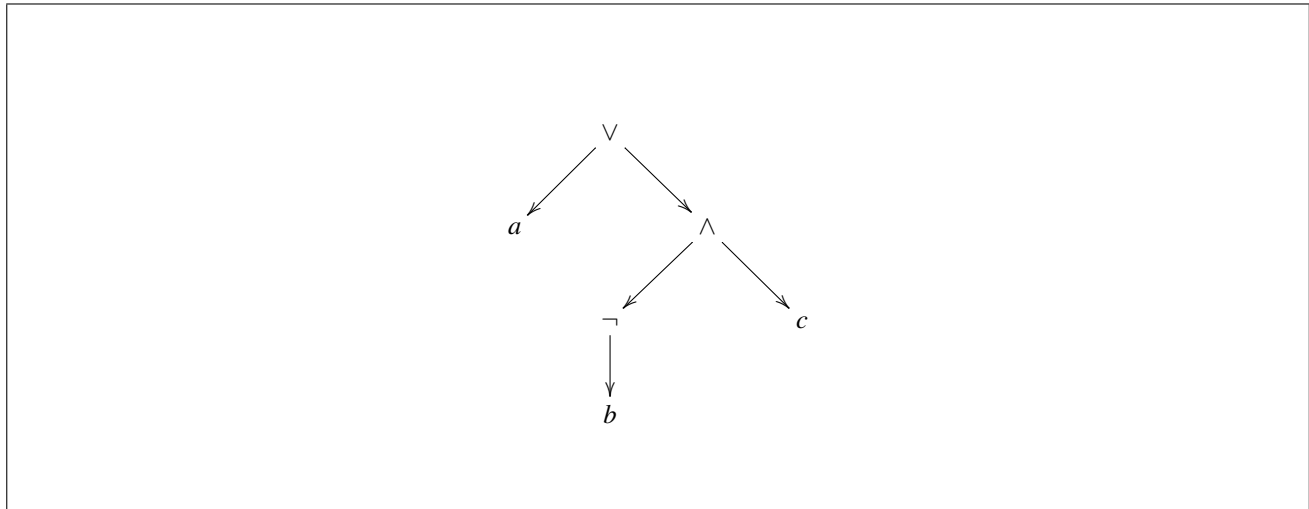
Dans la suite, nous désignons par \circ tout connecteur binaire, et nous appelons simplement formule une formule stricte. Les formules différentes de \top, \perp et des variables sont des formules décomposables.

Exemple 1.1.2 L'expression $(a \vee (\neg b \wedge c))$ est une formule stricte construite suivant les règles précédentes. En revanche, $a \vee (\neg b \wedge c)$ et $(a \vee (\neg(b) \wedge c))$ ne sont pas des formules au sens de la définition 1.1.1.

1. <http://www-verimag.imag.fr/~raymond/index.php/bddc-manual/>.

L'intérêt de la définition de formules strictes est que les parenthèses permettent de trouver sans ambiguïté la structure des formules. Nous représentons la structure des formules par un arbre, où les feuilles contiennent les constantes ou les variables et les nœuds les connecteurs logiques. Le nœud racine est le connecteur à appliquer en dernier.

Exemple 1.1.3 La structure de la formule $(a \vee (\neg b \wedge c))$ est mise en évidence par l'arbre suivant :



Une formule peut être vue comme une liste de symboles (connecteurs, parenthèses, variables, et constantes). Un facteur d'une telle liste est une suite de symboles consécutifs dans la liste.

Définition 1.1.4 (Sous-formule) Nous appelons sous-formule d'une formule (stricte) A tout facteur de A qui est une formule (stricte).

Exemple 1.1.5 $(\neg b \wedge c)$ est une sous-formule de $(a \vee (\neg b \wedge c))$.

Nous montrons que les formules sont décomposables d'une façon unique en leurs sous-formules. Ce résultat, précisé par le théorème 1.1.13 page suivante, est évident sur les exemples. Mais la preuve de ce résultat nécessite de nombreux résultats intermédiaires prouvés par récurrence sur la longueur d'une formule (définie ci-après). L'unicité de la décomposition implique que nous pouvons identifier une formule et son arbre de décomposition. Ainsi, une sous-formule de la formule A pourra être identifiée comme un sous-arbre de l'arbre représentant la formule A .

Définition 1.1.6 (Longueur d'une formule) La longueur d'une formule A est le nombre de symboles utilisés pour écrire A , dénotée $l(A)$.

Si nous voyons une formule comme un mot sur le vocabulaire dont les éléments sont les constantes, les variables, les parenthèses et les connecteurs. Un mot sur ce vocabulaire est une suite d'éléments sur ce vocabulaire et la longueur du mot est la longueur de la suite.

Exemple 1.1.7 Soient la formule $A = (a \vee b)$ et $B = (A \wedge \neg A)$, nous avons $l(A) = 5$ et $l(B) = 4 + 5 + 5 = 14$.

Lemme 1.1.8 (Équilibre des parenthèses) Toute formule a un nombre égal de parenthèses ouvrantes et de parenthèses fermantes.

Preuve : Par définition des formules, toute parenthèse ouvrante est associée à une parenthèse fermante. Le lemme s'en déduit immédiatement (par une récurrence trop simple pour mériter d'être explicitée). \square

Lemme 1.1.9 (Relation entre les parenthèses) Tout préfixe d'une formule a un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes.

Preuve : Supposons le lemme vérifié pour toute formule de longueur inférieure à n . Soit A une formule de longueur n . Montrons que le lemme est vrai pour A .

1. Soit A une variable ou une constante. Le lemme est vérifié.

2. Soit $A = \neg B$ où B est une formule. Un préfixe de A est vide ou s'écrit $\neg B'$, où B' est un préfixe de B . Par hypothèse de récurrence, B' a un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes. Par suite il en est de même de tout préfixe de A .
3. Soit $A = (B \circ C)$ où B et C sont des formules. Un préfixe de A est soit vide, soit s'écrit $(B'$ où B' est un préfixe de B , soit s'écrit $(B \circ C'$ où C' est un préfixe de C , soit est égal à A . Examinons ces différents cas.
 - (a) Le préfixe vide a un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes.
 - (b) Par hypothèse de récurrence, B' a un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes, donc il en est de même de $(B'$.
 - (c) Par hypothèse de récurrence, B et C' ont un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes, donc il en est de même de $(B \circ C'$.
 - (d) D'après le lemme 1.1.8, le nombre de parenthèses ouvrantes de A est égal (donc au moins égal) au nombre des parenthèses fermantes.

Ainsi, si le lemme est vrai pour toute formule de longueur inférieure à n , il est vrai pour toute formule de longueur n . Donc, par récurrence, il est vrai pour une formule de longueur quelconque. \square

Afin de raisonner par induction sur la structure d'une formule, nous définissons la taille d'une formule. Nous remarquons que la taille d'une formule correspond au nombre de connecteurs qu'elle contient.

Définition 1.1.10 (Taille d'une formule) La taille d'une formule A , notée $|A|$, est définie inductivement par :

- $|\top| = 0$ et $|\perp| = 0$.
- Si A est une variable alors $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Exemple 1.1.11 $|(a \vee (\neg b \wedge c))| =$

3.

La *taille* des formules définie dans la définition 1.1.10 page précédente est une mesure utile pour prouver par récurrence des propriétés sur les formules. La preuve du lemme 1.1.12 illustre comment écrire une preuve par récurrence sur la taille des formules. Rappelons qu'un préfixe strict d'une formule A est un préfixe de longueur strictement plus petite que la longueur de A .

Lemme 1.1.12 (Préfixe strict) Tout préfixe strict d'une formule n'est pas une formule.

Preuve : Nous effectuons une preuve par induction sur la taille de la formule :

Cas de base : Soit A une formule de taille 0, A est donc soit une variable soit une constante. Le seul préfixe strict de A est le mot vide qui n'est pas une formule.

Induction : Supposons le lemme vérifié pour toute formule de taille inférieure à n , avec $n > 0$. Soit A une formule de taille n . Montrons que le lemme est vrai pour A .

- Soit $A = \neg B$, où B est une formule. Supposons par contradiction que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $\neg B'$ où B' est une formule et un préfixe strict de B . C'est impossible d'après l'hypothèse de récurrence.
- Soit $A = (B \circ C)$ où B et C sont deux formules. Supposons par contradiction que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $(B' \circ C')$ où B' et C' sont des formules. La formule B est un préfixe de la formule B' ou la formule B' est un préfixe de la formule B . Il est impossible, d'après l'hypothèse de récurrence, que ces préfixes soient stricts. Donc $B = B'$. Par suite C' est un préfixe de C . D'après le lemme 1.1.8 page précédente, C' a autant de parenthèses ouvrantes que de fermantes. Donc C' est un préfixe de la formule C , qui a plus de parenthèses fermantes que de parenthèses ouvrantes, ce qui contredit le lemme 1.1.9 page précédente. Donc A n'a pas de préfixe strict qui est une formule.

Ainsi par récurrence, tout préfixe strict d'une formule n'est pas une formule. \square

Théorème 1.1.13 Pour toute formule A , un et un seul de ces cas se présente :

- A est une variable,
- A est une constante,
- A s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Preuve : Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules. Il suffit de montrer l'unicité de la dernière décomposition. Supposons que $A = (B \circ C) = (B' \circ C')$ où B, B', C, C' sont des formules. La formule B est préfixe de B' ou la formule B' est préfixe de B . D'après le lemme 1.1.12, il est impossible que ce soient des préfixes stricts, donc $B = B'$ et par suite $C = C'$. \square

Avec la définition de formules (strictes) nous écrivons de nombreuses parenthèses inutiles comme les parenthèses qui entourent chaque formule. Nous introduisons maintenant plus de souplesse dans notre syntaxe en définissant des priorités.

1.1.2 Formules à priorité

Pour éviter la surabondance des parenthèses, nous définissons les *formules à priorité*.

Définition 1.1.14 (Formule à priorité) Une formule à priorité est définie inductivement par :

- \top et \perp sont des formules à priorité,
- une variable est une formule à priorité,
- si A est une formule à priorité alors $\neg A$ est une formule à priorité,
- si A et B sont des formules à priorité alors $A \circ B$ est une formule à priorité,
- si A est une formule à priorité alors (A) est une formule à priorité.

Exemple 1.1.15 Considérons la formule $a \vee \neg b \wedge c$ qui est une formule à priorité mais pas une formule.

En général, une formule à priorité n'est pas une formule (stricte). Nous montrons dans l'exercice 2 page 35 que toute formule est une formule à priorité. Afin de pouvoir supprimer des parenthèses sans aucune ambiguïté nous définissons un ordre de priorité entre les différents connecteurs.

Définition 1.1.16 (Ordre de priorité des connecteurs) La négation est prioritaire, puis dans l'ordre des priorités décroissantes, nous trouvons la conjonction (\wedge), la disjonction (\vee), l'implication (\Rightarrow) et l'équivalence (\Leftrightarrow).

À priorité égale, le connecteur gauche est prioritaire, **sauf pour l'implication qui est associative à droite**.²

Nous considérons qu'une formule à priorité est l'*abréviation* de la formule reconstituable en utilisant les priorités. Sauf exception, nous identifions une formule et son abréviation. Autrement dit, ce qui nous intéresse dans une formule, ce n'est pas son écriture *superficielle*, c'est sa structure, qui est mise en évidence par la syntaxe « stricte ». Ainsi la taille d'une formule à priorité sera égale à la taille de la formule stricte dont elle est l'abréviation. De même, pour les sous-formules, nous considérerons toujours la formule stricte dont la formule à priorité est l'abréviation.

Exemple 1.1.17 Nous donnons plusieurs exemples d'abréviation de formule par une formule à priorité :

- $a \wedge b \wedge c$ est l'abréviation de

$$((a \wedge b) \wedge c).$$

- $a \wedge b \vee c$ est l'abréviation de

$$((a \wedge b) \vee c).$$

2. C'est-à-dire $a \Rightarrow b \Rightarrow c$ est l'abréviation de $(a \Rightarrow (b \Rightarrow c))$.

— $a \vee b \wedge c$ est l'abréviation de

$$(a \vee (b \wedge c)).$$

Maintenant que la syntaxe est définie, nous définissons le sens des formules.

1.2 Sens des formules

Nous cherchons à déterminer si une formule est vraie ou fausse indépendamment des valeurs affectées à ses variables. Nous définissons d'abord le sens des connecteurs logiques. Ensuite nous expliquons comment calculer la valeur d'une formule et montrons le théorème de compacité. Nous terminons cette section par la présentation de définitions de notions de base de la logique qui constituent le langage commun des logiciens.

1.2.1 Sens des connecteurs

Nous désignons les valeurs de vérité par 0 pour faux et par 1 pour vrai. La constante \top vaut 1 et la constante \perp vaut 0, ce qui nous conduit, le plus souvent, à confondre les constantes et leurs valeurs, et à utiliser indifféremment \top , 1 et vrai, respectivement \perp , 0 et faux. Le sens des connecteurs logiques est donné par la table 1.1 qui indique les valeurs des formules de la première ligne suivant les valeurs *assignées* aux variables x et y .

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

TABLE 1.1 – Table de vérité des connecteurs.

1.2.2 Valeur d'une formule

Chacun sait évaluer une formule : nous associons à chaque variable de la formule une valeur dans l'ensemble $\mathbb{B} = \{0, 1\}$. La valeur de la formule est obtenue en remplaçant les variables par leurs valeurs et en effectuant les opérations suivant la table 1.1 page précédente. Néanmoins, pour raisonner sur les formules, nous définissons formellement la valeur d'une formule.

Définition 1.2.1 (Assignment) Une assignation est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble \mathbb{B} .

Définition 1.2.2 (Valeur d'une formule) Soient A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v . La valeur de $[A]_v$ est définie par récurrence sur l'ensemble des formules. Soient A, B des formules, x une variable et v une assignation.

- $[x]_v = v(x)$.
- $[\top]_v = 1$, $[\perp]_v = 0$.
- $[\neg A]_v = 1 - [A]_v$ autrement dit pour calculer la valeur de $\neg A$, nous soustrayons à 1 la valeur de A .
- $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$.
- $[(A \wedge B)]_v = \min\{[A]_v, [B]_v\}$.
- $[(A \Rightarrow B)]_v = 1$ si $[A]_v = 0$ sinon $[B]_v$.
- $[(A \Leftrightarrow B)]_v = 1$ si $[A]_v = [B]_v$ sinon 0.

D'après le théorème 1.1.13 page 12, toute formule (stricte) se décompose de façon unique en l'un des cas ci-dessus. Ainsi l'extension de v aux formules est une application des formules dans \mathbb{B} . En effet, soient 4 formules A, A', B, B' et deux opérations \circ et \circ' telles que $(A \circ B) = (A' \circ' B')$. Par unicité de la décomposition, $A = A', B = B', \circ = \circ'$, donc la valeur de la formule $(A \circ B)$ est définie uniquement par une et une seule des lignes de la définition de la valeur. Il est clair que la valeur d'une formule ne dépend que de ses variables et de sa structure, aussi l'évaluation d'une formule est présentée sous la forme d'une *table de vérité*.

Définition 1.2.3 (Table de vérité d'une formule) Une table de vérité d'une formule A est un tableau qui représente la valeur de A pour toutes les valeurs possibles des variables de A .

Chaque ligne de la table de vérité définit une assignation pour les variables des formules présentes dans les colonnes de la table et chaque colonne donne la valeur d'une formule.

Exemple 1.2.4 Nous donnons la table de vérité des formules suivantes : $x \Rightarrow y$, $\neg x$, $\neg x \vee y$, $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ et $x \vee \neg y$.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

1.2.3 Définitions et notions élémentaires de logique

Nous listons les notions élémentaires de la logique. Nous illustrons par des exemples et contre-exemples chacune des définitions introduites afin d'en faire comprendre les subtilités.

Définition 1.2.5 (Formules équivalentes) Deux formules A et B sont équivalentes si elles ont la même valeur pour toute assignation.

Exemple 1.2.6 Les colonnes des deux formules $x \Rightarrow y$ et $\neg x \vee y$ sont identiques dans l'exemple 1.2.4. Ces deux formules sont donc équivalentes. Par contre les formules $x \Rightarrow y$ et $x \vee \neg y$ ne sont pas équivalentes car elles n'ont pas les mêmes tables de vérité.

Remarque 1.2.7 Nous n'utilisons pas le symbole du connecteur logique \Leftrightarrow pour dire que A et B sont équivalentes. Nous notons que les formules A et B sont équivalentes par $A \equiv B$ ou simplement $A = B$ si le contexte nous permet de comprendre que le signe égal indique l'équivalence. Ainsi, $x \Rightarrow y = \neg x \vee y$ signifie que la formule $x \Rightarrow y$ est équivalente à la formule $\neg x \vee y$.

Définition 1.2.8 (Valide, tautologie) Une formule est valide si elle a la valeur 1 pour toute assignation. Une formule valide est aussi appelée une tautologie.

Exemple 1.2.9 En regardant la table de vérité de l'exemple 1.2.4 page précédente nous obtenons que :

- la formule $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide ;
- la formule $x \Rightarrow y$ n'est pas valide car

elle est fausse pour l'assignation $x = 1$ et $y = 0$, ce n'est donc pas une tautologie.

Notation : $\models A$ dénote le fait que la formule A est valide. Nous pouvons écrire $\models x \vee \neg x$, car

$x \vee \neg x$ est une tautologie.

Propriété 1.2.10 Les formules A et B sont équivalentes si et seulement si la formule $A \Leftrightarrow B$ est valide.

Preuve : La propriété est une conséquence de la table 1.1 page 13 et des définitions précédentes.

- \Rightarrow Si les formules A et B sont équivalentes cela signifie qu'elles ont la même table de vérité, ainsi d'après la définition du connecteur \Leftrightarrow donnée dans la table 1.1 page 13 la table de vérité de $A \Leftrightarrow B$ contient uniquement des 1 donc $A \Leftrightarrow B$ est valide.

\Leftarrow Si la formule $A \Leftrightarrow B$ est valide, nous déduisons que la table de vérité de $A \Leftrightarrow B$ contient uniquement des 1, ainsi d'après la définition du connecteur \Leftrightarrow donnée dans la table 1.1 page 13 les tables de vérité de A et de B coïncident donc les formules A et B sont équivalentes.

□

Définition 1.2.11 (Modèle d'une formule) Une assignation v qui donne la valeur 1 à une formule est un modèle de la formule. Nous dirons aussi que v satisfait la formule ou que v rend vraie la formule.

Exemple 1.2.12 $x \Rightarrow y$ a pour modèle

l'assignation $x = 1, y = 1$ ou l'assignation $x = 0$ et y quelconque.

Par contre l'assignation $x = 1$ et $y = 0$ n'est pas un modèle de $x \Rightarrow y$.

Cette notion de modèle s'étend aux ensembles de formules comme suit.

Définition 1.2.13 (Modèle d'un ensemble de formules) Une assignation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de chaque formule de l'ensemble.

Exemple 1.2.14

L'assignation $a = 0, b = 0$ (et c quelconque) est modèle de $\{a \Rightarrow b, b \Rightarrow c\}$.

Propriété 1.2.15 Une assignation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de la conjonction des formules de l'ensemble.

Preuve : La preuve est demandée dans l'exercice 11 page 37.

□

Exemple 1.2.16 L'ensemble de formules $\{a \Rightarrow b, b \Rightarrow c\}$ et la formule $(a \Rightarrow b) \wedge (b \Rightarrow c)$ ont les mêmes modèles.

Définition 1.2.17 (Contre-Modèle) Une assignation v qui donne la valeur 0 à une formule est un contre-modèle de la formule. Nous dirons que v ne satisfait pas la formule ou v rend la formule fausse.

Exemple 1.2.18 $x \Rightarrow y$ a pour contre-modèle

l'assignation $x = 1, y = 0$.

Remarque 1.2.19 (Contre-modèle d'un ensemble de formules) La notion de contre-modèle s'étend aux ensembles de formules de la même manière que la notion de modèle.

Définition 1.2.20 (Formule satisfaisable) Une formule (respectivement un ensemble de formules) est satisfaisable s'il existe une assignation qui en est un modèle.

Définition 1.2.21 (Formule insatisfaisable) Une formule (respectivement un ensemble de formules) est insatisfaisable si elle (respectivement s'il) n'est pas satisfaisable.

Une formule (respectivement un ensemble de formules) insatisfaisable ne possède pas de modèle. Sa table de vérité ne comporte que des 0. La négation d'une tautologie est donc une formule insatisfaisable.

Exemple 1.2.22

$x \wedge \neg x$ est insatisfaisable, mais $x \Rightarrow y$ ne l'est pas.

Remarque 1.2.23 Les logiciens utilisent le mot consistant comme synonyme de satisfaisable et contradictoire comme synonyme d'insatisfaisable.

Définition 1.2.24 (Conséquence) Soient Γ un ensemble de formules et A une formule : A est conséquence de l'ensemble Γ d'hypothèses si tout modèle de Γ est modèle de A . Le fait que A soit conséquence de Γ est noté par $\Gamma \models A$.

Exemple 1.2.25 D'après la table de vérité suivante, la formule $a \Rightarrow c$ est conséquence des hypothèses $a \Rightarrow b$ et $b \Rightarrow c$.

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Remarque 1.2.26 (Validité et conséquence) Nous notons A est valide par $\models A$, car A est valide si et seulement si A est conséquence de l'ensemble vide.

Maintenant, nous établissons l'équivalence de la validité d'une formule composée d'hypothèses et d'une conclusion avec la conséquence de la conclusion à partir des hypothèses mais aussi avec l'insatisfaisabilité des hypothèses et de la négation de la conclusion. Ces relations sont constamment utilisées dans les exercices et les démonstrations.

Propriété 1.2.27 Soient $n + 1$ formules A_1, \dots, A_n, B . Soit H_n la conjonction des formules A_1, \dots, A_n . Les trois formulations suivantes sont équivalentes :

1. $A_1, \dots, A_n \models B$, c'est-à-dire B est conséquence des hypothèses A_1, \dots, A_n .
2. La formule $H_n \Rightarrow B$ est valide.
3. $H_n \wedge \neg B$ est insatisfaisable.

Preuve : La propriété est une conséquence de la table 1.1 page 13 et des définitions précédentes. Soient $n + 1$ formules A_1, \dots, A_n, B et $H_n = A_1 \wedge \dots \wedge A_n$.

1 \Rightarrow 2 : Supposons que $A_1, \dots, A_n \models B$, c'est-à-dire tout modèle de A_1, \dots, A_n est aussi modèle de B .

- Considérons une assignation v qui n'est pas modèle de A_1, \dots, A_n cela signifie qu'il existe un i tel que $[A_i]_v = 0$, par conséquence $[H_n]_v = 0$ (d'après la table 1.1 page 13), donc $[H_n \Rightarrow B]_v = 1$.
- Considérons une assignation v modèle de A_1, \dots, A_n cela signifie que $[A_i]_v = 1$ pour $i = 1, \dots, n$, par conséquence $[H_n]_v = 1$ (d'après la table 1.1 page 13). Par hypothèse v est un modèle de B donc $[B]_v = 1$, nous en déduisons $[H_n \Rightarrow B]_v = 1$.

Nous obtenons donc dans tous les cas que $H_n \Rightarrow B$ est valide.

2 \Rightarrow 3 : Supposons que $H_n \Rightarrow B$ est valide, cela signifie que pour toute assignation v , $[H_n \Rightarrow B]_v = 1$. D'après la table 1.1 page 13 nous avons deux possibilités soit $[H_n]_v = 0$, soit $[H_n]_v = 1$ et $[B]_v = 1$. Or $[H_n \wedge \neg B]_v = \min([H_n]_v, [\neg B]_v) = \min([H_n]_v, 1 - [B]_v)$. Dans les deux cas, nous obtenons $[H_n \wedge \neg B]_v = 0$. Nous concluons donc que $H_n \wedge \neg B$ est insatisfaisable.

3 \Rightarrow 1 : Supposons que $H_n \wedge \neg B$ est insatisfaisable, c'est-à-dire que pour toute assignation la formule $H_n \wedge \neg B$ est contradictoire. Montrons que les modèles de A_1, \dots, A_n sont aussi des modèles de B .

Soit v une assignation modèle de A_1, \dots, A_n , il en découle que $[A_i]_v = 1$ pour $i = 1, \dots, n$ donc $[H_n]_v = [A_1 \wedge \dots \wedge A_n]_v = 1$. D'après notre hypothèse, nous en déduisons que $[\neg B]_v = 0$. D'où, $1 - [B]_v = 0$. Nous avons donc $[B]_v = 1$, c'est-à-dire que v est aussi un modèle de B .

En utilisant le résultat de l'exercice 7 page 36 nous concluons. □

Exemple 1.2.28 Nous considérons les deux formules $a \Rightarrow b$ et $b \Rightarrow c$, nous illustrons le théorème précédent en prouvant que $a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$, soit en montrant que $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$ est une tautologie, soit que $(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge \neg(a \Rightarrow c)$ est insatisfaisable. Pour cela nous donnons la table de vérité associée à ces formules.

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge \neg(a \Rightarrow c)$
0	0	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	0

1.2.4 Compacité

Nous énonçons et prouvons un résultat central de la logique « un ensemble de formules a un modèle si et seulement si tous ses sous-ensembles finis ont un modèle ». La preuve ci-dessous de ce résultat est écrite pour la logique propositionnelle dans le cas où l'ensemble des variables est dénombrable. Cette restriction nous permet de faire une preuve par récurrence sur les entiers, en évitant l'usage de propriétés de la théorie des ensembles ou issues de la topologie. Le théorème 1.2.30 est utilisé dans la preuve du théorème de Herbrand (théorème 5.1.17 page 111). Aussi, nous remettons l'étude du paragraphe présent à ce moment.

Définition 1.2.29 (Fonction et prolongement) Une fonction f est un triplet $f = (G, X, Y)$ où G, X, Y sont des ensembles vérifiant les propriétés suivantes :

- $G \subset X \times Y$.
- Pour tout $x \in X$, il existe un et un seul $y \in Y$ tel que $(x, y) \in G$.

L'ensemble X est l'ensemble de départ ou source de f , et l'ensemble Y est l'ensemble d'arrivée ou but de f et G est le graphe de f . Soient deux fonctions $f = (G, X, Y)$ et $f' = (G', X', Y')$, la fonction f est un prolongement de f' , ou une extension de f' si nous avons les relations : $X' \subset X, Y' \subset Y$ et pour tout $x \in X'$, $f(x) = f'(x)$.

Théorème 1.2.30 (Compacité propositionnelle) Un ensemble de formules propositionnelles a un modèle si et seulement si tous ses sous-ensembles finis ont un modèle.

Preuve : Soit E un ensemble de formules propositionnelles. Soit $p_{i(i \in \mathbb{N})}$ la liste des variables propositionnelles de E . La condition nécessaire est triviale : si E a un modèle, celui-ci est modèle de tous les sous-ensembles de E , en particulier des sous-ensembles finis. Établissons la condition suffisante. Supposons que tout sous-ensemble fini de E a un modèle. Nous définissons une fonction δ puis nous prouvons que δ est un modèle de E .

1. Définition de δ : soit $\delta_{i(i \in \mathbb{N})}$ la suite de fonctions suivantes :
 - δ_0 est la fonction vide, autrement dit $\delta_0 = (\emptyset, \emptyset, \{0, 1\})$.
 - Le domaine de δ_i est $\{p_j \mid 0 \leq j < i\}$.
 - Soient η_i et ζ_i les fonctions prolongeant δ_i , de domaine $\{p_j \mid 0 \leq j \leq i\}$ telles que $\eta_i(p_i) = 0$ et $\zeta_i(p_i) = 1$. Si tout sous-ensemble fini de E a un modèle qui est une extension de η_i alors $\delta_{i+1} = \eta_i$ sinon $\delta_{i+1} = \zeta_i$.
 Soit δ la fonction de domaine $\{p_i \mid i \in \mathbb{N}\}$ ainsi définie : pour tout $i \in \mathbb{N}$, $\delta(p_i) = \delta_{i+1}(p_i)$.
2. Propriété des fonctions δ_i : Soit $P(i)$ la propriété telle que tout sous-ensemble fini de E a un modèle qui est une extension de δ_i . Par récurrence, nous prouvons que cette propriété est vraie pour tout entier.
 - (a) la propriété $P(0)$ est vraie, car comme $\delta_0 = (\emptyset, \emptyset, \{0, 1\})$ nous retrouvons notre hypothèse initiale : tout sous-ensemble fini de E a un modèle.
 - (b) Supposons que $P(i)$ est vraie. Montrons $P(i+1)$. Nous examinons deux cas suivant la définition de δ_{i+1} .
 - $\delta_{i+1} = \eta_i$. Par définition de δ_{i+1} , tout sous-ensemble fini de E a un modèle qui est une extension de δ_{i+1} , donc $P(i+1)$ est vérifiée.
 - $\delta_{i+1} = \zeta_i$. Supposons que $P(i+1)$ ne soit pas vérifiée. Alors, par définition de δ_{i+1} , il existe un sous-ensemble fini F de E , qui n'a pas de modèle qui est une extension de η_i et, puisque $P(i+1)$ n'est pas vraie, il existe un sous-ensemble fini G de E , qui n'a pas de modèle qui soit une extension de ζ_i . Par suite $F \cup G$ est un sous-ensemble fini de E , qui n'a pas de modèle qui soit une extension de δ_i , ce qui contredit l'hypothèse $P(i)$. Par suite $P(i+1)$ est vérifiée.
 Ainsi $P(0)$ est vraie et pour tout i , $P(i)$ implique $P(i+1)$. Donc, par récurrence, la propriété P est vraie pour tout entier naturel.
3. Prouvons que δ est modèle de E : Soit A une formule de E . Soit k le plus grand indice d'une variable propositionnelle apparaissant dans la formule A . D'après la propriété P , le singleton $\{A\}$ a un modèle η qui est une extension de δ_{k+1} . Puisque δ et η donnent la même valeur aux variables de A , δ est modèle de A , formule quelconque de E . Donc δ est modèle de E .

□

1.2.5 Équivalences remarquables

Raisonnement par équivalence c'est utiliser les propriétés de l'équivalence (réflexivité, symétrie, transitivité), et la propriété de remplacement d'une formule par une autre formule équivalente pour obtenir de nouvelles équivalences à partir des équivalences déjà prouvées ou admises. Ci-dessous, nous listons des équivalences remarquables de la logique.

1. La disjonction est :
 - Associative, c'est-à-dire, $x \vee (y \vee z) \equiv (x \vee y) \vee z$.
 - Commutative, c'est-à-dire, $x \vee y \equiv y \vee x$.
 - 0 est l'élément neutre de la disjonction, c'est-à-dire, $0 \vee x \equiv x$.
2. La conjonction est :
 - Associative, c'est-à-dire, $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$.
 - Commutative, c'est-à-dire, $x \wedge y \equiv y \wedge x$.
 - 1 est l'élément neutre de la conjonction, c'est-à-dire, $1 \wedge x \equiv x$.
3. La conjonction est distributive sur la disjonction : $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$.
4. La disjonction est distributive sur la conjonction : $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$.
5. Les lois de la négation :
 - $x \wedge \neg x \equiv 0$.
 - $x \vee \neg x \equiv 1$ (Le tiers-exclus).
6. $\neg \neg x \equiv x$.
7. $\neg 0 \equiv 1$.
8. $\neg 1 \equiv 0$.
9. Les lois de De Morgan :
 - $\neg(x \wedge y) \equiv \neg x \vee \neg y$.
 - $\neg(x \vee y) \equiv \neg x \wedge \neg y$.
10. 0 est l'élément absorbant de la conjonction : $0 \wedge x \equiv 0$.

11. 1 est l'élément absorbant de la disjonction : $1 \vee x \equiv 1$.

12. idempotence de la disjonction : $x \vee x \equiv x$.

13. idempotence de la conjonction : $x \wedge x \equiv x$.

Les équivalences 1 à 5 se démontrent à l'aide de tables de vérité³. Les équivalences 6 à 13 se déduisent des équivalences 1 à 5, comme nous le montrons dans la section 1.5 page 25. Ci-dessous, nous donnons quelques lois de simplification qui permettent d'alléger les raisonnements logiques.

Propriété 1.2.31 (Lois de simplification) *Pour tout x, y nous avons :*

- $x \vee (x \wedge y) \equiv x$.
- $x \wedge (x \vee y) \equiv x$.
- $x \vee (\neg x \wedge y) \equiv x \vee y$.
- $x \wedge (\neg x \vee y) \equiv x \wedge y$.

Preuve : En utilisant les équivalences remarquables nous prouvons les trois lois de simplification. La preuve est demandée dans l'exercice 12 page 37. \square

1.3 Substitution et remplacement

Dans ce paragraphe, nous étendons l'ensemble des formules valides par substitution et remplacement.

1.3.1 Substitution

Définition 1.3.1 (Substitution) *Une substitution est une application de l'ensemble des variables dans l'ensemble des formules. L'application d'une substitution σ à une formule consiste à remplacer dans la formule toute variable x par la formule $\sigma(x)$.*

Notation : Soit A une formule, nous notons $A\sigma$ ou $\sigma(A)$ l'application de la substitution σ à la formule A .

Définition 1.3.2 (Support d'une substitution, substitution à support fini) *Le support d'une substitution σ est l'ensemble des variables x telles que $x\sigma \neq x$. Une substitution σ à support fini est notée $\langle x_1 := A_1, \dots, x_n := A_n \rangle$, où A_1, \dots, A_n sont des formules, x_1, \dots, x_n sont des variables distinctes et la substitution vérifie :*

- pour i de 1 à n , $x_i\sigma = A_i$
- pour toute variable y telle que $y \notin \{x_1, \dots, x_n\}$, nous avons : $y\sigma = y$

Exemple 1.3.3 Soient la formule $A = x \vee x \wedge y \Rightarrow z \wedge y$ et la substitution $\sigma = \langle x := a \vee b, z := b \wedge c \rangle$, σ appliquée à A donne :

$$A\sigma = (a \vee b) \vee (a \vee b) \wedge y \Rightarrow (b \wedge c) \wedge y.$$

Le support de σ est fini et comporte les variables x et z .

Propriété 1.3.4 Soient A une formule, v une assignation et σ une substitution, nous avons $[A\sigma]_v = [A]_w$ où pour toute variable x , $w(x) = [\sigma(x)]_v$.

Preuve :

3. Nous laissons le soin au lecteur de se convaincre de la véracité de ces équivalences en construisant les tables correspondantes.

Soient A une formule, v une assignation et σ une substitution, nous voulons montrer que $[A\sigma]_v = [A]_w$ où pour toute variable x , $w(x) = [\sigma(x)]_v$. Par une récurrence sur la taille des formules, nous obtenons :

Cas de base : $|A| = 0$.

- Soit $A = k$ une constante et donc $[k\sigma]_v = [k]_v = [k]_w$ (pour toute assignation v , $[\perp]_v = 0$ et $[\top]_v = 1$).
- Soit $A = x$ une variable et donc par construction $[x\sigma]_v = w(x) = [x]_w$.

Induction :

Supposons que la propriété soit vraie pour toute formule de taille inférieure ou égale à n , montrons qu'elle est vraie pour une formule A de taille $n + 1$:

- Soit $A = \neg B$ nous avons donc $[A\sigma]_v = [\neg B\sigma]_v = 1 - [B\sigma]_v$ et $[A]_w = [\neg B]_w = 1 - [B]_w$. Comme $|B| = n$ nous obtenons donc par hypothèse d'induction $[B\sigma]_v = [B]_w$ où pour toute variable x , $w(x) = [\sigma(x)]_v$, ce qui implique que $[A\sigma]_v = [A]_w$.
- Soit $A = (B \circ C)$, nous avons donc $[A\sigma]_v = [(B\sigma \circ C\sigma)]_v = f_\circ([B\sigma]_v, [C\sigma]_v)$ et $[A]_w = [(B \circ C)]_w = f_\circ([B]_w, [C]_w)$, où f_\circ est la fonction associée à \circ correspondant à la définition 1.2.1 page 14. Comme $|B| < n + 1$ et $|C| < n + 1$ nous obtenons donc par hypothèse d'induction $[B\sigma]_v = [B]_w$ et $[C\sigma]_v = [C]_w$ où pour toute variable x , $w(x) = [\sigma(x)]_v$, ce qui implique que $[A\sigma]_v = [A]_w$.

□

Exemple 1.3.5 Soit $A = x \vee y \vee d$ une formule. Soit $\sigma = \langle x := a \vee b, y := b \wedge c \rangle$ une substitution. Soit v une assignation telle que $v(a) = 1$, $v(b) = 0$, $v(c) = 0$, $v(d) = 0$. Nous avons :

$A\sigma = (a \vee b) \vee (b \wedge c) \vee d$ et $[A\sigma]_v = (1 \vee 0) \vee (0 \wedge 0) \vee 0 = 1$ et $[A]_w = 1 \vee 0 \vee 0 = 1$ où l'assignation w est définie comme suit :

- $w(x) = [\sigma(x)]_v = [a \vee b]_v = 1 \vee 0 = 1$.
- $w(y) = [\sigma(y)]_v = [b \wedge c]_v = 0 \wedge 0 = 0$.
- $w(d) = [\sigma(d)]_v = [d]_v = 0$.

Théorème 1.3.6 L'application d'une substitution à une formule valide donne une formule valide.

Preuve : Soient A une formule valide, σ une substitution et v une assignation quelconque. D'après la propriété 1.3.4 page ci-contre : $[A\sigma]_v = [A]_w$ où pour toute variable x , $w(x) = [\sigma(x)]_v$. Puisque A est valide, $[A]_w = 1$. Par suite $A\sigma$ vaut 1 dans toute assignation, c'est donc une formule valide. □

Exemple 1.3.7 Soit A la formule $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$. Cette formule est valide, c'est une équivalence remarquable. Soit σ la substitution suivante : $\langle p := (a \vee b), q := (c \wedge d) \rangle$. La formule

$A\sigma = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$ est donc valide.

La substitution est définie sur les formules, pour l'appliquer sans erreur aux formules à priorité, il suffit que l'image, par la substitution, de toute variable soit une variable, une constante ou une formule entre parenthèses.

1.3.2 Remplacement

La notion de substitution ne nous permet pas de remplacer une formule par une formule, nous introduisons la notion de remplacement à cette fin.

Définition 1.3.8 (Remplacement) Soient A, B, C, D des formules. La formule D est obtenue en remplaçant dans C certaines occurrences de A par B , s'il existe une formule E et une variable x telles que, $C = E < x := A >$ et $D = E < x := B >$.

Exemple 1.3.9 Considérons la formule $C = ((a \Rightarrow b) \vee \neg(a \Rightarrow b))$.

— La formule obtenue en remplaçant toutes les occurrences de $(a \Rightarrow b)$ par $(a \wedge b)$ dans C est

$$D = ((a \wedge b) \vee \neg(a \wedge b)),$$

elle est obtenue en considérant la formule $E = (x \vee \neg x)$ et les substitutions $< x := (a \wedge b) >$ et $< x := (a \Rightarrow b) >$.

— La formule obtenue en remplaçant la première occurrence de $(a \Rightarrow b)$ par $(a \wedge b)$ dans C est

$$D = ((a \wedge b) \vee \neg(a \Rightarrow b)),$$

elle est obtenue en considérant la formule $E = (x \vee \neg(a \Rightarrow b))$ et les substitutions $< x := (a \wedge b) >$ et $< x := (a \Rightarrow b) >$.

La différence entre une substitution et un remplacement est qu'une substitution remplace un ensemble de variables par des formules alors qu'un remplacement remplace les occurrences de certaines formules par une autre formule en utilisant des substitutions.

Théorème 1.3.10 Soient C une formule et D la formule obtenue en remplaçant, dans C , des occurrences de la formule A par la formule B , alors $(A \Leftrightarrow B) \models (C \Leftrightarrow D)$.

Preuve : Par définition du remplacement, il existe une formule E et une variable x telles que, $C = E < x := A >$ et $D = E < x := B >$. Supposons que v est une assignation modèle de $(A \Leftrightarrow B)$. Nous avons donc $[A]_v = [B]_v$. D'après la propriété 1.3.4 page 20 :

— $[C]_v = [E]_w$ où w est identique à v sauf que $w(x) = [A]_v$

— $[D]_v = [E]_{w'}$ où w' est identique à v sauf que $w'(x) = [B]_v$

Puisque $[A]_v = [B]_v$, les assignations w et w' sont identiques, donc $[C]_v = [D]_v$. Par suite v est modèle de $(C \Leftrightarrow D)$. \square

Corollaire 1.3.11 Soient C une formule et D la formule obtenue en remplaçant, dans C , une occurrence de la formule A par la formule B , alors $A \equiv B$ implique $C \equiv D$.

Preuve : Si $A \equiv B$, alors la formule $(A \Leftrightarrow B)$ est valide (propriété 1.2.10), donc la formule $(C \Leftrightarrow D)$ également puisqu'elle est, d'après le théorème ci-dessus, la conséquence de $(A \Leftrightarrow B)$, par suite $C \equiv D$. \square

Exemple 1.3.12 Le remplacement d'une occurrence d'une formule A par une occurrence de B est mis en évidence par des boîtes marquant ces occurrences.

— D'après théorème 1.3.10 : $p \Leftrightarrow q \models (p \vee (\boxed{p} \Rightarrow r)) \Leftrightarrow (p \vee (\boxed{q} \Rightarrow r))$.

— D'après le corollaire 1.3.11 : $(\neg(p \vee q) \Rightarrow (\neg(p \vee q) \vee r)) \equiv (\neg(p \vee q) \Rightarrow ((\neg p \wedge \neg q) \vee r))$, puisque $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$.

Remarque 1.3.13 Le théorème précédent et son corollaire s'appliquent aux formules. Quand nous faisons un remplacement directement sur une formule à priorité, nous devons nous assurer que ce remplacement reste correct sur les formules (strictes) sous peine de commettre des erreurs. Par exemple, considérons les deux équivalences $a \wedge \boxed{b} \equiv a \wedge \boxed{b}$ et $\neg c \Rightarrow d \equiv c \vee d$. Remplaçons b à gauche par $\neg c \Rightarrow d$ et à droite par $c \vee d$. Nous observons que bien que $\neg c \Rightarrow d \equiv c \vee d$,

$a \wedge \neg c \Rightarrow d \not\equiv a \wedge c \vee d$, car pour $a = c = d = 0$, la formule de gauche vaut 1 et celle de droite vaut 0. Ici le corollaire ne doit pas être appliqué à l'occurrence encadrée car $a \wedge \neg c \Rightarrow d$ est une abréviation de $((a \wedge \neg c) \Rightarrow d)$, donc $\neg c \Rightarrow d$ n'apparaît pas comme une occurrence possible de $a \wedge \neg c \Rightarrow d$.

Nous avons défini les remplacements à partir des substitutions, nous allons maintenant appliquer les remplacements à une formule afin de la transformer en une formule en forme normale équivalente.

1.4 Formes Normales

Mettre une formule en forme normale consiste à la transformer en une formule équivalente ayant des propriétés structurelles. Nous introduisons deux notions de formes normales : la forme normale disjonctive qui permet de mettre en évidence les modèles et la forme normale conjonctive qui exhibe les contre-modèles. La définition de forme normale nécessite l'introduction des concepts de littéral, monôme et clause.

Définition 1.4.1 (Littéral, monôme, clause)

- Un littéral est une variable ou la négation d'une variable.
- Un monôme est une conjonction de littéraux.
- Une clause est une disjonction de littéraux.

Exemple 1.4.2

Nous illustrons par quelques exemples simples ces nouvelles notions :

- $x, y, \neg z$ sont des littéraux.
- $x \wedge \neg y \wedge z$ est un monôme dont l'unique modèle est $x = 1, y = 0, z = 1$.
- Le monôme $x \wedge \neg y \wedge z \wedge \neg x$ comporte une variable et sa négation : il vaut 0.
- $x \vee \neg y \vee z$ est une clause dont l'unique contre modèle est $x = 0, y = 1, z = 0$.
- La clause $x \vee \neg y \vee z \vee \neg x$ comporte une variable et sa négation : elle vaut 1.

1.4.1 Transformation en forme normale

Nous introduisons la notion de forme normale et montrons qu'il est toujours possible de transformer une formule en une formule équivalente en forme normale.

Définition 1.4.3 (Forme normale) Une formule est en forme normale si elle n'utilise que les opérateurs \wedge, \vee, \neg et que les négations sont uniquement appliquées aux variables.

Exemple 1.4.4 La formule $\neg a \vee b$ est en forme normale, alors que la formule $a \Rightarrow b$ n'est pas en forme normale bien qu'elle soit équivalente à la première.

Nous expliquons maintenant comment transformer toute formule en une formule en forme normale équivalente grâce à des remplacements. Appliquer l'équivalence $A \equiv B$ à la formule C , c'est remplacer dans C une occurrence de A par une occurrence de B : nous avons prouvé dans le théorème 1.3.10 page ci-contre qu'un tel remplacement change C en une formule équivalente. Ainsi pour transformer une formule en une formule équivalente de forme normale, nous appliquons les transformations suivantes :

1. **Élimination des équivalences** : remplacer une occurrence de $A \Leftrightarrow B$ par l'une des sous-formules :
 - (a) $(\neg A \vee B) \wedge (\neg B \vee A)$.
 - (b) $(A \wedge B) \vee (\neg A \wedge \neg B)$.
2. **Élimination des implications** : remplacer une occurrence de $A \Rightarrow B$ par $\neg A \vee B$.
3. **Déplacement des négations** : remplacer une occurrence de
 - (a) $\neg\neg A$ par A .
 - (b) $\neg(A \vee B)$ par $\neg A \wedge \neg B$.

(c) $\neg(A \wedge B)$ par $\neg A \vee \neg B$.

ainsi les négations ne portent que sur des variables.

En appliquant ces trois transformations dans l'ordre indiqué, il est clair que la formule initiale a été transformée en une formule en forme normale équivalente. Dans l'exercice 25 page 38, nous prouvons que l'ordre des transformations n'est pas important : en effectuant les transformations ci-dessus dans un ordre quelconque, nous obtenons finalement une formule équivalente en forme normale.

Remarque 1.4.5 Il est par exemple recommandé de remplacer une sous-formule de la forme $\neg(A \Rightarrow B)$ par $A \wedge \neg B$, ce qui en fait combine une élimination de l'implication et un déplacement de la négation. En pratique, il est plus efficace de simplifier le plus tôt possible de la façon suivante :

- Remplacer par 0 une conjonction qui comporte soit une formule et sa négation, soit un 0.
- Remplacer par 1 une disjonction qui comporte soit une formule et sa négation, soit un 1.
- Remplacer $\neg 1$ par 0 et $\neg 0$ par 1.
- Enlever les 0 des disjonctions et les 1 des conjonctions.
- Appliquer les simplifications $x \vee (x \wedge y) \equiv x$, $x \wedge (x \vee y) \equiv x$, $x \vee (\neg x \wedge y) \equiv x \vee y$.
- Appliquer l'idempotence de la disjonction et de la conjonction.

1.4.2 Transformation en forme normale disjonctive (somme de monômes)

La forme normale disjonctive permet de trouver facilement des modèles.

Définition 1.4.6 (Forme normale disjonctive) Une formule est une forme normale disjonctive (en bref *fnd*) si et seulement si elle est une disjonction (somme) de monômes.

L'intérêt des formes normales disjonctives est de mettre en évidence les modèles.

Exemple 1.4.7 $(x \wedge y) \vee (\neg x \wedge \neg y \wedge z)$ est une *fnd* composée de deux monômes, chacun d'eux nous donne un des deux modèles possibles :

$$x = 1, y = 1 \text{ et } x = 0, y = 0, z = 1.$$

En partant d'une forme normale et en distribuant toutes les conjonctions sur les disjonctions, nous obtenons une disjonction de monômes. Pour cela il faut aussi savoir regrouper plusieurs applications de la distributivité.

Exemple 1.4.8 Dans une transformation en disjonction de monômes, nous pouvons appliquer de gauche à droite l'équivalence $(a \vee b) \wedge (c \vee d \vee e) \equiv$

$$(a \wedge c) \vee (a \wedge d) \vee (a \wedge e) \vee (b \wedge c) \vee (b \wedge d) \vee (b \wedge e).$$

La transformation par équivalence d'une formule en une disjonction de monômes peut être utilisée pour déterminer si une formule est valide ou non. Soit A une formule dont nous souhaitons déterminer la validité : Nous transformons $\neg A$ en une disjonction de monômes équivalente à $\neg A$. Si $\neg A = 0$ alors $A = 1$ donc A est valide, sinon, toutes simplifications étant faites, $\neg A$ est égal à une disjonction de monômes non nuls, qui nous donnent des modèles de $\neg A$, donc des contre-modèles de A .

Exemple 1.4.9 Soit $A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \wedge q \Rightarrow r)$. Déterminons si A est valide par transformation de $\neg A$ en disjonction de monômes.

$$\neg A$$

$$\equiv (p \Rightarrow (q \Rightarrow r)) \wedge \neg(p \wedge q \Rightarrow r)$$

par l'identité $\neg(B \Rightarrow C) \equiv B \wedge \neg C$

$$\equiv (\neg p \vee \neg q \vee r) \wedge \neg(p \wedge q \Rightarrow r)$$

en éliminant deux implications

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

par l'identité $\neg(B \Rightarrow C) \equiv B \wedge \neg C$

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

par distributivité de la conjonction sur la disjonction

$$\equiv 0$$

car chaque monôme vaut 0

Donc $\neg A = 0$ et $A = 1$, c'est-à-dire A est valide.

Exemple 1.4.10 Soit $A = ((a \Rightarrow b) \wedge c) \vee (a \wedge d)$. Déterminer si A est valide par transformation de $\neg A$ en disjonction de monômes.

$$\neg A$$

$$\equiv \neg((a \Rightarrow b) \wedge c) \wedge \neg(a \wedge d)$$

par déplacement des négations

$$\equiv (\neg(a \Rightarrow b) \vee \neg c) \wedge (\neg a \vee \neg d)$$

par déplacement des négations

$$\equiv (a \wedge \neg b \vee \neg c) \wedge (\neg a \vee \neg d)$$

par combinaison du déplacement d'une négation

et élimination de l'implication

$$\equiv (a \wedge \neg b \wedge \neg a) \vee (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d)$$

par distributivité de la disjonction sur la conjonction

$$\equiv (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d)$$

par simplification

Nous obtenons 3 modèles de $\neg A$ ($a = 1, b = 0, d = 0$; $a = 0, c = 0$; $c = 0, d = 0$), c'est-à-dire, des contre-modèles de A . Donc A n'est pas valide.

1.4.3 Transformation en forme normale conjonctive (produit de clauses)

La forme normale conjonctive permet d'exhiber facilement des contre-modèles.

Définition 1.4.11 (Forme normale conjonctive) Une formule est une forme normale conjonctive (en bref fnc) si et seulement si elle est une conjonction (produit) de clauses.

L'intérêt des formes normales conjonctives est de mettre en évidence des contre-modèles.

Exemple 1.4.12 $(x \vee y) \wedge (\neg x \vee \neg y \vee z)$ est une fnc, qui a deux contre-modèles

$$x = 0, y = 0 \text{ et } x = 1, y = 1, z = 0.$$

Par convention, nous considérons que 0 et 1 sont des disjonctions de monômes et des conjonctions de clauses. Nous appliquons la distributivité (inhabituelle) de la disjonction sur la conjonction, autrement dit nous remplaçons toute sous-formule $A \vee (B \wedge C)$ par $(A \vee B) \wedge (A \vee C)$, et toute sous-formule $(B \wedge C) \vee A$ par $(B \vee A) \wedge (C \vee A)$.

Exemple 1.4.13 Nous reprenons l'exemple 1.4.8 page ci-contre et obtenons $(a \wedge b) \vee (c \wedge d \wedge e) \equiv$

$$(a \vee c) \wedge (a \vee d) \wedge (a \vee e) \wedge (b \vee c) \wedge (b \vee d) \wedge (b \vee e).$$

La transformation par équivalence d'une formule en une conjonction de clauses de littéraux peut aussi être utilisée pour déterminer si une formule est valide ou non. Soit A une formule dont nous souhaitons déterminer la validité : Nous transformons A en une conjonction de clauses *équivalente* à A . Si $A = 1$ alors A est valide, sinon, toutes simplifications étant faites, A est égale à une conjonction de clauses non égales à 1, et chacune de ces disjonctions nous donne un contre-modèle de A .

1.5 Algèbre de Boole

Cette notion fut introduite par le mathématicien britannique George Boole au milieu du XIX^e siècle. Elle permet notamment de traduire les propositions en équations (généralement, cette écriture est plus concise). Nous rappelons d'abord la définition d'une algèbre de Boole. Nous déduisons alors que la logique propositionnelle est une algèbre de Boole. Ensuite nous prouvons certaines propriétés usuelles des algèbres de Boole. Nous terminons cette section en présentant la notion de dualité.

1.5.1 Définition et notations

La définition d'algèbre de Boole donnée ci-dessous est minimale en nombre d'axiomes.

Définition 1.5.1 Une algèbre de Boole est un ensemble d'au moins deux éléments, 0, 1, et trois opérations, complément (le complément de x est noté \bar{x}), somme (+) et produit (.), qui vérifient les axiomes suivants :

1. la somme est :
 - associative : $x + (y + z) = (x + y) + z$,
 - commutative : $x + y = y + x$,
 - 0 est élément neutre de la somme : $0 + x = x$,
2. le produit est :
 - associatif : $x.(y.z) = (x.y).z$,
 - commutatif : $x.y = y.x$,
 - 1 est élément neutre du produit : $1.x = x$,
3. le produit est distributif sur la somme : $x.(y + z) = (x.y) + (x.z)$,
4. la somme est distributive sur le produit : $x + (y.z) = (x + y).(x + z)$,
5. les lois de la négation :
 - $x + \bar{x} = 1$,
 - $x.\bar{x} = 0$.

Attention, le fait que la distributivité de la somme sur le produit n'est pas usuelle rend son application propice aux erreurs.

Comme indiqué précédemment dans la sous-section 1.2.5 page 19, nous pouvons prouver en utilisant des tables de vérité que la logique propositionnelle vérifie l'ensemble des axiomes d'une algèbre de Boole. Ainsi nous pouvons considérer la logique propositionnelle comme la plus petite algèbre de Boole, car elle contient deux éléments. De ce fait, nous pouvons utiliser les notations booléennes (plus condensées) en lieu et place des notations propositionnelles, comme indiqué dans la table de correspondance donnée dans la figure 1.1. Nous conseillons de les utiliser pour effectuer de gros calculs. Par ailleurs, observez que ces notations sont fréquemment utilisées dans le domaine du matériel.

Remarque 1.5.2 (Conventions d'écriture)

- Parfois, la négation est utilisée à la place du complémentaire. Par exemple, $\neg a$ dénote \bar{a} et $\neg(a + b)$ représente $\overline{a + b}$.
- Afin de permettre l'omission du point, dans les notations booléennes, les variables sont souvent dénotées par une seule lettre (avec ou sans indice). Ainsi lorsque a et b sont des variables, des formules entre parenthèses ou sous une barre de négation, nous abrégons $a.b$ en ab suivant une pratique usuelle en mathématiques.
- Enfin l'équivalence de deux formules est systématiquement notée avec le symbole $=$ au lieu du symbole \equiv .

notations logiques	notations booléennes
$\neg a$	\bar{a}
$a \wedge b$	$a.b$
$a \vee b$	$a + b$
$a \Rightarrow b$	$\bar{a} + b$
$a \Leftrightarrow b$	$a.b + (\bar{a}.\bar{b})$ ou $(a + \bar{b}).(\bar{a} + b)$

FIGURE 1.1 – Correspondance entre l’algèbre de Boole et la logique propositionnelle.

Il est important de noter que la logique propositionnelle n’est pas l’unique algèbre de Boole. Par exemple, l’ensemble $\mathcal{P}(X)$ des sous-ensembles d’un ensemble X est une algèbre de Boole, comme nous l’indique la mise en correspondance des opérateurs ensemblistes avec leur désignation dans l’algèbre de Boole dans la figure 1.2 page suivante.

Algèbre de Boole	$\mathcal{P}(X)$
1	X
0	\emptyset
\bar{p}	$X - p$
$p + q$	$p \cup q$
$p.q$	$p \cap q$

FIGURE 1.2 – Correspondance entre l’algèbre de Boole et $\mathcal{P}(X)$.

Puisque les lois de simplification sont déduites des lois de l’algèbre de Boole, comme nous allons le montrer dans la suite, elles sont vérifiées dans toute algèbre de Boole. En particulier dans l’algèbre de Boole $\mathcal{P}(X)$ nous avons $A \cup (A \cap B) = A$. Une *preuve dans l’algèbre de Boole* est une suite d’égalités, permettant de passer d’une formule à une autre formule équivalente en utilisant les axiomes ou simplifications de l’algèbre de Boole.

1.5.2 Propriétés

Nous présentons certaines des propriétés qui découlent de la définition de l’algèbre de Boole et qui sont couramment utilisées dans les raisonnements. Pour raccourcir les preuves de ces propriétés, nous utilisons implicitement l’associativité et la commutativité de la somme et du produit.

Propriété 1.5.3 *Dans une algèbre de Boole, pour tout x , il y a un et un seul y tel que $x + y = 1$ et $xy = 0$, autrement dit la négation est unique.*

Preuve : Il y a un élément y vérifiant $x + y = 1$ et $xy = 0$, c’est \bar{x} d’après les axiomes de l’algèbre de Boole. Montrons qu’il est unique. Raisonnons par contradiction, supposons qu’il n’est pas unique, nous avons donc qu’il existe y et z deux éléments distincts tels que $x + y = 1$, $xy = 0$, $x + z = 1$, et $xz = 0$. Nous considérons $u = xy + z$. Puisque $xy = 0$, nous avons : $u = z$. Par distributivité de la somme sur le produit, nous avons : $u = (x + z)(y + z)$. Puisque $x + z = 1$ et que 1 est l’élément neutre du produit, nous avons : $u = y + z$. Nous concluons donc que $z = y + z$. En considérant $v = xz + y$ et échangeant les rôles de y et z , nous obtenons de façon analogue $y = xz + y$. Par suite $y = z$, ce qui prouve donc l’unicité de la négation. \square

Propriété 1.5.4 *Dans une algèbre de Boole, les équivalences suivantes sont vraies pour tout x et tout y :*

1. $\bar{\bar{1}} = 0$.
2. $\bar{\bar{0}} = 1$.
3. $\bar{\bar{x}} = x$.
4. Idempotence du produit : $x.x = x$.
5. Idempotence de la somme : $x + x = x$.
6. 1 est élément absorbant de la somme : $1 + x = 1$.

7. 0 est élément absorbant du produit : $0.x = 0$.

8. Lois de De Morgan :

$$- \overline{xy} = \bar{x} + \bar{y}.$$

$$- \overline{x+y} = \bar{x}.\bar{y}.$$

Preuve :

1. $\bar{1} = 0$.

Par définition de la négation, $x.\bar{x} = 0$. Donc, $1.\bar{1} = 0$. Puisque 1 est l'élément neutre du produit, nous avons $\bar{1} = 0$.

2. $\bar{0} = 1$.

Par définition de la négation, $x + \bar{x} = 1$. Donc, $0 + \bar{0} = 1$. Puisque 0 est l'élément neutre de la somme, nous avons $\bar{0} = 1$.

3. $\bar{\bar{x}} = x$.

D'après les propriétés de la négation et la commutativité, nous avons : $\bar{x} + x = 1$, $\bar{x}.x = 0$, $\bar{x} + \bar{\bar{x}} = 1$, et $\bar{x}.\bar{\bar{x}} = 0$. À cause de l'unicité de la négation prouvée ci-dessus (propriété 1.5.3 page précédente), nous en déduisons que $x = \bar{\bar{x}}$.

4. Idempotence du produit : $x.x = x$.

Dans cette preuve et dans celles qui suivent, nous avons omis la justification des égalités. Nous suggérons au lecteur de vérifier ces preuves en y ajoutant les justifications omises.

$$\begin{aligned} x &= x.1 \\ &= x.(x + \bar{x}) \\ &= x.x + x.\bar{x} \\ &= x.x + 0 \\ &= x.x \end{aligned}$$

5. Idempotence de la somme : $x + x = x$.

$$\begin{aligned}x &= x + 0 \\&= x + x.\bar{x} \\&= (x + x).(x + \bar{x}) \\&= (x + x).1 \\&= x + x\end{aligned}$$

6. 1 est élément absorbant de la somme : $1 + x = 1$.

Nous utilisons l'idempotence de la somme.

$$\begin{aligned}1 + x &= (x + \bar{x}) + x \\&= x + \bar{x} \\&= 1\end{aligned}$$

7. 0 est élément absorbant du produit : $0.x = 0$.

Nous utilisons l'idempotence du produit.

$$\begin{aligned}0.x &= (x.\bar{x}).x \\&= x.\bar{x} \\&= 0\end{aligned}$$

8. Lois de De Morgan :

$$\text{— } \overline{x.y} = \bar{x} + \bar{y} :$$

Nous utilisons l'unicité de la négation (propriété 1.5.3 page 27). Nous montrons d'abord que $x.y + (\bar{x} + \bar{y}) = 1$

$$\begin{aligned} x.y + (\bar{x} + \bar{y}) &= (x + \bar{x} + \bar{y}).(y + \bar{x} + \bar{y}) \\ &= (1 + \bar{y}).(1 + \bar{x}) \\ &= 1.1 \\ &= 1 \end{aligned}$$

Nous prouvons aussi que $x.y.(\bar{x} + \bar{y}) = 0$.

$$\begin{aligned} x.y.(\bar{x} + \bar{y}) &= x.y.\bar{x} + x.y.\bar{y} \\ &= 0.y + x.0 \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

Puisque la négation est unique, $\bar{x} + \bar{y}$ est bien la négation de xy .

— $\overline{x + y} = \bar{x}.\bar{y}$.

Nous utilisons l'unicité de la négation. Nous montrons d'abord que $(x + y) + \bar{x}.\bar{y} = 1$

$$\begin{aligned} (x + y) + \bar{x}.\bar{y} &= (x + y + \bar{x}).(\bar{x} + \bar{y}) \\ &= (1 + \bar{y}).(1 + \bar{x}) \\ &= 1.1 \\ &= 1 \end{aligned}$$

Nous prouvons aussi que $(x + y).\bar{x}.\bar{y} = 0$.

$$\begin{aligned} (x + y).\bar{x}.\bar{y} &= (x.\bar{x}.\bar{y}) + (y.\bar{x}.\bar{y}) \\ &= (0.\bar{y}) + (0.\bar{x}) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

De l'unicité de la négation, nous concluons que $\bar{x}.\bar{y}$ est bien la négation de $(x + y)$.

□

1.5.3 Dualité

La présentation de l'algèbre de Boole montre que 0 et 1, la somme et le produit jouent des rôles symétriques. Dans ce paragraphe, nous nous restreignons aux formules qui s'écrivent avec ces symboles plus la négation. La *duale* d'une formule est obtenue en échangeant les produits et les sommes ainsi que les \perp et les \top .

Définition 1.5.5 (Formule duale) Nous notons A^* la formule duale de A , définie de manière inductive par :

- $x^* = x$, si x est une variable,
- $\perp^* = \top$,
- $\top^* = \perp$,
- $(\neg A)^* = (\neg A^*)$,
- $(A \vee B)^* = (A^* \wedge B^*)$,
- $(A \wedge B)^* = (A^* \vee B^*)$.

Exemple 1.5.6 $(a \wedge (\neg b \vee c))^* =$

$$(a \vee (\neg b \wedge c)).$$

Théorème 1.5.7 En logique propositionnelle, si deux formules sont équivalentes, alors leurs duales sont aussi équivalentes.

Preuve : Voir exercice 29 page 39. □

Corollaire 1.5.8 En logique propositionnelle, si une formule est valide, sa duale est contradictoire.

Preuve : Voir exercice 29 page 39. □

En fait les résultats ci-dessus sont vrais non seulement pour la logique propositionnelle, qui est une algèbre de Boole à deux éléments, mais aussi pour toute algèbre de Boole. Nous définissons l'égalité de deux formules dans une algèbre de Boole.

Définition 1.5.9 (Égalité booléenne) Une formule A est égale à une formule B dans une algèbre de Boole si :

- A et B sont syntaxiquement identiques,
- A et B constituent les deux membres d'un axiome de l'algèbre de Boole,
- B est égale à A (l'égalité est symétrique),
- il existe une formule C telle que A est égale à C et C est égale à B (transitivité de l'égalité),
- il existe deux formules C et D telles que C est égale à D , et B est obtenue en remplaçant dans A une occurrence de C par D .

Théorème 1.5.10 Si deux formules sont égales dans une algèbre de Boole, alors leurs duales sont aussi égales.

Preuve : La preuve est similaire à l'exercice 29 page 39 en remplaçant l'algèbre booléenne à deux éléments par une algèbre booléenne quelconque. □

1.6 Fonctions booléennes

Nous regardons maintenant nos formules comme des fonctions mathématiques sur le domaine $\mathbb{B} = \{0, 1\}$. Nous définissons formellement ce qu'est une fonction booléenne, puis nous revisitons les formes normales conjonctives et disjonctives avec ce formalisme. Remarquons que les fonctions booléennes sont des notions utilisées en cryptologie par exemple dans la construction de chiffrements symétriques (le chiffrement One-Time-Pad, un des chiffrements les plus sûrs est basé sur la fonction booléenne du « ou-exclusif »).

Définition 1.6.1 (Fonction booléenne) Une fonction booléenne f est une fonction dont les arguments et le résultat sont dans le domaine $\mathbb{B} = \{0, 1\}$.

$$f : \mathbb{B}^n \rightarrow \mathbb{B}$$

Exemple 1.6.2 Nous donnons des exemples et contre-exemples de fonctions booléennes :

- La fonction $f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x$

est une fonction booléenne.

- La fonction $f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = x \bmod 2$

n'est pas une fonction booléenne.

- La fonction $f : \mathbb{B} \rightarrow \mathbb{N} : f(x) = x + 1$

n'est pas une fonction booléenne.

- La fonction $f : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B} : f(x, y) = \neg(x \wedge y)$

est une fonction booléenne.

1.6.1 Fonctions booléennes et somme de monômes

Grâce aux fonctions booléennes, nous sommes capables à partir des valeurs de vérité d'une formule de reconstruire la somme de monômes correspondante. Pour toute variable x , nous posons $x^0 = \bar{x}$ et $x^1 = x$.

Théorème 1.6.3 Soit f une fonction booléenne à n arguments. Cette fonction est représentée à l'aide de n variables x_1, \dots, x_n . Soit A la formule suivante :

$$A = \sum_{f(a_1, \dots, a_n)=1} x_1^{a_1} \dots x_n^{a_n}.$$

Les a_i sont des valeurs booléennes et A est la somme des monômes $x_1^{a_1} \dots x_n^{a_n}$ tels que $f(a_1, \dots, a_n) = 1$. Par convention, si la fonction f vaut toujours 0 alors $A = 0$.

Pour toute assignation v telle que $v(x_1) = a_1, \dots, v(x_n) = a_n$, nous avons $f(a_1, \dots, a_n) = [A]_v$. En omettant la distinction entre une variable et sa valeur, ce résultat s'écrit plus brièvement : $f(x_1, \dots, x_n) = A$.

Avant de lire la preuve, nous conseillons d'examiner l'exemple 1.6.4. La preuve n'est que la généralisation de cet exemple avec des notations, qui peuvent sembler complexes.

Preuve : Soit v une assignation quelconque. Notons que pour toute variable x , $[x^a]_v = 1$ si et seulement si $v(x) = a$. De cette remarque, nous déduisons la propriété suivante :

$$[x_1^{a_1} \dots x_n^{a_n}]_v = 1 \quad \text{si et seulement si} \quad v(x_1) = a_1, \dots, v(x_n) = a_n. \quad (1.1)$$

Soient a_1, \dots, a_n une liste de n valeurs booléennes et v une assignation telle que $v(x_1) = a_1, \dots, v(x_n) = a_n$. Considérons les deux cas suivants :

1. $f(a_1, \dots, a_n) = 1$: Le monôme $x_1^{a_1} \dots x_n^{a_n}$ est alors un des monômes de A . D'après 1.1, nous avons $[x_1^{a_1} \dots x_n^{a_n}]_v = 1$. Puisque, par définition de A , ce monôme est élément de la somme A , nous avons $[A]_v = 1$.
2. $f(a_1, \dots, a_n) = 0$: Supposons, par contradiction, que $[A]_v = 1$. Il existe alors un monôme de A , $x_1^{b_1} \dots x_n^{b_n}$, tel que $[x_1^{b_1} \dots x_n^{b_n}]_v = 1$. Par définition de A , nous avons $f(b_1, \dots, b_n) = 1$. Or, d'après 1.1, nous avons $[x_1^{b_1} \dots x_n^{b_n}]_v = 1$ si et seulement si $v(x_1) = b_1, \dots, v(x_n) = b_n$, donc par définition de v , $a_1 = b_1, \dots, a_n = b_n$. Nous obtenons donc une contradiction avec $f(a_1, \dots, a_n) = 0$, par suite $[A]_v = 0$.

□

Exemple 1.6.4 La fonction *maj* à 3 arguments vaut 1 lorsqu'au moins 2 de ses arguments valent 1. Nous donnons la table représentant cette fonction :

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	$\bar{x}_1 x_2 x_3$	$x_1 \bar{x}_2 x_3$	$x_1 x_2 \bar{x}_3$	$x_1 x_2 x_3$	$\bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	1
1	1	0	1	0	0	1	0	1
1	1	1	1	0	0	0	1	1

Nous notons sur la table de vérité, que chaque monôme ne vaut 1 que sur une ligne de la table et que la somme des monômes est équivalente à $maj(x_1, x_2, x_3)$. La fonction *maj* peut donc s'exprimer par

$$maj(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3.$$

1.6.2 Fonctions booléennes et produit de clauses

Nous effectuons le même raisonnement pour la forme duale.

Théorème 1.6.5 Soit f une fonction booléenne à n arguments. Cette fonction est représentée à l'aide de n variables x_1, \dots, x_n . Soit A la formule suivante :

$$A = \prod_{f(a_1, \dots, a_n)=0} x_1^{\bar{a}_1} + \dots + x_n^{\bar{a}_n}.$$

Les a_i sont des valeurs booléennes et A est le produit des clauses $x_1^{\bar{a}_1} + \dots + x_n^{\bar{a}_n}$ telles que $f(a_1, \dots, a_n) = 0$. Par convention si la fonction f vaut toujours 1 alors $A = 1$.

Pour toute assignation v telle que $v(x_1) = a_1, \dots, v(x_n) = a_n$, nous avons $f(a_1, \dots, a_n) = [A]_v$. En omettant la distinction entre une variable et sa valeur, ce résultat s'écrit plus brièvement : $f(x_1, \dots, x_n) = A$.

Preuve :

Soit v une assignation quelconque. Notons que pour toute variable x , $[x^a]_v = 0$ si et seulement si $v(x) \neq a$. De cette remarque, nous déduisons la propriété suivante :

$$[x_1^{\overline{a_1}} + \dots + x_n^{\overline{a_n}}]_v = 0 \quad \text{si et seulement si} \quad v(x_1) \neq \overline{a_1}, \dots, v(x_n) \neq \overline{a_n} \quad (1.2)$$

$$\text{si et seulement si} \quad v(x_1) = a_1, \dots, v(x_n) = a_n. \quad (1.3)$$

Des propriétés ci-dessus, nous déduisons comme précédemment que $f(x_1, \dots, x_n) = A$.

□

Exemple 1.6.6 Nous représentons la fonction *maj* définie dans l'exemple 1.6.4 page ci-contre par un produit de clauses :

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	$x_1 + x_2 + x_3$	$x_1 + x_2 + \overline{x_3}$	$x_1 + \overline{x_2} + x_3$	$\overline{x_1} + x_2 + x_3$	$\frac{(x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})}{(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)}$
0	0	0	0	0	1	1	1	0
0	0	1	0	1	0	1	1	0
0	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0
1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

La fonction *maj* peut donc s'exprimer en produit de clauses comme suit :

$$maj(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$$

1.7 L'outil BDDC

BDDC (*Binary Decision Diagram based Calculator*) est un outil pour la manipulation de formules propositionnelles développé par Pascal Raymond et disponible à l'adresse suivante :

<http://www-verimag.imag.fr/~raymond/tools/bddc-manual/>.

Cet outil est une « calculatrice propositionnelle » basée sur les *diagrammes de décision binaires*. Ces diagrammes permettent d'avoir une représentation interne des formules logiques sous la forme d'un graphe orienté acyclique ayant pour propriété d'être canonique. Cette calculatrice offre notamment la possibilité :

- d'évaluer si une formule est une tautologie,
- d'évaluer si une formule a un modèle,
- de transformer une formule en forme normale conjonctive,
- de transformer une formule en forme normale disjonctive,
- ...

Ainsi, il est possible d'utiliser BDDC pour résoudre plusieurs des exercices proposés ci-après, parmi lesquels les exercices 8 page 36 à 16 page 37 et 23 page 38 à 24 page 38.

1.8 Exercices

Exercice 1 (Formules strictes, formules à priorité) Parmi les expressions suivantes, indiquer celles qui sont et celles qui ne sont pas des formules, au sens de la syntaxe stricte décrite dans la définition 1.1.1 page 10 :

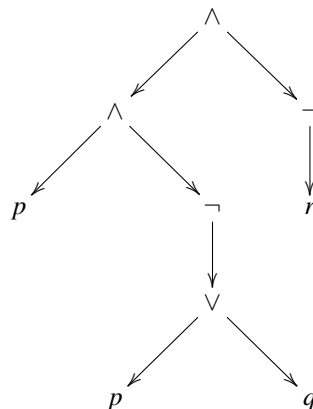
1. x
2. $(x$
3. (x)
4. $(x \Rightarrow (y \wedge z)$
5. $(x \Rightarrow (y \wedge z))$
6. $\neg(x \vee y)$
7. $(\neg(x \vee y))$
8. $\neg(x \Rightarrow y) \vee \neg(y \wedge z)$
9. $\neg(x)$
10. $(\neg((x \vee y) \wedge z) \Leftrightarrow (u \Rightarrow v)))$

□

Exercice 2 (\Leftrightarrow Formules et priorités) Montrer que toute formule est une formule à priorité.

□

Exercice 3 (Formules et priorités) L'arbre suivant représente la structure de la formule $((p \wedge \neg(p \vee q)) \wedge \neg r)$.



En utilisant les priorités, elle peut s'écrire avec le moins de parenthèses possibles sous l'une des formes : $p \wedge \neg(p \vee q) \wedge \neg r$ avec les notations logiques, $p.\overline{p} + q.\overline{r}$ avec les notations booléennes. Comme sur cet exemple, pour chacune des formules suivantes, indiquer sa structure sous forme d'arbre, sa forme la moins parenthésée avec les opérations logiques, sa forme la moins parenthésée avec les opérations booléennes :

1. $(\neg(a \wedge b) \Leftrightarrow (\neg a \vee \neg b))$.
2. $((\neg a \vee b) \wedge (\neg b \vee a))$.
3. $((((a \wedge b) \wedge c) \vee ((\neg a \wedge \neg b) \wedge \neg c))$.
4. $(p \Rightarrow (q \Rightarrow r))$.
5. $((p \Rightarrow q) \Rightarrow r)$.

□

Exercice 4 (Formules et priorités) Indiquer sous forme d'arbres, les structures des formules à priorité suivantes :

1. $p \Leftrightarrow \neg q \vee r$.
2. $p \vee q \Rightarrow r \wedge s$.
3. $p \vee q \Rightarrow r \Leftrightarrow s$.
4. $p \vee q \wedge r \Rightarrow \neg s$.
5. $p \Rightarrow r \wedge s \Rightarrow t$.

6. $p \vee q \wedge s \vee t$.
7. $p \wedge q \Leftrightarrow \neg r \vee s$.
8. $\neg p \wedge q \vee r \Rightarrow s \Leftrightarrow t$.

□

Exercice 5 (Hauteur,*) Nous définissons la hauteur h d'une formule à priorité par :

- $h(\top) = 0$ et $h(\perp) = 0$.
- Si A est une variable $h(A) = 0$.
- $h(\neg A) = 1 + h(A)$.
- $h((A)) = h(A)$.
- $h(A \circ B) = \max(h(A), h(B)) + 1$, si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$.

1. Montrer que cette définition est ambiguë, c'est-à-dire trouver une formule qui peut avoir deux hauteurs différentes.
2. Proposer une nouvelle définition récursive de la hauteur n'ayant pas ce problème.

□

Exercice 6 (Validité) Donner les tables de vérités des formules suivantes :

1. $p \Rightarrow (q \Rightarrow p)$.
2. $p \Rightarrow (q \Rightarrow r)$.
3. $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$.
4. $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$.

Indiquer celles qui sont valides. Indiquer les équivalences.

□

Exercice 7 (\Leftrightarrow Raisonnement circulaire) Donner les tables de vérité des formules suivantes : $a \Rightarrow b$, $b \Rightarrow c$, $c \Rightarrow a$, $a \Leftrightarrow b$, $b \Leftrightarrow c$ et $c \Leftrightarrow a$. Conclure que $(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge (c \Rightarrow a) \equiv (a \Leftrightarrow b) \wedge (b \Leftrightarrow c) \wedge (c \Leftrightarrow a)$.

□

Exercice 8 (Équivalence) Donner les tables de vérités des formules suivantes :

1. $p \wedge (p \vee q)$.
2. $\neg p \wedge \neg q$.
3. $\neg(p \wedge q)$.
4. $\neg(p \vee q)$.
5. $p \vee (p \wedge q)$.
6. $\neg p \vee \neg q$.
7. p .

Parmi ces formules, indiquer les formules équivalentes.

□

Exercice 9 (Équivalence) Donner les tables de vérités des formules suivantes :

1. $(p \Rightarrow q) \wedge (q \Rightarrow p)$.
2. $p \Rightarrow q$.
3. $p \Leftrightarrow q$.
4. $\neg q \Rightarrow \neg p$.
5. $(p \wedge q) \vee (\neg p \wedge \neg q)$.

Parmi ces formules, Indiquer les formules équivalentes.

□

Exercice 10 (Équivalence) Parmi les formules suivantes, indiquer celles qui sont équivalentes à $p \Rightarrow q \vee r$?

1. $q \wedge \neg r \Rightarrow p$.
2. $p \wedge \neg r \Rightarrow q$.
3. $\neg q \wedge \neg r \Rightarrow \neg p$.
4. $q \vee \neg p \vee r$.

□

Exercice 11 (\Leftrightarrow **Modèle d'un ensemble de formules,***) Montrer qu'une assignation est un modèle d'un ensemble de formules, si et seulement si elle est un modèle de la conjonction des formules de l'ensemble. □

Exercice 12 (\Leftrightarrow **Lois de simplification**) Prouver que pour tout x, y :

- $x \vee (x \wedge y) \equiv x$.
 - $x \wedge (x \vee y) \equiv x$.
 - $x \vee (\neg x \wedge y) \equiv x \vee y$.
 - $x \wedge (\neg x \vee y) \equiv x \wedge y$.
-

Exercice 13 (\Leftrightarrow **Simplification de formule,***) Montrer par simplification que la formule suivante est une tautologie.

$$(a \vee b) \wedge (\neg b \vee c) \Rightarrow (a \vee c)$$
□

Exercice 14 (**Modèles et formes normales**) Soit A la formule suivante :

$$(((a \Rightarrow \neg b) \Leftrightarrow \neg c) \wedge (c \vee d)) \wedge (a \Leftrightarrow d).$$

1. A est-elle une tautologie ? (justifier)
 2. A est-elle une contradiction ? (justifier)
 3. Donner la forme normale conjonctive de A (détailler).
 4. Donner la forme normale disjonctive de A (détailler).
-

Exercice 15 (\Leftrightarrow **Récurrence,***) Montrer que toute formule à une seule variable construite sans négation et uniquement avec \vee et \wedge est équivalente à une formule de taille 0. □

Exercice 16 (**Algèbre de Boole**) À l'aide de tables de vérité, indiquer pour les opérations suivantes $\Rightarrow, \Leftrightarrow$ si elles sont commutatives, associatives, idempotentes, transitives. □

Exercice 17 (**Algèbre de Boole**) Montrer qu'une formule avec une seule variable, notée p , est équivalente à l'une des formules, 0, 1, p , $\neg p$. □

Exercice 18 (**Algèbre de Boole**) Donner 16 formules telles que toute formule avec deux variables, notées p et q , soit équivalente à l'une des 16 formules. □

Exercice 19 (**Fonction booléenne**) Indiquer en fonction de k combien il y a de fonctions booléennes à k arguments. □

Exercice 20 (**Conséquence**) Au cours de l'enquête dont il est chargé, l'adjudant Tifrice effectue le raisonnement suivant :

- Si le meurtre a eu lieu le jour, le meurtrier est l'ami de la victime
- Or le meurtre a eu lieu la nuit

Donc le meurtrier n'est pas l'ami de la victime. Le raisonnement de l'adjudant Tifrice est-il correct ? Pour cela nous procédons en trois étapes :

1. Formalisation des faits.
 2. Formalisation du raisonnement permettant de déduire la conclusion à partir des hypothèses.
 3. Démontrer que le raisonnement est correct ou non.
-

Exercice 21 (**Conséquence**) Pinocchio, Quasimodo et Roméo s'apprêtent à chanter en canon. Ils décident entre eux que :

1. Si Pinocchio ne chante pas alors Quasimodo chantera.
2. Si Quasimodo chante alors Pinocchio et Roméo chanteront.
3. Si Roméo chante alors Quasimodo ou Pinocchio, l'un des deux au moins, ne chantera pas.

Peut-on en conclure que Pinocchio chantera ? Justifier votre réponse en formalisant le raisonnement. □

Exercice 22 (Conséquence) Formaliser les énoncés suivants en dégagant les connecteurs logiques.

- (a) Si Pierre est rentré chez lui, alors Jean est allé au cinéma.
- (b) Marie est à la bibliothèque ou Pierre est rentré chez lui ou les deux.
- (c) Si Jean est allé au cinéma, alors Marie est à la bibliothèque ou Pierre est rentré chez lui ou les deux.
- (d) Marie n'est pas à la bibliothèque et Jean est allé au cinéma.
- (e) Pierre est rentré chez lui.

Le dernier énoncé est-il conséquence des énoncés qui le précèdent ? □

Exercice 23 (Formes normales) Pour chaque formule donnée ci-après, donner sa forme normale disjonctive et prouver si elle est ou non satisfaisable (en donnant si besoin un modèle de la formule).

- $\neg(a \Leftrightarrow b) \vee (b \wedge c) \Rightarrow c$.
 - $(a \Rightarrow b) \wedge (b \Rightarrow \neg a) \wedge (\neg a \Rightarrow b) \wedge (b \Rightarrow a)$.
-

Exercice 24 (Formes normales) Soit A la formule $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$. Transformer A en une fnc. La formule A est-elle valide ? □

Exercice 25 (\Leftrightarrow Formes normales,)** Nous revenons sur les preuves des transformations en forme normale.

1. Montrer que l'application de l'élimination des équivalences, des implications et de déplacement des négations sur une formule donne une formule en forme normale.
 2. Montrer aussi que quel que soit l'ordre d'application de l'élimination des équivalences, des implications et de déplacement des négations sur une formule, l'algorithme de transformation en forme normale termine.
-

Exercice 26 (Conséquence et formes normales) L'adjudant Tifrice enquête une nouvelle fois. Ses hypothèses sont les suivantes :

- Si Jean n'a pas rencontré Pierre l'autre nuit, c'est que Pierre est le meurtrier ou que Jean est un menteur
- Si Pierre n'est pas le meurtrier, alors Jean n'a pas rencontré Pierre l'autre nuit et le crime a eu lieu après minuit
- Si le crime a eu lieu après minuit, alors Pierre est le meurtrier ou Jean est un menteur.

Il en conclut que Pierre est le meurtrier. Son raisonnement est-il correct ? Donner la réponse en construisant la conjonction des hypothèses et de la négation de la conclusion et en mettant cette conjonction sous forme de somme de monômes. Nous rappelons que le raisonnement est incorrect si et seulement si un des monômes de la somme ne comporte pas de littéraux complémentaires : ce monôme donne un modèle des hypothèses, qui est un contre-modèle de la conclusion. □

Exercice 27 (Formalisation, somme de monômes,*) Nous nous trouvons sur une île dont les indigènes sont répartis en deux tribus, les Purs et les Pires. Les Purs disent toujours la vérité tandis que les Pires mentent toujours. Nous rencontrons deux indigènes, Aha et Bébé⁴.

- (a) Aha déclare : « L'un de nous au moins est un Pire ». Pouvons-nous en déduire ce que sont Aha et Bébé ?
 - (b) Aha déclare : « L'un de nous deux au plus est un Pire ». Pouvons-nous en déduire ce que sont Aha et Bébé ?
 - (c) Aha déclare : « Nous sommes tous les deux de la même tribu ». Pouvons-nous en déduire ce que sont Aha et Bébé ?
-

Exercice 28 (Ensemble complet et incomplet de connecteurs,)** Un ensemble de constantes et de connecteurs est dit complet, si toute fonction booléenne est exprimable avec ces constantes et ces connecteurs. D'après le théorème 1.6.3 page 32, l'ensemble $\{0, 1, -, +, \bullet\}$ est complet.

1. Montrer que l'ensemble $\{-, +\}$ est complet.
Indication : il suffit de montrer que $\{0, 1, \bullet\}$ sont définissables à l'aide de $-$ et $+$.
2. Montrer que l'ensemble $\{0, \Rightarrow\}$ est complet.

4. Cette énigme provient du livre de Raymond M. Smullyan : « Quel est le titre de ce livre ? », qui contient bien d'autres problèmes amusants à propos de ces indigènes.

3. Soit $|$ l'opération suivante : $x | y$ est vrai si et seulement si ni l'un, ni l'autre ne sont vrais, autrement dit $x | y = 1$ si et seulement si $x = 0$ et $y = 0$. Cette opération est aussi appelée ni car $x | y$ est vrai si et seulement si ni x , ni y ne sont vrais.

Montrer que cette opération est complète.

4. (**) Montrer que l'ensemble $\{0, 1, +, \bullet\}$ est incomplet.

Indication : nous trouvons une propriété caractéristique des fonctions booléennes définies avec ces opérations et qui n'est pas vérifiée par toute fonction booléenne.

Nous appelons monotone, une fonction f telle que si $a_1 \leq b_1, \dots, a_n \leq b_n$ alors $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$.

Montrer que toute fonction booléenne définie avec $\{0, 1, +, \bullet\}$ est monotone.

Proposer une fonction booléenne non monotone.

5. (*) L'ensemble $\{1, \Rightarrow\}$ est-il complet ?

6. (**) L'ensemble $\{0, \Leftrightarrow\}$ est-il complet ?

□

Exercice 29 (Dualité, récurrence, **) Considérons des formules dont les opérations sont limitées à $0, 1, \neg, \wedge, \vee$. Nous rappelons que la formule duale d'une formule est obtenue en échangeant les conjonctions et les disjonctions ainsi que les 0 et les 1 et que nous notons A^* la duale de la formule A .

1. Montrer que si 2 formules sont équivalentes, il en est de même de leurs duales. Nous donnons une indication qui réduit la question à une récurrence simple. Soit v une assignation. Soit \bar{v} l'assignation complémentaire de v : pour toute variable x , $\bar{v}(x) = \overline{v(x)}$. Par exemple l'assignation complémentaire de $a = 1, b = 0$ est $a = 0, b = 1$.

(a) Montrer que pour toute formule A : $[A^*]_v = \overline{[A]_{\bar{v}}}$.

(b) En déduire que si 2 formules sont équivalentes, il en est de même de leurs duales.

2. En déduire que si une formule est valide, sa duale est contradictoire.

□

Exercice 30 (Représentation canonique, *)** Soit $x_i (0 \leq i \leq n)$ une liste de n variables distinctes. Dans cet exercice, les variables des formules sont toutes dans cette liste. À cette liste et à une formule A , nous associons la fonction booléenne f à n arguments définie par $f(x_1, \dots, x_n) = A$. Cette notation signifie que pour $a_1, \dots, a_n \in \mathbb{B}$, la valeur de $f(a_1, \dots, a_n)$ est la valeur de la formule A quand $x_1 = a_1, \dots, x_n = a_n$.

1. Combien y a-t-il de classes de formules ?

2. Notons par \oplus le ou-exclusif. Nous avons $x \oplus y = 1$ si et seulement soit $x = 1$, soit $y = 1$ mais pas $x = y = 1$ (d'où le nom de ou-exclusif). Nous pouvons aussi définir cette opération par $x \oplus y = \neg(x \Leftrightarrow y)$. Il est clair que cette opération est commutative, associative, que l'opération \wedge est distributive sur \oplus et que 0 est l'élément neutre de \oplus . Exprimer la négation, la disjonction, l'implication et l'équivalence comme somme ou-exclusive de 1 et de conjonctions de variables.

3. Montrer que toute formule booléenne est équivalente à une somme ou-exclusive de 1 et de produits de variables.

4. Par convention 1 est considéré comme le produit de zéro variable et 0 comme la somme ou-exclusive de zéro produit de variables. Montrer que toute formule booléenne est équivalente à une somme ou-exclusive de produits dont les ensembles de variables sont tous distincts.

5. Fixons un ordre arbitraire des n variables utilisées dans les formules (lorsque les variables sont des identificateurs, cet ordre est souvent l'ordre lexicographique). Une somme ou-exclusive de produits de variables est canonique (forme de Reed-Muller) si et seulement si :

- ses produits ne comportent pas de variables répétées et dans un produit les variables sont ordonnées de gauche à droite dans l'ordre croissant,
- les produits sont ordonnés de gauche à droite dans l'ordre lexicographique décroissant déduit de l'ordre des variables.

Montrer que toute formule booléenne est équivalente à une somme ou-exclusive canonique. En déduire l'unicité de cette forme canonique.

□

Chapitre 2

Résolution propositionnelle

Sommaire

2.1	Résolution	41
2.1.1	Définitions	41
2.1.2	Cohérence	44
2.1.3	Complétude pour la réfutation	45
2.1.4	Réduction d'un ensemble de clauses	47
2.1.5	Clauses minimales pour la déduction	48
2.1.6	Clauses minimales pour la conséquence	48
2.2	Stratégie complète	50
2.2.1	Algorithme de la stratégie complète	50
2.2.2	Arrêt de l'algorithme	52
2.2.3	Le résultat de l'algorithme est équivalent à l'ensemble initial de clauses	53
2.2.4	Le résultat de l'algorithme est l'ensemble des clauses minimales pour la déduction de l'ensemble initial de clauses	53
2.3	Algorithme de Davis-Putnam-Logemann-Loveland	55
2.3.1	Suppression des clauses contenant des littéraux isolés	55
2.3.2	Résolution unitaire	56
2.3.3	Suppression des clauses valides	57
2.3.4	L'algorithme	57
2.3.5	Solveurs SAT	60
2.4	Exercices	62

LES tables de vérité et les transformations en sommes de monômes ou produits de clauses permettent de répondre aux questions : une formule est-elle valide ? et un raisonnement est-il correct ? Cependant le temps de réponse de ces méthodes augmente exponentiellement avec le nombre de variables. Pour illustrer cela, considérons la conséquence suivante :

$$a \Rightarrow b, b \Rightarrow c, c \Rightarrow d, d \Rightarrow e, e \Rightarrow f, f \Rightarrow g, g \Rightarrow h, h \Rightarrow i, i \Rightarrow j \models a \Rightarrow j$$

En utilisant une table de vérité il faut tester $2^{10} = 1024$ lignes. Or par *déduction*, en utilisant la transitivité de l'implication, nous savons immédiatement que le raisonnement ci-dessus est correct. Ainsi, en logique propositionnelle, les déductions peuvent s'avérer très utiles, voire nécessaires lorsque la taille du problème à traiter est importante. Nous étudions maintenant un système de déduction très simple. Dans ce système, les hypothèses et les formules déduites ne sont pas des formules quelconques mais des *clauses*, c'est-à-dire des disjonctions de littéraux. De plus, le système comporte une seule règle appelée *la résolution*.

Plan : Nous introduisons d'abord la méthode de *résolution* inventée en 1965 par J. Alan Robinson [?] dans le cadre plus général de la recherche de preuve au premier ordre. Nous présentons ensuite une procédure pour calculer tous les résol-vants possibles, c'est-à-dire toutes les applications possibles de la règle de résolution : la *stratégie complète*. Puis, nous étudions l'algorithme DPLL, proposé dans les années soixante par Martin Davis, Hilary Putnam, George W. Logemann et Donald W. Loveland [?, ?]. En utilisant la notion de résolution, des méthodes de simplifications et le branchement/retour

arrière, cet algorithme prouve efficacement si une formule a un modèle ou non. Bien qu'inventé il y a plus de quarante ans, il est toujours à la base de la plupart des solveurs SAT complets actuels. Notons que dans ce chapitre nous adoptons la notation booléenne plus concise.

2.1 Résolution

Nous souhaitons prouver que $a + c$ est une conséquence de $a + b$ et $\bar{b} + c$, c'est-à-dire $a + b, \bar{b} + c \models a + c$. En utilisant la propriété 1.2.27 page 17, il suffit de démontrer que la formule $(a + b).(\bar{b} + c) \Rightarrow (a + c)$ est valide. Pour cela, nous pouvons utiliser une table de vérité (8 lignes) ou alors une simplification de formules comme dans l'exercice 13 page 37. Ces deux méthodes sont fastidieuses même pour des formules aussi simples (3 variables uniquement). En utilisant la règle de résolution présentée dans ce chapitre, nous déduisons directement la clause $a + c$ à partir des deux clauses $a + b$ et $\bar{b} + c$.

Nous donnons maintenant la définition de la résolution et certaines propriétés utiles qui en dérivent. Nous prouvons ensuite la *cohérence* et la *complétude pour la réfutation* de notre système basé sur la déduction. La cohérence nous permet d'affirmer que si une clause C est obtenue par résolution à partir d'un ensemble Γ de clauses alors elle en est conséquence, c'est-à-dire tout modèle de Γ est modèle de C . La complétude pour la réfutation nous permet d'affirmer que si \perp est conséquence de Γ , autrement dit si Γ est insatisfaisable, alors \perp peut être retrouvé par résolution à partir de Γ .

2.1.1 Définitions

Dans un premier temps nous introduisons les notions nécessaires à la définition de la résolution. Nous rappelons que les notions de clauses et littéraux sont définies dans la section 1.4 du chapitre précédent (page 22). Comme nous le verrons dans la suite deux clauses qui ont le même ensemble de littéraux jouent le même rôle dans la résolution. Aussi, nous définissons deux clauses égales comme étant deux clauses avec le même ensemble de littéraux.

Définition 2.1.1 (Éléments d'une clause, clauses égales et sous-clauses)

- Un littéral est élément d'une clause, s'il est élément de l'ensemble des littéraux de la clause.
- Une clause A est incluse dans une clause B , si tous les littéraux de la clause A sont éléments de la clause B . Dans ce cas, A est une sous-clause de B .
- Deux clauses sont égales si elles ont le même ensemble de littéraux.

Exemple 2.1.2 Nous illustrons les notions introduites par des exemples simples :

- Le littéral p est élément de la clause $\bar{q} + p + r + p$.
- La clause $p + \bar{q}$ est incluse dans la clause $\bar{q} + p + r + p$.
- Enlever le littéral p de la clause $\bar{q} + p + r + p$ donne la clause $\bar{q} + r$, enlever le littéral p de la clause $p + p + p$ donne la clause \perp .
- Ajouter le littéral r à la clause p donne la clause $p + r$, ajouter le littéral p à la clause \perp donne la clause p .
- Les clauses $p + \bar{q}$, $\bar{q} + p$, et $p + \bar{q} + p$ sont égales.

Notation : Nous notons $s(A)$ l'ensemble des littéraux de la clause A . Par convention \perp est la clause vide et $s(\perp) = \emptyset$.

Exemple 2.1.3 $s(\bar{q} + p + r + p + \bar{p}) =$

$$\{\bar{q}, p, r, \bar{p}\}.$$

Dans l'exercice 37 page 63 nous demandons de formaliser ce qu'est une clause et son ensemble de littéraux. Nous introduisons la notion de littéral *complémentaire*, noté L^c pour un littéral L , afin de manipuler dans nos raisonnements

uniquement des littéraux. En effet, la négation ne suffit pas car $\bar{\bar{L}}$ est équivalent à L mais n'est pas un littéral : un littéral est une variable ou la négation d'une variable mais la double négation d'une variable n'est pas un littéral.

Définition 2.1.4 (Littéral complémentaire) Notons L^c le littéral complémentaire d'un littéral L , défini par :

- Si L est une variable, L^c est la négation de L .
- Si L est la négation d'une variable, L^c est obtenue en enlevant la négation de L .

Exemple 2.1.5 $x^c = \bar{x}$ et $\bar{x}^c = x$.

Nous présentons le calcul de résolvant entre deux clauses, ce qui constitue l'unique règle du système de résolution.

Définition 2.1.6 (Résolvant) Soient A et B deux clauses. La clause C est un résolvant de A et B si et seulement s'il y a un littéral L tel que $L \in s(A)$, $L^c \in s(B)$ et $s(C) = (s(A) - \{L\}) \cup (s(B) - \{L^c\})$. Nous utilisons la représentation suivante lorsque C est un résolvant de A et B :

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

Si la clause C est un résolvant de A et B , alors nous disons que C est engendrée par A et B , et que A et B sont les parents de la clause C .

Exemple 2.1.7 Nous proposons quelques exemples de résolution :

$$\frac{p+q+r \quad p+\bar{q}+r}{p+r} \quad \frac{p+\bar{q} \quad \bar{p}+q+r}{\bar{p}+p+r} \quad \frac{p+\bar{q} \quad \bar{p}+q+r}{\bar{q}+q+r} \quad \frac{p \quad \bar{p}}{\perp}$$

Propriété 2.1.8 Si l'un des parents d'un résolvant est valide, le résolvant est valide ou contient l'autre parent.

Preuve : Cette preuve est demandée dans l'exercice 39 page 63. □

Étant données deux clauses A et B , la formule $A + B$ n'est pas une clause si l'un des deux opérandes de la disjonction est la clause vide. Par exemple $\perp + p$ n'est pas une clause, bien que les deux opérandes de la disjonction soient des clauses. Aussi nous introduisons un nouvel opérande, $\dot{+}$, afin de combiner la disjonction et l'élimination de \perp . Cet opérande va nous permettre de simplifier la définition de résolvant.

Définition 2.1.9 ($C \dot{+} D$) Soient C et D deux clauses. Nous notons $C \dot{+} D$ la clause suivante :

- Si $C = \perp$ alors $C \dot{+} D = D$,
- sinon si $D = \perp$ alors $C \dot{+} D = C$ sinon $C \dot{+} D = C + D$.

Ajouter le littéral L à la clause C , c'est construire $C \dot{+} L$. Munis de cette définition, nous pouvons reformuler plus simplement la règle de résolution.

Définition 2.1.10 (Résolvant) Soient A et B deux clauses. La clause C est un résolvant de A et B si et seulement s'il y a un littéral L tel que :

- L est élément de la clause A , L^c est élément de la clause B
- C est égale à la clause $A' \dot{+} B'$ où $A' = A - \{L\}$ est obtenue en enlevant L de A et $B' = B - \{L^c\}$ est obtenue en enlevant L^c de B .

La combinaison de résolutions successives permet de générer de nouvelles clauses. Une clause issue de ces résolvants sera une preuve par résolution de cette clause à partir d'un ensemble de clauses.

Définition 2.1.11 (Preuve) Soient Γ un ensemble de clauses et C une clause. Une preuve de C à partir de Γ est une liste de clauses qui se termine par C . Toute clause de la preuve est égale à un élément de Γ ou est un résolvant de deux clauses la précédant dans la preuve. La clause C est déduite de Γ , notée $\Gamma \vdash C$, s'il y a une preuve de C à partir de Γ .

Exemple 2.1.12 Soit Γ l'ensemble de clauses $\bar{p}+q, p+\bar{q}, \bar{p}+\bar{q}, p+q$. Nous montrons que $\Gamma \vdash \perp$ par la preuve suivante :

1	$p+q$	Hypothèse
2	$p+\bar{q}$	Hypothèse
3	p	Résolvant de 1, 2
4	$\bar{p}+q$	Hypothèse
5	q	Résolvant de 3, 4
6	$\bar{p}+\bar{q}$	Hypothèse
7	\bar{p}	Résolvant de 5, 6
8	\perp	Résolvant de 3, 7

La 3^e colonne de la preuve est la colonne des justifications qui permettent de suivre le déroulement de la preuve.

Nous pouvons aussi voir une preuve comme un *arbre* dont les feuilles sont les hypothèses et dont les sommets internes sont obtenus par construction des résolvants.

$p+q$	$p+\bar{q}$		
p	$\bar{p}+q$		
q	$\bar{p}+\bar{q}$	$p+q$	$p+\bar{q}$
\bar{p}		p	
\perp			

La présentation en arbre a un inconvénient, elle oblige à répéter des parties de preuve. Par exemple, dans la preuve en arbre précédente, nous répétons deux fois la preuve de p .

Définition 2.1.13 (Taille de preuve) Une preuve P de C à partir d'un ensemble de clauses Γ est de taille n si elle contient n lignes.

La taille de la preuve donnée dans l'exemple 2.1.12 page précédente est 8, ce qui n'est pas forcément intuitif sur la représentation en arbre. Nous utiliserons les deux propriétés énoncées ci-dessous souvent de manière implicite dans les preuves.

Propriété 2.1.14 (Monotonie et composition) Soient Γ, Δ deux ensembles de clauses et A, B deux clauses.

1. Monotonie de la déduction : Si $\Gamma \vdash A$ et si Γ est inclus dans Δ alors $\Delta \vdash A$
2. Composition des déductions : Si $\Gamma \vdash A$, $\Gamma \vdash B$ et si C est un résolvant de A et B alors $\Gamma \vdash C$.

Preuve : Demandée dans l'exercice 38 page 63.

□

2.1.2 Cohérence

La cohérence de la résolution signifie que si une clause C est obtenue après une à plusieurs applications de la règle de résolution à un ensemble de clauses Γ alors tout modèle de Γ est aussi modèle de C (autrement dit, C est une conséquence de Γ). La première partie de la preuve consiste à démontrer la cohérence d'une application de la règle de résolution (théorème 2.1.15). Nous généralisons ensuite par induction dans le théorème 2.1.16 page ci-contre.

Théorème 2.1.15 (Cohérence de la règle de résolution) *Si C est un résolvant de A et B alors $A, B \models C$.*

Preuve : Supposons que C soit un résolvant de A et B . Par définition il y a un littéral L tel que $L \in s(A), L^c \in s(B), s(C) = (s(A) - \{L\}) \cup (s(B) - \{L^c\})$. Soit v une assignation modèle de A et B , c'est-à-dire, $[A]_v = 1$ et $[B]_v = 1$. Montrons que v est modèle de C . Nous distinguons deux cas possibles :

- Supposons que v soit modèle de L . Dans ce cas v n'est pas modèle de L^c . L'assignation v est donc modèle d'un littéral de $(s(B) - \{L^c\})$ car $[B]_v = 1$. Nous concluons donc que v est modèle de C .
- Supposons que v soit modèle de L^c . Dans ce cas v n'est pas modèle de L . L'assignation v est donc modèle de $(s(A) - \{L\})$ car $[A]_v = 1$. Nous concluons donc que v est modèle de C .

Puisque toute assignation est modèle de L ou L^c , v est modèle de C .

□

Théorème 2.1.16 (Cohérence de la déduction) *Soient Γ un ensemble de clauses et C une clause. Si $\Gamma \vdash C$ alors $\Gamma \models C$.*

Preuve : Supposons que $\Gamma \vdash C$. Par définition, il existe une preuve P de C à partir de Γ . Supposons que pour toute preuve de D à partir de Γ , plus courte que P , nous avons $\Gamma \models D$. Montrons que $\Gamma \models C$. Nous avons deux cas possibles :

1. C a le même ensemble de littéraux qu'une clause E de Γ , dans ce cas tout modèle de Γ rendra vrai au moins un littéral de E . Or comme C est une clause composée des mêmes littéraux que E , elle aura au moins un littéral vrai et donc sera valide ainsi $E \models C$ par conséquence $\Gamma \models C$.
2. C n'a pas le même ensemble de littéraux qu'une clause de Γ , C est donc résolvant de deux clauses A et B précédant C dans la preuve P . Donc il y a une preuve de A et une preuve de B à partir de Γ plus courtes que P . Par hypothèse de récurrence, nous avons : $\Gamma \models A$ et $\Gamma \models B$. D'après le théorème 2.1.15 page précédente : $A, B \models C$, ainsi $\Gamma \models C$.

□

Corollaire 2.1.17 *D'après le théorème 2.1.16, si $\Gamma \vdash \perp$ alors $\Gamma \models \perp$, autrement dit Γ est insatisfaisable.*

2.1.3 Complétude pour la réfutation

La complétude pour la réfutation est la propriété suivante : si Γ est un ensemble insatisfaisable de clauses, alors il y a une preuve de la clause vide à partir de Γ . Nous montrons dans la suite de cette section la complétude pour la réfutation lorsque Γ est un ensemble *fini* de clauses.

Nous prouvons la complétude réfutationnelle par récurrence sur le nombre de variables. Pour cela, nous construisons à partir de Γ deux ensembles de clauses contenant tous deux une variable en moins. Plus précisément, nous construisons les ensembles $\Gamma[L := 1]$ et $\Gamma[L := 0]$ en assignant un littéral L de Γ à vrai et à faux, respectivement. Nous étudions ensuite les propriétés de ces deux ensembles par rapport à Γ (lemmes 2.1.22 page suivante et 2.1.23 page 47). Ces propriétés seront utilisées lors du pas de récurrence.

Définition 2.1.18 ($\Gamma[L := 1]$) Soient Γ un ensemble de clauses et L un littéral. $\Gamma[L := 1]$ est l'ensemble de clauses obtenu en supprimant les clauses dont L est élément et en enlevant L^c des autres clauses. Nous posons $\Gamma[L := 0] = \Gamma[L^c := 1]$.

$\Gamma[L := x]$ est égal à l'ensemble de clauses obtenu en donnant à L la valeur x ($x = 0$ ou $x = 1$) et en simplifiant le résultat obtenu.

Exemple 2.1.19 Soit Γ l'ensemble de clauses $\bar{p} + q, \bar{q} + r, p + q, p + r$. Nous avons :

$$- \Gamma[p := 1] =$$

$$\{q, \bar{q} + r\}.$$

$$- \Gamma[p := 0] =$$

$$\{\bar{q} + r, q, r\}.$$

Afin de donner une intuition de comment calculer $\Gamma[p := 1]$ et $\Gamma[p := 0]$, nous considérons le produit de clauses suivant $(\bar{p} + q)(\bar{q} + r)(p + q)(p + r)$ et observons :

$$- (\bar{1} + q)(\bar{q} + r)(1 + q)(1 + r) =$$

$$q(\bar{q} + r) = \Gamma[p := 1].$$

$$- (\bar{0} + q)(\bar{q} + r)(0 + q)(0 + r) =$$

$$(\bar{q} + r)qr = \Gamma[p := 0].$$

Nous notons $v[L := 1]$ l'assignation qui donne à L la valeur 1, à L^c la valeur 0 et ne change pas la valeur des autres littéraux.

Définition 2.1.20 ($v[L := 1]$) Soit une assignation v , l'assignation $v[L := 1]$ est l'assignation identique à v sauf éventuellement pour x la variable de L . Si $L = x$ alors $v[L := 1](x) = 1$, si $L = \bar{x}$ alors $v[L := 1](x) = 0$.

Nous posons $v[L := 0] = v[L^c := 1]$. La propriété suivante se déduit du fait que si Γ a un modèle v , alors v donne soit la valeur 1 soit la valeur 0 au littéral L .

Propriété 2.1.21 Soient Γ un ensemble de clauses et L un littéral. Γ a un modèle si et seulement si $\Gamma[L := 1]$ ou $\Gamma[L := 0]$ a un modèle.

Preuve :

\Rightarrow Supposons que l'assignation v soit modèle de Γ . Nous considérons deux cas selon la valeur de L dans cette assignation.

1. Supposons que v donne à L la valeur 1 et montrons que v est modèle de $\Gamma[L := 1]$. Soit C une clause de $\Gamma[L := 1]$. Il y a dans Γ une clause C' telle que C est obtenue en enlevant L^c de C' . Puisque v est modèle de Γ , v est modèle de C' donc d'un littéral qui n'est pas L^c (car L^c vaut 0 dans cette assignation). Par suite v est modèle de C . Puisque C est une clause quelconque de $\Gamma[L := 1]$, v est modèle de $\Gamma[L := 1]$.
2. Supposons que v donne à L la valeur 0. Nous nous ramenons au cas précédent en échangeant L et L^c et nous montrons que v est modèle de $\Gamma[L := 0]$.

\Leftarrow Nous considérons deux cas suivant que $\Gamma[L := 1]$ ou $\Gamma[L := 0]$ a un modèle.

1. Supposons que l'assignation v soit modèle de $\Gamma[L := 1]$. Montrons que $v[L := 1]$ est modèle de Γ . Soit C une clause de Γ .
 - (a) Supposons que L soit un littéral de C , alors $v[L := 1]$ est modèle de C puisque cette assignation donne à L la valeur 1.
 - (b) Supposons que L ne soit pas un littéral de C . Alors il y a une clause C' élément de $\Gamma[L := 1]$ telle que C' est obtenue en enlevant L^c de C . La variable de L n'est pas une variable de C' . Par suite v et $v[L := 1]$ donnent la même valeur à C' . Puisque v est modèle de $\Gamma[L := 1]$, v est modèle de C' donc $v[L := 1]$ est modèle de C' . Puisque C' est incluse dans C , $v[L := 1]$ est modèle de C .
 Puisque C est une clause quelconque de Γ , $v[L := 1]$ est modèle de Γ .
2. Supposons que l'assignation v est modèle de $\Gamma[L := 0]$. Par une preuve analogue, nous montrons que $v[L := 0]$ est modèle de Γ .

□

Lemme 2.1.22 Soient Γ un ensemble de clauses, C une clause et L un littéral. Si $\Gamma[L := 1] \vdash C$ alors $\Gamma \vdash C$ ou $\Gamma \vdash C \tilde{\vdash} L^c$.

Rappel : L'opération $\tilde{\vdash}$ est introduite dans la définition 2.1.9 page 43.

Preuve : Partant d'une preuve de C à partir de $\Gamma[L := 1]$, nous obtenons une preuve de C ou de $C \tilde{\vdash} L^c$ à partir de Γ en ajoutant le littéral L^c aux clauses où nous l'avions enlevé. Formalisons cette ébauche de preuve. Supposons que $\Gamma[L := 1] \vdash C$. Il y a une preuve P de C à partir de $\Gamma[L := 1]$. Supposons que pour toute preuve de D à partir de $\Gamma[L := 1]$, plus courte que P , nous avons $\Gamma \vdash D$ ou $\Gamma \vdash D \tilde{\vdash} L^c$. Nous avons deux cas possibles :

1. C est égal à un élément de $\Gamma[L := 1]$. Donc il y a une clause C' élément de Γ telle que $s(C') = s(C)$ ou $s(C') = s(C) \cup \{L^c\}$. Examinons ces deux cas.
 - (a) Supposons $s(C') = s(C)$. Par définition des preuves, $\Gamma \vdash C$.
 - (b) Supposons $s(C') = s(C) \cup \{L^c\}$. Nous avons $s(C') = s(C \tilde{\vdash} L^c)$ donc par définition des preuves, $\Gamma \vdash C \tilde{\vdash} L^c$.
2. C est résolvant de 2 clauses A et B précédant C dans la preuve P . Donc par hypothèse de récurrence :
 - $\Gamma \vdash A$ ou $\Gamma \vdash A \tilde{\vdash} L^c$.
 - $\Gamma \vdash B$ ou $\Gamma \vdash B \tilde{\vdash} L^c$.
 Ce qui donne 4 cas à examiner.
 - (a) Supposons $\Gamma \vdash A$ et $\Gamma \vdash B$. Puisque C est résolvant de A et B , d'après la propriété 2.1.14 page 44, nous avons $\Gamma \vdash C$.
 - (b) Supposons $\Gamma \vdash A$ et $\Gamma \vdash B \tilde{\vdash} L^c$. Puisque C est résolvant de A et B , il existe M tel que $M \in A$ et $M^c \in B$ et $s(C) = (s(A) - \{M\}) \cup (s(B) - \{M^c\})$. Aucune clause de $\Gamma[L := 1]$ ne comporte le littéral L^c . Donc B qui en est déduite, ne comporte pas le littéral L^c (voir exercice 40 page 63) et par suite $L^c \neq M^c$. Par conséquent $(s(B) - \{M^c\}) \cup \{L^c\} = (s(B) \cup \{L^c\}) - \{M^c\} = s(B \tilde{\vdash} L^c) - \{M^c\}$. Nous avons donc $s(C \tilde{\vdash} L^c) = (s(A) - \{M\}) \cup (s(B) - \{M^c\}) \cup \{L^c\} = (s(A) - \{M\}) \cup (s(B \tilde{\vdash} L^c) - \{M^c\})$. D'après la dernière égalité, $C \tilde{\vdash} L^c$ est un résolvant de A et $B \tilde{\vdash} L^c$. Donc par la propriété 2.1.14 page 44, $\Gamma \vdash C \tilde{\vdash} L^c$.
 - (c) Supposons $\Gamma \vdash A \tilde{\vdash} L^c$ et $\Gamma \vdash B$, en échangeant ci-dessus le rôle de A et B , nous obtenons $\Gamma \vdash C \tilde{\vdash} L^c$.
 - (d) Supposons $\Gamma \vdash A \tilde{\vdash} L^c$ et $\Gamma \vdash B \tilde{\vdash} L^c$, comme ci-dessus nous obtenons $\Gamma \vdash C \tilde{\vdash} L^c$.
 Par suite dans les quatre cas, nous avons $\Gamma \vdash C$ ou $\Gamma \vdash C \tilde{\vdash} L^c$.

□

Lemme 2.1.23 Soient Γ un ensemble de clauses, C une clause et L un littéral. Si $\Gamma[L := 0] \vdash C$ alors $\Gamma \vdash C$ ou $\Gamma \vdash C \tilde{\vdash} L$.

Preuve : Supposons $\Gamma[L := 0] \vdash C$. Puisque $\Gamma[L := 0] = \Gamma[L^c := 1]$ et que $L^{cc} = L$, d'après le lemme 2.1.22 page précédente nous avons $\Gamma \vdash C$ ou $\Gamma \vdash C \tilde{\vdash} L$. □

Théorème 2.1.24 Soit Γ un ensemble fini de clauses. Si Γ est insatisfaisable alors $\Gamma \vdash \perp$.

Preuve : Supposons que Γ est insatisfaisable. Soit n le nombre de variables de Γ . Nous montrons que $\Gamma \vdash \perp$ par récurrence sur n . Supposons que pour tout ensemble Δ de clauses insatisfaisable avec moins de n variables, nous avons $\Delta \vdash \perp$.

Cas de base : Supposons que n soit nul. Donc $\Gamma = \emptyset$ ou $\Gamma = \{\perp\}$. Le premier cas est impossible, car l'ensemble vide est valide (toute assignation en est modèle). Donc $\Gamma = \{\perp\}$ et par suite $\Gamma \vdash \perp$.

Induction : Supposons que n soit non nul. Soit x une variable figurant dans Γ . D'après la propriété 2.1.21 page ci-contre, $\Gamma[x := 0]$ et $\Gamma[x := 1]$ sont insatisfaisables. Puisque la variable x ne figure pas dans ces deux ensembles de clauses, l'hypothèse de récurrence s'applique, donc : $\Gamma[x := 0] \vdash \perp$ et $\Gamma[x := 1] \vdash \perp$. Des lemmes 2.1.22 page précédente et 2.1.23, nous déduisons soit $\Gamma \vdash \perp$, soit $\Gamma \vdash \bar{x}$ et $\Gamma \vdash x$. Dans le premier cas, la preuve est terminée. Dans le deuxième cas, puisque \perp est un résolvant de \bar{x} et x , nous avons également $\Gamma \vdash \perp$. □

Corollaire 2.1.25 Soit Γ un ensemble fini de clauses. Γ est insatisfaisable si et seulement si $\Gamma \vdash \perp$.

Preuve : Ce corollaire est une conséquence immédiate du lemme ci-dessus et de la remarque faite au début du paragraphe 2.1.3 page 45. □

2.1.4 Réduction d'un ensemble de clauses

Réduire un ensemble de clauses, c'est lui enlever les clauses valides et les clauses contenant une *autre* clause de l'ensemble. Un ensemble de clauses est réduit s'il n'est plus réductible.

Définition 2.1.26 (Ensemble de clauses réduit) Un ensemble de clauses est réduit s'il ne contient aucune clause valide et aucune clause n'est incluse dans une autre clause.

Exemple 2.1.27 La réduction de l'ensemble de clauses $\{p + q + \bar{p}, p + r, p + r + \bar{s}, r + q\}$ donne l'ensemble réduit :

$$\{p + r, r + q\}.$$

Propriété 2.1.28 Un ensemble de clauses E est équivalent à l'ensemble de clauses réduit obtenu à partir de E .

Preuve :

Enlever une clause valide donc valant 1, c'est appliquer l'identité $x.1 = x$. Enlever une clause incluant une autre clause, c'est appliquer l'identité $x.(x + y) = x$. □

2.1.5 Clauses minimales pour la déduction

Définition 2.1.29 (Clause minimale pour la déduction) Soit Γ un ensemble de clauses. Une clause minimale pour la déduction de Γ est une clause non valide déduite de Γ et ne contenant strictement aucune clause déduite de Γ .

Exemple 2.1.30 Considérons l'ensemble de clauses $\Gamma = \{a + \bar{b}, b + c + d\}$ la clause $a + c + d$ est minimale pour la déduction. Par contre si on rajoute la clause $\bar{a} + c$ à Γ alors $a + c + d$ n'est pas minimale car nous pouvons déduire $c + d$ qui est inclus dans la clause $a + c + d$.

L'algorithme de la stratégie complète, présenté dans le paragraphe suivant, construit, à partir d'un ensemble Γ de clauses, l'ensemble des clauses minimales pour la déduction de Γ .

Propriété 2.1.31 Soit Θ l'ensemble des clauses minimales pour la déduction de l'ensemble de clauses Γ . L'ensemble Γ est insatisfaisable si et seulement si $\perp \in \Theta$.

Preuve :

- Supposons $\perp \in \Theta$, alors $\Gamma \vdash \perp$, donc par cohérence de la résolution, Γ est insatisfaisable.
- Supposons Γ insatisfaisable, par complétude de la résolution, $\Gamma \vdash \perp$. Par suite \perp est minimal pour la déduction de Γ , donc $\perp \in \Theta$.

□

2.1.6 Clauses minimales pour la conséquence

Définition 2.1.32 (Clause minimale pour la conséquence) Soit Γ un ensemble de clauses. Une clause minimale pour la conséquence de Γ est une clause non valide conséquence de Γ et ne contenant strictement aucune clause conséquence de Γ .

Exemple 2.1.33 Considérons l'ensemble de clauses $\Gamma = \{a + d, \bar{a} + b, \bar{b} + c\}$. La clause $d + c$ est minimale pour la conséquence de Γ .

Conséquence : $d + c$ est une conséquence de Γ car dans tout modèle de Γ soit d est vrai soit c est vrai. Pour s'en convaincre essayons de construire un modèle où d et c sont faux. Dans ce cas pour satisfaire la première et la dernière clause de Γ , il faut que $a = 1$ et $b = 0$ dans le modèle ce qui rend faux la seconde clause de Γ .

Minimalité : Il existe des modèles de Γ qui ne sont pas modèles de d (respectivement c) : $a = 1, d = 0, c = 1$ et $b = 1$ (respectivement $a = 0, d = 1, c = 0$ et $b = 0$).

Nous montrons ci-dessous, qu'une clause est minimale pour la déduction d'un ensemble de clauses si et seulement si elle est minimale pour la conséquence de ce même ensemble de clauses.

Propriété 2.1.34 Soit Γ un ensemble de clauses. Pour toute clause C non valide, conséquence de Γ , il existe une sous-clause de C déduite de Γ .

Preuve : Soient C une clause non valide, conséquence de Γ et $\Gamma' = \Gamma \cup \{L^c \mid L \in C\}$. Une assignation v est modèle de Γ' si et seulement si v est modèle de Γ et n'est pas modèle de C , donc, puisque C est conséquence de Γ , Γ' est insatisfaisable. De plus, par complétude réfutationnelle, nous obtenons $\Gamma' \vdash \perp$.

Nous montrons la propriété (P) suivante : pour toute clause D déduite de Γ' :

- Soit D est le complémentaire d'un littéral de C .
- Soit il existe une clause E telle que $E \subset C$ et $D \cup E$ est déductible de Γ .

En appliquant la propriété (P) à la clause vide qui est déduite de Γ' , il en résulte qu'il existe une sous-clause de C déduite de Γ .

La preuve de la propriété (P) est une preuve par induction sur la longueur n de la preuve par résolution de D . Le cas de base pour $n = 0$ est immédiat, car $D \in \Gamma$. Supposons que pour toute preuve de longueur inférieure à n la propriété (P) soit vraie. Nous distinguons deux cas :

1. Supposons $D \in \Gamma'$, donc soit D est le complémentaire d'un littéral de C , soit $D \in \Gamma$ et par suite D est déduite de Γ , donc la propriété (P) est vraie.
2. Supposons $D \notin \Gamma'$. Donc D est le résolvant de deux clauses E et F ayant une preuve de longueur inférieure à n . Par suite, il existe un littéral L avec $L \in E, L^c \in F$ et $D = (E - \{L\}) \cup (F - \{L^c\})$. Puisque C n'est pas valide, la résolution ne peut pas être appliquée à deux clauses qui sont des littéraux complémentaires de ceux de C . Donc au moins l'une des clauses E, F n'est pas un littéral complémentaire d'un littéral de C . Supposons que cette clause non complémentaire d'un littéral de C soit la clause E . Par hypothèse de récurrence, il existe une clause G telle que $G \subset C$ et $E \cup G$ est déductible de Γ . Nous distinguons deux cas, suivant que F soit ou ne soit pas le complémentaire d'un littéral de C .
 - (a) F est le complémentaire d'un littéral de C . Dans ce cas, F est le littéral L^c et $D = E - \{L\}$. De plus, comme $L \in E$, nous avons $L \notin D$ et $E = D \cup \{L\}$. Ensuite, puisque $G \subset C$, nous avons $(G - \{L\}) \subset C$. Puisque $E \cup G = D \cup \{L\} \cup G$ et $F = \{L^c\}$ sont déductibles de Γ (par hypothèse de récurrence) et $D \cup (G - \{L\})$ est un résolvant de $E \cup G$ et F , nous pouvons conclure que $D \cup (G - \{L\})$ est déductible de Γ . Ainsi, la propriété est vérifiée pour D .
 - (b) F n'est pas le complémentaire d'un littéral de C . Par hypothèse de récurrence, il existe une clause H telle que $H \subset C$ et $F \cup H$ est déductible de Γ . Puisque $G \subset C$ et $H \subset C$, nous avons $G \cup H \subset C$. Puisque $E \cup G$ ainsi que $F \cup H$ sont déductibles de Γ et que D est un résolvant de E et F , alors $D \cup (G \cup H)$ est un résolvant de $E \cup G$ et de $F \cup H$ et par suite est aussi déductible de Γ . Donc la propriété (P) est vérifiée pour D .

□

Remarquons que cette propriété généralise la propriété de la complétude réfutationnelle, si la clause vide est conséquence d'un ensemble de clauses, alors elle en est déduite.

Théorème 2.1.35 Soit Γ un ensemble de clauses. Une clause est minimale pour la déduction de Γ si et seulement si elle est minimale pour la conséquence de Γ .

Preuve : Nous effectuons une preuve par double implication.

- \Rightarrow Soit C une clause minimale pour la déduction de Γ . Par cohérence de la résolution, $\Gamma \models C$. Nous supposons que C ne soit pas minimale pour la conséquence et cherchons une contradiction. Dans ce cas, il existerait une clause D , sous-clause stricte de C et conséquence de Γ . Puisque C n'est pas valide, D aussi n'est pas valide, d'après la propriété 2.1.34, il existerait E sous-clause de D déduite de Γ . Donc, contrairement à l'hypothèse, C ne serait pas minimale pour la déduction. Par suite, C est minimale pour la conséquence.
- \Leftarrow Soit C minimale pour la conséquence de Γ . Puisque C n'est pas valide, d'après la propriété 2.1.34, il existe D , une sous-clause de C déduite de Γ . Par cohérence de la résolution, D est conséquence de Γ . Et puisque C est minimale, $D = C$, donc C est déduite de Γ . Nous supposons que C ne soit pas minimale pour la déduction. Il existerait D , sous-clause stricte de C , déduite de Γ . Par cohérence de la résolution, D serait conséquence de Γ . Donc, contrairement à l'hypothèse, C ne serait pas minimale pour la conséquence. Par suite, C est minimale pour la déduction.

□

2.2 Stratégie complète

Pour savoir si un ensemble fini Γ de clauses est contradictoire, il y a une méthode simple mais peu efficace : construire toutes les clauses que nous pouvons déduire de Γ . D'après le corollaire 2.1.25 page 47, Γ est contradictoire si et seulement si la clause vide a été construite. Il faut préciser ce que nous entendons par construire toutes les clauses déduites de Γ . Si nous considérons comme différentes les clauses qui s'écrivent différemment, deux clauses ont une infinité de résolvents. Par exemple les clauses $p + \bar{q}$ et $r + q$ ont comme résolvents la clause $p + r$ mais aussi les clauses $r + p$, $p + r + p$, $p + \dots + p + r + \dots + r$. Donc l'ensemble des clauses déduites de Γ est infini. Aussi, il faut, comme précédemment, considérer comme égales deux clauses, qui ont le même ensemble de littéraux (voir définition 2.1.1 page 42) : avec cette définition de l'égalité de deux clauses, l'ensemble des résolvents de deux clauses est fini ainsi que l'ensemble des clauses déduites de Γ . Supposons que l'ensemble des littéraux de Γ ait n éléments, comme les clauses déduites de Γ ne comportent que des littéraux de Γ (voir exercice 40 page 63), il y a donc au plus 2^n clauses déduites de Γ . Pour simplifier la suite, comme ci-dessus, nous confondons une clause et son ensemble de littéraux.

2.2.1 Algorithme de la stratégie complète

Soit Γ un ensemble de clauses. Pour obtenir l'ensemble des clauses minimales déduites de Γ , on construit deux suites d'ensembles de clauses $\Delta_{i(i \geq 0)}$ et $\Theta_{i(i \geq 0)}$ en s'arrêtant au premier indice k tel que $\Delta_k = \emptyset$.

1. Δ_0 est obtenu en réduisant Γ et Θ_0 est l'ensemble vide.
2. Δ_{i+1} est obtenu de la façon suivante
 - (a) Nous construisons l'ensemble des résolvents entre les clauses de Δ_i et celles de $\Delta_i \cup \Theta_i$ puis nous réduisons cet ensemble.
 - (b) Nous enlevons de cet ensemble de résolvents, les clauses qui contiennent une clause de $\Delta_i \cup \Theta_i$.
3. Θ_{i+1} est obtenu en enlevant de l'ensemble $\Delta_i \cup \Theta_i$ les clauses qui contiennent une clause de Δ_{i+1} .

Nous illustrons notre algorithme par deux exemples.

Exemple 2.2.1 Nous appliquons l'algorithme à l'ensemble de clauses $a + b + \bar{a}, a + b, a + b + c, a + \bar{b}, \bar{a} + b, \bar{a} + \bar{b}$. L'algorithme effectue alors le calcul suivant. L'ensemble Δ_0 comprend les clauses :

$$1 \quad a + b$$

$$2 \quad a + \bar{b}$$

$$3 \quad \bar{a} + b$$

$$4 \quad \bar{a} + \bar{b}$$

L'ensemble Θ_0 est vide. L'algorithme calcule les résolvants non valides suivants :

$$5 \quad a \quad \text{résolvant de 1 et 2}$$

$$6 \quad b \quad \text{résolvant de 1 et 3}$$

$$7 \quad \bar{b} \quad \text{résolvant de 2 et 4}$$

$$8 \quad \bar{a} \quad \text{résolvant de 3 et 4}$$

Donc l'ensemble Δ_1 comprend les mêmes clauses que nous numérotions de 5 à 8.

L'ensemble Θ_1 est

vide.

L'ensemble Δ_2 comprend la clause :

$$9 \quad \perp \quad \text{de 5 et 8}$$

Car l'algorithme obtient deux fois \perp .

L'ensemble Θ_2 est

vide.

L'ensemble Δ_3 est

vide.

L'ensemble Θ_3 est égal à

Δ_2 .

Nous notons que l'algorithme a construit la preuve suivante :

$1 \quad a + b$
 $2 \quad a + \bar{b}$
 $5 \quad a \quad \text{résolvant de 1 et 2}$
 $3 \quad \bar{a} + b$
 $4 \quad \bar{a} + \bar{b}$
 $8 \quad \bar{a} \quad \text{résolvant de 3 et 4}$
 $9 \quad \perp \quad \text{résolvant de 5 et 8}$

Nous pouvons utiliser la représentation ci-dessous sous forme de tableau représentant la construction de ces ensembles.

k	Δ_k	Θ_k	$\Delta_k \cup \Theta_k$	Résolvants de Δ_k et $\Delta_k \cup \Theta_k$
0	$a + b, a + \bar{b}, \bar{a} + b, \bar{a} + \bar{b}$	\emptyset	$a + b, a + \bar{b}, \bar{a} + b, \bar{a} + \bar{b}$	$b, b + \bar{b}, a, a + \bar{a}, \bar{a}, \bar{b}$
1	a, b, \bar{b}, \bar{a}	\emptyset	a, b, \bar{b}, \bar{a}	\perp
2	\perp	\emptyset	\perp	\emptyset
3	\emptyset	\perp		

Comme le montre cet exemple, les clauses de Δ_i ont des preuves de hauteur i , et celles de Θ_i des preuves de hauteur inférieure à i .

Exemple 2.2.2 L'application de la stratégie complète sur l'ensemble $\{a, c, \bar{a} + \bar{b}, \bar{c} + e\}$ nous donne l'ensemble $\{a, c, e, \bar{b}\}$.

k	Δ_k	Θ_k	$\Delta_k \cup \Theta_k$	Résolvants de Δ_k et $\Delta_k \cup \Theta_k$
0	$a, c, \bar{a} + \bar{b}, \bar{c} + e$	\emptyset	$a, c, \bar{a} + \bar{b}, \bar{c} + e$	e, \bar{b}
1	e, \bar{b}	a, c	a, c, e, \bar{b}	\emptyset
2	\emptyset	a, c, e, \bar{b}		

Nous montrons que l'algorithme se termine et qu'il est correct, c'est-à-dire que l'ensemble Θ_k est l'ensemble des clauses minimales déduites de Γ . D'après la propriété 2.1.31 page 48, Γ est insatisfaisable si et seulement si la clause vide est élément de Θ_k .

2.2.2 Arrêt de l'algorithme

Soit n le nombre de littéraux de Γ . Les clauses éléments des Δ_i sont des clauses déduites de Γ , donc, comme nous l'avons rappelé ci-dessus, elles ne comportent que des littéraux de Γ . Par suite, l'ensemble $\bigcup_{i < k} \Delta_i$ comprend au plus 2^n clauses. D'après l'algorithme, les ensembles Δ_i sont non vides pour $i < k$. Nous montrons ci-dessous que ces ensembles sont disjoints. Par conséquent $k \leq 2^n + 1$, autrement dit l'algorithme s'arrête.

Propriété 2.2.3 Soit $i \leq k$. Toute clause de $\bigcup_{j \leq i} \Delta_j$ contient une clause de $\Delta_i \cup \Theta_i$.

Preuve : Nous effectuons une preuve par induction.

- Pour $i = 0$ la propriété est triviale car $\bigcup_{j \leq 0} \Delta_j = \Delta_0$ et $\Delta_i \cup \Theta_i = \Delta_0$ ($\Theta_0 = \emptyset$).
- Supposons la propriété vraie au rang i et montrons qu'elle reste vraie au rang $i + 1$. Soit $C \in \bigcup_{j \leq i+1} \Delta_j$. Montrons que C contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$. Nous examinons tous les cas possibles pour C .
 1. $C \in \Delta_{i+1}$. Donc C contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$.
 2. $C \in \bigcup_{j \leq i} \Delta_j$. Par hypothèse de récurrence, C contient une clause $D \in \Delta_i \cup \Theta_i$. Nous distinguons deux situations pour D .
 - (a) $D \in \Theta_{i+1}$. Donc C contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$.
 - (b) $D \notin \Theta_{i+1}$. Par construction de Θ_{i+1} , puisque $D \in \Delta_i \cup \Theta_i$ et que $D \notin \Theta_{i+1}$, c'est que D contient une clause de Δ_{i+1} . Or C contient D , donc C contient aussi une clause de $\Delta_{i+1} \cup \Theta_{i+1}$.

□

Propriété 2.2.4 Pour tout $i \leq k$, les ensembles Δ_i sont disjoints entre eux.

Preuve : Nous effectuons une récurrence sur les ensembles Δ_j avec $0 \leq j \leq i$ et $i \leq k$.

Cas de base : Si $i = 0$, il n'y a qu'un seul ensemble, donc la propriété est vérifiée.

Récurrence : Soit $i < k$. Supposons que tous les ensembles Δ_j où $j \leq i$ sont disjoints entre eux.

Montrons que Δ_{i+1} est disjoint de ces ensembles.

Soit $C \in \Delta_{i+1}$. Supposons, au contraire, que $C \in \bigcup_{j \leq i} \Delta_j$. D'après la propriété précédente, C inclut une clause de $\Delta_i \cup \Theta_i$. Donc par construction de Δ_{i+1} , $C \notin \Delta_{i+1}$, contradiction. Par suite, $C \notin \bigcup_{j \leq i} \Delta_j$.

□

2.2.3 Le résultat de l'algorithme est équivalent à l'ensemble initial de clauses

Nous montrons ci-dessous que l'ensemble Θ_k est équivalent à Γ .

Propriété 2.2.5 Pour tout $i < k$, les ensembles $\Delta_i \cup \Theta_i$ et $\Delta_{i+1} \cup \Theta_{i+1}$ sont équivalents.

Preuve :

1. Toute clause de $\Delta_{i+1} \cup \Theta_{i+1}$ est conséquence de $\Delta_i \cup \Theta_i$. En effet toute clause de $\Delta_{i+1} \cup \Theta_{i+1}$ est élément de $\Delta_i \cup \Theta_i$ ou résolvant de deux éléments de cet ensemble, donc est conséquence de cet ensemble.
2. Toute clause de $\Delta_i \cup \Theta_i$ est conséquence de $\Delta_{i+1} \cup \Theta_{i+1}$. Soit $C \in \Delta_i \cup \Theta_i$. Nous distinguons deux cas possibles :
 - (a) $C \in \Theta_{i+1}$, ainsi C est conséquence de $\Delta_{i+1} \cup \Theta_{i+1}$.
 - (b) $C \notin \Theta_{i+1}$, ainsi C contient une clause de Δ_{i+1} donc est conséquence de $\Delta_{i+1} \cup \Theta_{i+1}$.

□

Propriété 2.2.6 Les ensembles Γ et Θ_k sont équivalents.

Preuve : Par définition Δ_0 est l'ensemble obtenu par réduction de Γ , d'après la propriété 2.1.28 page 48, ces deux ensembles sont équivalents. Puisque Θ_0 est vide, Γ est équivalent à $\Delta_0 \cup \Theta_0$. D'après la propriété 2.2.5 et par récurrence, $\Delta_0 \cup \Theta_0$ est équivalent à l'ensemble de clauses $\Delta_k \cup \Theta_k$. Puisque l'algorithme termine quand Δ_k est l'ensemble vide, les ensembles Γ et Θ_k sont équivalents.

□

2.2.4 Le résultat de l'algorithme est l'ensemble des clauses minimales pour la déduction de l'ensemble initial de clauses

Nous montrons que l'ensemble Θ_k est l'ensemble des clauses minimales pour la déduction de Γ .

Propriété 2.2.7 Toute clause de Δ_i (où $i \leq k$) a une preuve sans tautologie de hauteur i à partir de Γ . Toute clause de Θ_i (où $i \leq k$) a une preuve sans tautologie de hauteur inférieure à i à partir de Γ .

Preuve : Nous effectuons une preuve par récurrence sur i .

- Pour $i = 0$, c'est évident.
- Admettons cette propriété au rang i et montrons qu'elle reste vraie au rang $i + 1$. Soit $C \in \Delta_{i+1}$, par construction C est un résolvant entre une clause $D \in \Delta_i$ et une clause $E \in \Delta_i \cup \Theta_i$ et C n'est pas une tautologie. Par hypothèse de récurrence, D a une preuve, sans tautologie, de hauteur i , à partir de Γ , et E a une preuve, sans tautologie, de hauteur au plus i , à partir de Γ . Donc C a une preuve, sans tautologie, de hauteur $i + 1$, à partir de Γ . Soit $C \in \Theta_{i+1}$, par construction $C \in \Delta_i \cup \Theta_i$, donc par hypothèse de récurrence, C a une preuve sans tautologie à partir de Γ de hauteur au plus i .

□

Propriété 2.2.8 Pour tout $i \leq k$, l'ensemble $\Delta_i \cup \Theta_i$ est réduit.

Preuve : Nous faisons un raisonnement par récurrence.

- Pour $i = 0$, c'est évident car l'ensemble Δ_0 est obtenu par réduction de Γ et que $\Theta_0 = \emptyset$.
- Supposons que l'ensemble $\Delta_i \cup \Theta_i$ soit réduit. Montrons qu'il en est de même de l'ensemble $\Delta_{i+1} \cup \Theta_{i+1}$. Par construction, l'ensemble Δ_{i+1} est réduit. Par construction, l'ensemble Θ_{i+1} est un sous-ensemble de $\Delta_i \cup \Theta_i$, donc par hypothèse de récurrence, il est réduit. Supposons que l'ensemble $\Delta_{i+1} \cup \Theta_{i+1}$ ne soit pas réduit, il n'y a que deux possibilités :
 1. $C \in \Delta_{i+1}$ contient une clause $D \in \Theta_{i+1}$: par construction de Θ_{i+1} , $D \in \Delta_i \cup \Theta_i$, ce qui implique, par construction $C \notin \Delta_{i+1}$. Nous avons une contradiction.
 2. $C \in \Theta_{i+1}$ contient une clause $D \in \Delta_{i+1}$: par construction de Θ_{i+1} , ce cas est aussi impossible.

Par suite, l'ensemble $\Delta_{i+1} \cup \Theta_{i+1}$ est réduit.

□

Propriété 2.2.9 Pour tout $i \leq k$, $\Theta_i \subset \bigcup_{j < i} \Delta_j$.

Preuve : Nous montrons le résultat par induction sur i .

- Pour $i = 0$, la propriété est vraie car Θ_0 est vide.
- Supposons que pour $i < k$, nous avons : $\Theta_i \subset \bigcup_{j < i} \Delta_j$. Par construction, Θ_{i+1} est un sous-ensemble de $\Delta_i \cup \Theta_i$. En utilisant l'hypothèse de récurrence, c'est donc un sous-ensemble de $\bigcup_{j < i+1} \Delta_j$.

□

Propriété 2.2.10 Pour tout $i \leq k$, un résolvant non valide de deux clauses de Θ_i contient une clause de $\Delta_i \cup \Theta_i$.

Preuve : Par induction sur i nous avons :

- Pour $i = 0$, la propriété est vraie car Θ_0 est vide.
- Supposons la propriété vérifiée pour $i < k$. Soient $C, D \in \Theta_{i+1}$ ayant un résolvant E non valide. Montrons que E contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$. D'après la propriété 2.2.9 page précédente, $C, D \in \bigcup_{j < i+1} \Delta_j$. Nous distinguons deux cas possibles :
 1. Supposons $C \in \Delta_i$ ou $D \in \Delta_i$. Nous avons deux cas :
 - (a) $E \in \Delta_{i+1}$, donc E contient un élément de $\Delta_{i+1} \cup \Theta_{i+1}$
 - (b) $E \notin \Delta_{i+1}$, donc, par construction, soit E contient une autre clause de Δ_{i+1} donc de $\Delta_{i+1} \cup \Theta_{i+1}$, soit E contient une clause $F \in \Delta_i \cup \Theta_i$. Nous distinguons deux cas pour F :
 - i. $F \in \Theta_{i+1}$ donc E contient $F \in \Delta_{i+1} \cup \Theta_{i+1}$.
 - ii. $F \notin \Theta_{i+1}$ donc, par construction de cet ensemble, F inclut une clause $G \in \Delta_{i+1}$. Donc E contient $G \in \Delta_{i+1} \cup \Theta_{i+1}$.

Ainsi dans ces deux cas, E contient une clause élément de $\Delta_{i+1} \cup \Theta_{i+1}$.

2. Supposons $C \notin \Delta_i$ et $D \notin \Delta_i$. Puisque les clauses de Θ_{i+1} sont des clauses de $\Delta_i \cup \Theta_i$, nous avons donc $C, D \in \Theta_i$. Par hypothèse de récurrence, E contient une clause $F \in \Delta_i \cup \Theta_i$. Comme dans le cas précédent, nous en déduisons que E contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$.

Par suite, dans ces deux cas, E contient une clause de $\Delta_{i+1} \cup \Theta_{i+1}$.

□

Propriété 2.2.11 Si la clause C contient une clause de $\Delta_i \cup \Theta_i$ alors, pour tout j tel que $i \leq j \leq k$, C contient une clause de $\Delta_j \cup \Theta_j$.

Preuve : La preuve est une récurrence élémentaire sur j .

- Pour $j = i$, la propriété est triviale.
 - Supposons que C contient une clause $D \in \Delta_j \cup \Theta_j$ où $i \leq j < k$. Montrons que C contient une clause de $\Delta_{j+1} \cup \Theta_{j+1}$. Nous distinguons deux cas pour D .
 1. $D \in \Theta_{j+1}$: donc C contient une clause de $\Delta_{j+1} \cup \Theta_{j+1}$.
 2. $D \notin \Theta_{j+1}$. Par construction de Θ_{j+1} , D contient E où $E \in \Delta_{j+1}$. Donc C contient une clause de $\Delta_{j+1} \cup \Theta_{j+1}$.
- Ainsi dans les deux cas, C contient une clause de $\Delta_{j+1} \cup \Theta_{j+1}$. □

Propriété 2.2.12 *Toute clause non valide déduite de Γ contient un élément de Θ_k .*

Preuve : Supposons que toute clause non valide ayant une preuve de longueur inférieure à n à partir de Γ contienne un élément de Θ_k . Soit C une clause non valide déduite de Γ par une preuve de longueur n . Montrons que C contient une clause de Θ_k . Soit $C \in \Gamma$, soit C est résolvant de deux clauses D et E précédant C dans la preuve de C . Examinons ces deux cas.

1. $C \in \Gamma$. Puisque C n'est pas valide et que Δ_0 est obtenu par réduction de Γ , C contient une clause de Δ_0 . Puisque Θ_0 est vide, C contient une clause de $\Delta_0 \cup \Theta_0$. D'après la propriété 2.2.11, C contient une clause de $\Delta_k \cup \Theta_k$ donc de Θ_k , puisque Δ_k est vide.
2. C est résolvant de D et E , clauses déduites de Γ par des preuves de longueurs inférieures à n . D'après la propriété 2.1.8 page 43, si D et E étaient valides, il en serait de même de C , donc l'une des clauses D, E n'est pas valide.
 - (a) Supposons E valide. Donc D n'est pas valide et, par hypothèse de récurrence, contient $D' \in \Theta_k$. Puisque, d'après cette même propriété 2.1.8 page 43, C contient D , C contient un élément de Θ_k .
 - (b) Supposons D valide. De façon analogue, C contient un élément de Θ_k .
 - (c) Supposons que ni D , ni E ne soient valides. Par hypothèse de récurrence, D contient $D' \in \Theta_k$ et E contient $E' \in \Theta_k$. Puisque C est résolvant de D et E , il existe un littéral L tel que $L \in D, L^c \in E$ et $C = (D - \{L\}) \cup (E - \{L^c\})$. Examinons les deux cas exhaustifs suivants :
 - i. Supposons $L \notin D'$ ou $L^c \notin E'$. Nous avons $D' \subset C$ ou $E' \subset C$. Par suite, C contient un élément de Θ_k .
 - ii. Supposons $L \in D'$ et $L^c \in E'$. Soit $C' = (D' - \{L\}) \cup (E' - \{L^c\})$. C' est un résolvant de D' et E' et C contient C' . D'après la propriété 2.2.10 page ci-contre, puisque D' et E' sont éléments de Θ_k , C' est élément de $\Delta_k \cup \Theta_k$, donc de Θ_k , puisque Δ_k est vide. Donc C contient un élément de Θ_k .

Ainsi dans tous les cas, C contient un élément de Θ_k . □

Propriété 2.2.13 Θ_k est l'ensemble des clauses minimales pour la déduction de Γ .

Preuve :

1. Soit C une clause minimale pour la déduction de Γ . Montrons que $C \in \Theta_k$. Puisque C est une clause non valide déduite de Γ , d'après la propriété 2.2.12 page précédente, C contient une clause $D \in \Theta_k$. D'après la propriété 2.2.7 page 53, la clause D est aussi déduite de Γ et puisque C est minimale pour la déduction de Γ et que C contient D , nous avons $C = D$, donc $C \in \Theta_k$.
2. Soit $C \in \Theta_k$. Montrons que C est minimale pour la déduction de Γ . Supposons, au contraire, que C contienne strictement une clause D déduite de Γ . D'après la propriété 2.2.12 page précédente, D contient une clause de Θ_k , donc $C \in \Theta_k$ contient strictement une clause de Θ_k . Par suite, l'ensemble Θ_k ne serait pas réduit, ce qui contredit la propriété 2.2.8 page 53. En effet d'après cette propriété, l'ensemble $\Delta_k \cup \Theta_k$ est réduit et, puisque Δ_k est vide, l'ensemble Θ_k est réduit. □

2.3 Algorithme de Davis-Putnam-Logemann-Loveland

La méthode de Davis-Putnam-Logemann-Loveland (DPLL) fut créée en 1960 par Martin Davis et Hillary Putnam, puis améliorée en 1962 par Martin Davis, George Logemann et Donald Loveland. L'algorithme DPLL permet de décider si un ensemble de clauses est satisfaisable. Bien que cet algorithme date de plus de 50 ans, de nombreuses variantes en sont encore étudiées, car il est bien meilleur que ceux précédemment étudiés (table de vérité, transformation en somme de monômes ou produit de sommes, stratégie complète). Nous remarquons d'abord qu'il y a deux types de transformations de formules :

1. Celles qui préservent le sens, c'est-à-dire transforment une formule en une autre formule équivalente.
2. Celles qui préservent seulement la satisfaisabilité, c'est-à-dire transforment une formule satisfaisable en une autre formule satisfaisable.

L'efficacité de l'algorithme DPLL vient de l'utilisation des transformations préservant seulement la satisfaisabilité, car ces transformations sont moins coûteuses que celles préservant le sens. Nous introduisons d'abord la suppression des clauses contenant des littéraux isolés, puis la résolution unitaire et enfin la simplification des clauses valides. Ces transformations constituent les briques de base de l'algorithme DPLL. L'idée principale de cet algorithme, partant d'un ensemble de clauses « irréductibles », est de choisir une variable et de considérer les deux situations possibles : le cas où elle est vraie et le cas où elle est fausse. Ainsi les deux ensembles de clauses obtenus peuvent éventuellement se simplifier, sinon le choix d'une nouvelle variable est nécessaire. Dans le pire des cas il s'agit de construire la table de vérité correspondante aux clauses, mais en pratique de nombreuses simplifications apparaissent et permettent à cet algorithme de conclure efficacement.

2.3.1 Suppression des clauses contenant des littéraux isolés

Comme nous ne cherchons pas à préserver l'équivalence de formule mais la satisfaisabilité, nous avons la liberté de choisir la valeur de certaines variables. En particulier une variable qui n'apparaît que sous la forme d'un littéral positif ou négatif sera dite *isolée*. Nous pouvons alors affecter la valeur adéquate à cette variable afin de rendre vraies toutes les occurrences du littéral associé, ceci sans changer la satisfaisabilité de l'ensemble de clauses considéré.

Définition 2.3.1 (Littéral isolé) Soient Γ un ensemble de clauses et L un littéral élément d'une clause de Γ , L est un littéral isolé (relativement à Γ), si aucune clause de Γ ne comporte de littéral complémentaire de L .

Lemme 2.3.2 La suppression des clauses qui comportent un littéral isolé, préserve la satisfaisabilité.

La preuve est demandée dans l'exercice 47 page 64.

Exemple 2.3.3 Soit Γ l'ensemble de clauses $\{p + q + r, \bar{q} + \bar{r}, q + s, \bar{s} + t\}$.

Les littéraux p et t sont isolés. Donc nous pouvons supprimer les clauses $p + q + r$ et $\bar{s} + t$. Après cette suppression, nous obtenons l'ensemble $\Theta = \{\bar{q} + \bar{r}, q + s\}$. Les littéraux \bar{r} et s sont à leur tour isolés pour l'ensemble Θ . En enlevant les clauses qui contiennent ces littéraux, nous obtenons l'ensemble vide, dont toute assignation est modèle. Le lemme 2.3.2 page précédente nous permet de dire que Γ a un modèle (en prenant par exemple $p = 1, t = 1, s = 1$ et $r = 0$). Notons que toute assignation est modèle de l'ensemble vide alors qu'il existe des contre-modèles à Γ (par exemple, $p = 0, q = 0$ et $r = 0$), ceci montre que la suppression des clauses avec un littéral isolé ne maintient pas le sens.

2.3.2 Résolution unitaire

Définition 2.3.4 Une clause unitaire est une clause qui ne comporte qu'un littéral.

Les clauses unitaires sont aussi des clauses particulières qui sont :

- soit isolées et donc seront traitées par la simplification précédente,
- soit de potentiels résolvants, dans ce cas il faut effectuer la résolution dite « unitaire » de ces clauses.

Lemme 2.3.5 Soit Γ un ensemble de clauses. Soit L l'ensemble des littéraux des clauses unitaires de Γ . Soit Θ l'ensemble de clauses ainsi obtenu à partir de Γ :

- Si L comporte deux littéraux complémentaires, alors $\Theta = \{\perp\}$.
- Sinon Θ est obtenu ainsi :
 - Nous enlevons les clauses qui comportent un élément de L .
 - À l'intérieur des clauses restantes nous enlevons les littéraux complémentaires des éléments de L .

Γ a un modèle si et seulement si Θ en a un.

La preuve est demandée dans l'exercice 48 page 64.

Exemple 2.3.6

- Soit Γ l'ensemble de clauses : $p + q, \bar{p}, \bar{q}$.

Par la résolution unitaire, nous obtenons la clause vide, donc Γ n'a pas de modèle.

- Soit Γ l'ensemble de clauses : $a + b + \bar{d}, \bar{a} + c + \bar{d}, \bar{b}, d, \bar{c}$.

Par une première résolution unitaire, nous obtenons : a, \bar{a} .

Par une deuxième résolution unitaire, nous obtenons la clause vide, donc Γ n'a pas de modèle.

- Soit Γ l'ensemble de clauses : $p, q, p + r, \bar{p} + r, q + \bar{r}, \bar{q} + s$.

Par résolution unitaire, nous obtenons : r, s . Cet ensemble de clauses a un modèle donc Γ en a un. Notons aussi que l'ensemble de clauses $\{r, s\}$ n'est pas équivalent à Γ , ce qui montre que la résolution unitaire ne maintient pas le sens des formules.

2.3.3 Suppression des clauses valides

Enfin la dernière transformation utilisée dans l'algorithme DPLL est de simplement enlever les clauses valides d'un ensemble de clauses. Ceci peut paraître une évidence mais cette transformation est une des idées centrales de cet algorithme, une fois une variable affectée à une valeur, elle est implicitement utilisée dans la suppression de clauses contenant des littéraux isolés.

Lemme 2.3.7 Soit Γ un ensemble de clauses. Soit Θ l'ensemble de clauses obtenu en supprimant les clauses valides de Γ . Γ a un modèle si et seulement si Θ en a un.

Preuve :

- Supposons que Γ a un modèle v , comme Θ est un sous-ensemble des clauses de Γ , v est aussi modèle de Θ . Donc, Θ a un modèle.
- Supposons que Θ a un modèle v . Soit v' une assignation de Γ telle que $v'(x) = v(x)$ pour toute variable x présente à la fois dans Γ et Θ . Soit C une clause de Γ . Si C est aussi une clause de Θ , alors v' est un modèle de C car v et v' donnent la même valeur à C . Si C n'est pas une clause de Θ , alors C est valide, en conséquence toute assignation, v' en particulier, est modèle de C . D'où, Γ a un modèle : v' .

□

2.3.4 L'algorithme

Nous donnons une version de l'algorithme `DPLL` dans la figure 2.1 page suivante dans la fonction `Algo_DPLL`. Nous pouvons constater dans l'algorithme que nous supprimons une seule fois les clauses valides avant de réellement commencer les simplifications. Cela se justifie par le fait que l'algorithme ne pourra pas produire de nouvelles clauses valides. En effet, il ne fait qu'enlever des littéraux des clauses initialement données. Par suite, il est inutile d'enlever des clauses valides lors de l'étape de réduction, puisqu'il n'y aura plus de telles clauses. La fonction `DPLL` teste d'abord si la clause vide a été générée, puis effectue les simplifications possibles avant de choisir une variable pour appliquer récursivement la même fonction dans le cas où cette variable est affectée à vraie et dans le cas où elle est affectée à faux.

```

bool fonction Algo_DPLL(  $\Gamma$  : ensemble de clauses)
0  Supprimer les clauses valides de  $\Gamma$ .
   Si  $\Gamma = \emptyset$ , retourner (vrai).
   Sinon retourner (DPLL( $\Gamma$ ))

bool fonction DPLL(  $\Gamma$  : ensemble de clauses non valides)
La fonction retourne vrai si et seulement si  $\Gamma$  est satisfaisable
1  Si  $\perp \in \Gamma$ , retourner(faux).
   Si  $\Gamma = \emptyset$ , retourner (vrai).
2  Réduire  $\Gamma$  : il suffit d'enlever toute clause contenant une autre clause.
3  Enlever de  $\Gamma$  les clauses comportant des littéraux isolés (cf. paragraphe 2.3.1 page 55).
   Si l'ensemble  $\Gamma$  a été modifié, aller en 1.
4  Appliquer à  $\Gamma$  la résolution unitaire (cf paragraphe 2.3.2 page 56).
   Si l'ensemble  $\Gamma$  a été modifié, aller en 1.
5  Choisir  $x$  une variable quelconque de  $\Gamma$ 
   retourner (DPLL( $\Gamma[x := 0]$ ) ou alors DPLL( $\Gamma[x := 1]$ ))

```

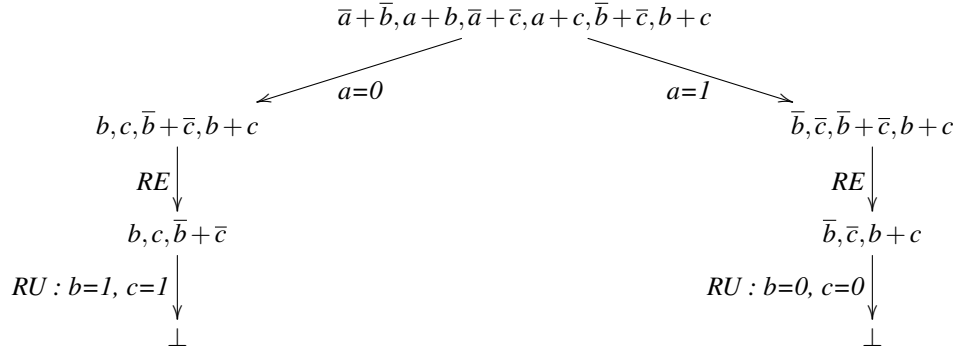
FIGURE 2.1 – Algorithme de Davis-Putnam-Logemann-Loveland.

Pour obtenir une trace de l'algorithme, nous supprimons les clauses valides, en abrégé VAL, puis nous dessinons l'arbre des appels avec l'argument de la fonction ainsi que les ensembles obtenus par les étapes 2 (réduction, en abrégé RE), 3 (enlèvement des clauses ayant des littéraux isolés, en abrégé ELI) et 4 (résolution unitaire, abrégé en RU). À cause du « ou alors », il est inutile de poursuivre la construction de cet arbre dès que la réponse vrai (attachée à un ensemble vide) a été obtenue. Dans ce cas, un modèle peut être exhibé en remontant la récursion, en tenant compte des simplifications : le modèle doit rendre vrai chaque littéral isolé ou clause unitaire trouvé.

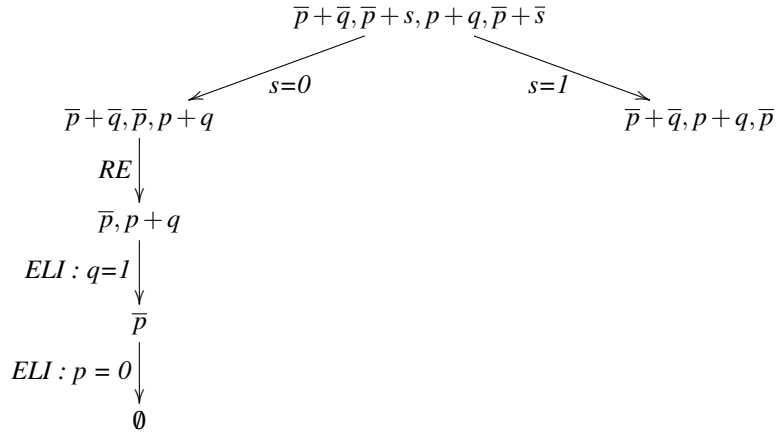
Exemple 2.3.8 Nous illustrons l'application de cet algorithme sur deux ensembles de clauses.

— Soit Γ l'ensemble de clauses : $\bar{a} + \bar{b}, a + b, \bar{a} + \bar{c}, a + c, \bar{b} + \bar{c}, b + c$. Nous donnons la trace de l'algorithme.

Puisque toutes les feuilles portent la clause vide, l'ensemble Γ est insatisfaisable.



— Soit Γ l'ensemble de clauses : $\bar{p} + \bar{q}, \bar{p} + s, p + q, \bar{p} + \bar{s}$. Nous donnons la trace de l'algorithme.



Puisqu'une feuille porte l'ensemble vide, l'ensemble Γ est satisfaisable. Il est inutile de poursuivre la construction de la branche droite.

Théorème 2.3.9 L'algorithme *Algo_DPLL* est correct.

Preuve : La réduction et les transformations 2.3.1 page 55, 2.3.2 page 56 et 2.3.3 page 57 préservent l'existence d'un modèle, donc au pas 0 et 1, l'algorithme vérifie l'invariant suivant : la valeur courante de l'ensemble de clauses de Γ a un modèle si et seulement si Γ a un modèle. Nous en déduisons immédiatement que les réponses de l'algorithme, fournies au pas 0 ou 1, sont correctes. Au pas 5, pour les mêmes raisons, la valeur courante de l'ensemble de clauses de Γ , lors de l'appel de *DPLL*, a un modèle si et seulement si Γ a un modèle. Supposons les appels récursifs corrects :

- Si *DPLL*($\Gamma[x := 0]$) est vrai, par récurrence $\Gamma[x := 0]$ est satisfaisable donc Γ est satisfaisable (nous utilisons le résultat 2.1.21 page 46 : Γ est satisfaisable ssi $\Gamma[x := 0]$ est satisfaisable ou $\Gamma[x := 1]$ est satisfaisable), ce qui correspond à la valeur vrai de *DPLL*(Γ).
- Si *DPLL*($\Gamma[x := 0]$) est faux, par récurrence $\Gamma[x := 0]$ est insatisfaisable. Dans ce cas, *DPLL*(Γ) vaut *DPLL*($\Gamma[x := 1]$) :
- Supposons que *DPLL*($\Gamma[x := 1]$) soit vrai alors par récurrence $\Gamma[x := 1]$ est satisfaisable donc Γ est satisfaisable, ce qui correspond à la valeur vrai de *DPLL*(Γ).

- Supposons que $\text{DPLL}(\Gamma[x := 1])$ soit faux, alors par récurrence $\Gamma[x := 1]$ est insatisfaisable. Donc Γ est insatisfaisable, ce qui correspond à la valeur faux de $\text{DPLL}(\Gamma)$.

□

Théorème 2.3.10 *L’algorithme Algo_DPLL se termine.*

Preuve : Au pas 0, la fonction Algo_DPLL consiste en un test, puis soit le renvoi de la valeur vrai soit l’appel de la fonction $\text{DPLL}(\Gamma)$. Le pas 0 consiste en la suppression de clauses de l’ensemble de départ, cet ensemble étant fini, le pas 0 termine. Donc, la fonction Algo_DPLL termine si la fonction $\text{DPLL}(\Gamma)$ termine.

Considérons un appel de la fonction $\text{DPLL}(\Gamma)$. Le pas 1 est constitué d’un test et d’instructions élémentaires, donc il termine. Le pas 2 consiste à réduire Γ qui est un ensemble fini. La réduction consistant à supprimer des clauses, elle termine. Le pas 3 consiste à supprimer des clauses, puis éventuellement retourner au pas 1. Le nombre de clauses étant fini, soit l’algorithme termine en 1, soit le pas 4 débute. De la même manière, le pas 4 consiste à supprimer des clauses et des variables, qui sont en nombres finis, puis éventuellement retourner au pas 1. Le nombre de clauses et de variables étant fini, soit l’algorithme termine en 1, soit le pas 5 débute. Le pas 5 consiste en un ou deux appels récursifs sur un ensemble de clauses où une variable a disparu (la valeur d’une variable est fixée). Ainsi, la récurrence en 5 se termine car à chaque appel récursif le nombre de variables diminue strictement. D’où, l’algorithme Algo_DPLL termine. □

Remarque 2.3.11 (Oubli de simplifications) *L’algorithme Algo_DPLL a été donné avec les étapes de « simplification » : suppression des clauses valides (0), réduction (2), enlèvement des littéraux isolés (3) et réduction unitaire (4). L’algorithme reste correct si nous omettons ces étapes, ou si nous ne faisons qu’une partie des simplifications, ce qui est souvent le cas lorsque nous faisons une trace de l’exécution sans machine. Car sans machine il est fréquent d’oublier des simplifications, ce qui ne nuit pas car l’algorithme reste correct.*

Remarque 2.3.12 (Choix de la variable) *Un bon choix pour la variable x de l’étape (5), consiste à choisir la variable qui apparaît le plus souvent. Un meilleur choix consiste à choisir la variable qui va entraîner par la suite le plus de simplifications : il faut faire des compromis entre le temps passé à choisir la « bonne » variable et l’importance des simplifications induites par ce choix. Plusieurs heuristiques classiques de choix de variable sont présentées dans le paragraphe 2.3.5.1.*

2.3.5 Solveurs SAT

Le problème SAT est un problème de décision qui consiste à déterminer si une formule propositionnelle en forme normale conjonctive admet ou non un modèle. Ce problème est le problème NP-complet de référence [?]. Généralement, seule une version de ce problème, appelée 3-SAT, où toutes les clauses de la formule sont constituées d’exactement trois littéraux, est considérée, car des *réductions linéaires* existent pour passer de SAT à 3-SAT. Un exemple de réduction SAT vers 3-SAT est étudié dans l’exercice 49 page 64.

Des programmes performants sont dédiés à la résolution du problème SAT¹. Ces programmes sont appelés des *solveurs SAT* et sont généralement classés en deux catégories :

- Les solveurs SAT « complets » sont généralement des versions améliorées de l’algorithme DPLL. Parmi les solveurs SAT complets les plus utilisés à l’heure actuelle, nous pouvons citer par exemple *zchaff*, *sat*, *march* ou encore *GRASP*.
- Les solveurs SAT « incomplets » sont des algorithmes de semi-décision qui finissent par trouver un modèle à la formule, s’il existe et qui ne terminent pas dans le cas contraire. Ces algorithmes sont fondés sur des marches aléatoires dans l’espace d’état. Dans cette catégorie, nous pouvons citer les algorithmes par exemple *WalkSAT* et *GSAT*.

Nous décrivons maintenant, de manière non-exhaustive, les principales améliorations de l’algorithme DPLL utilisées dans les solveurs SAT complets.

2.3.5.1 Heuristique de branchement

Le choix de la prochaine variable à affecter a un impact important sur la taille de l’arbre de recherche développé par DPLL. Ainsi, le choix de l’heuristique de branchement est déterminant dans les algorithmes fondés sur DPLL. Les heuristiques de branchement sont souvent basées sur des arguments syntaxiques tels que la taille des clauses, le nombre d’occurrences, *etc.* Parmi les heuristiques les plus connues, nous pouvons citer :

1. Dans ces programmes, le formalisme 3-SAT est généralement préféré au formalisme SAT, car son aspect régulier permet d’obtenir des solutions plus efficaces.

MOMS : L'heuristique *MOMS* (pour Maximum Occurrences in Clauses of Minimum Size) consiste, comme son nom l'indique, à sélectionner la variable ayant le plus d'occurrences dans les clauses les plus courtes. Les choix effectués par cette heuristique favorisent la propagation unitaire.

JW : L'heuristique *JW* (pour Jeroslow-Wang) utilise également la longueur des clauses. En effet, un poids est affecté à chaque littéral en fonction de la taille des clauses où il apparaît. La possibilité qu'une variable soit sélectionnée par *JW* est inversement proportionnelle à la somme des tailles des clauses où elle apparaît.

UP : Contrairement aux deux heuristiques précédentes, l'heuristique *UP* (pour Unit Propagation) n'est pas fondée sur des arguments syntaxiques. Cette heuristique essaie de sélectionner une variable en prévoyant à l'avance son effet sur la recherche. Elle consiste à calculer un poids pour chaque variable en fonction des simplifications qu'elle génère par propagation unitaire.

2.3.5.2 Ajout de clauses

L'idée est d'ajouter des clauses, en pré-traitement, à l'ensemble initial de clauses. Le nouvel ensemble de clauses reste équivalent à l'ensemble initial. Par exemple, nous pouvons citer la *résolution restreinte* qui consiste à ajouter des résolvants de taille bornée. Ce type de méthode permet de réduire la taille de l'arbre recherche en amenant aux contradictions plus tôt dans la recherche.

2.3.5.3 Analyse des conflits et retour-arrière non chronologique

Lorsque l'algorithme *DPLL* atteint une contradiction (un conflit), il revient (chronologiquement) au dernier branchement de variable effectué. Or, ce branchement n'est pas nécessairement l'une des décisions à l'origine de la contradiction. L'analyse des conflits remédie à ce problème en tenant compte des décisions à l'origine du conflit. Cette analyse permet ainsi d'effectuer un retour-arrière non-chronologique, c'est-à-dire un retour direct au niveau, plus haut dans l'arbre, qui est à l'origine du conflit sans tester les niveaux intermédiaires.

2.3.5.4 Apprentissage

L'apprentissage consiste à analyser et apprendre les raisons qui ont amené à une contradiction : lorsqu'une assignation partielle amène à une contradiction, il est possible d'isoler un sous-ensemble d'assignations et un sous-ensemble de clauses, qui sont responsables du conflit. À partir de ces assignations, il est possible de construire (d'apprendre) une clause qui est impliquée par ces clauses. Cette nouvelle clause est ajoutée à l'ensemble de clauses du problème. Les assignations pertinentes sont déterminées par un graphe de dépendance entre les clauses et les assignations, appelé *graphe d'implication*. Les clauses apprises permettent de ne pas refaire plusieurs fois les mêmes « erreurs » dans l'arbre de recherche.

2.3.5.5 Redémarrage

La plupart des solveurs SAT actuels utilisent une stratégie de redémarrage, qui consiste à redémarrer la recherche en prenant en compte de nouvelles clauses apprises durant la recherche précédente. La plupart des solveurs effectuent un redémarrage après qu'un seuil de clauses apprises soit atteint. Ainsi, à chaque redémarrage, le parcours de l'espace de recherche change car l'ensemble de clauses est modifié. Il faut noter que la plupart des solveurs effacent régulièrement les clauses apprises. Ainsi, le mixage des redémarrages et des oublis peut dans certains cas mettre en cause la complétude des solveurs. Un paramétrage des redémarrages est donc nécessaire pour garder cette dernière propriété. Le solveur Chaff, par exemple, augmente à chaque redémarrage le seuil au-delà duquel une clause est oubliée. Ainsi, de plus en plus de clauses sont gardées, ce qui assure la complétude.

2.3.5.6 Structures de données paresseuses

Dans les solveurs SAT modernes, 80% du temps de calcul est consommé par la propagation unitaire. Ainsi, des structures de données dites « paresseuses » ont été introduites dans le but d'optimiser la propagation unitaire.

2.4 Exercices

Exercice 31 (Résolvant) Par résolution, nous avons :

$$\frac{a+b \quad \bar{a}+\bar{b}}{b+\bar{b}}$$

Montrer que : $b+\bar{b} \not\models (a+b).(\bar{a}+\bar{b})$.

□

Exercice 32 (Résolvant) Nous rappelons que deux clauses sont égales si et seulement si elles ont les mêmes ensembles de littéraux.

- Les clauses $p+q+\bar{r}+q+p+s+q+\bar{r}$ et $s+q+\bar{r}+p$ sont-elles égales ?
- Les clauses $p+q+r+p$ et $q+r+q+r+q+r$ sont-elles égales ? L'une est-elle incluse dans l'autre ? L'une est-elle la conséquence de l'autre ? Sont-elles équivalentes ?
- Indiquer tous les résolvants des clauses $a+\bar{b}+c$ et $a+b+\bar{c}$. Ces résolvants sont-ils valides ?

□

Exercice 33 (Preuve) Les ensembles de formules suivants sont insatisfaisables.

- $\{a, a \Rightarrow b, \bar{b}\}$.
- $\{a+b, \bar{a}+c, \bar{a}+\bar{d}, d+\bar{c}, \bar{b}+a\}$.
- $\{a+b+c, \bar{a}+b, \bar{b}+c, \bar{c}+a, \bar{a}+\bar{b}+\bar{c}\}$.

En donner une preuve par résolution.

□

Exercice 34 (Formalisation et résolution,*) Remarquons que : « x à moins que y » se formalise en $\bar{x} \Leftrightarrow \bar{y}$. Dans une maison hantée, les esprits se manifestent sous deux formes différentes, un **chant obscène** et un **rire sardonique**, cependant nous pouvons influencer le comportement en jouant de l'orgue ou en brûlant de l'encens. Compte-tenu des données suivantes :

- (i) Le chant ne se fait pas entendre, à moins que nous jouions de l'orgue sans que le rire ne se fasse entendre.
- (ii) Si nous brûlons de l'encens, le rire se fait entendre si et seulement si le chant se fait entendre.
- (iii) (En ce moment) Le chant se fait entendre et le rire est silencieux.

Et de la conclusion :

- (iv) (En ce moment) Nous jouons de l'orgue et ne brûlons pas d'encens.

Nous posons :

- c : le chant se fait entendre.
- o : nous jouons de l'orgue.
- r : le rire se fait entendre.
- e : nous brûlons de l'encens.

1. Simplifier en produit de clauses $\bar{x} \Leftrightarrow \bar{y}$.
2. Formaliser sous forme de produit de clauses les hypothèses et la négation de la conclusion.
3. Prouver par résolution que le raisonnement est correct.

Autrement dit transformer le produit des hypothèses et de la négation de la conclusion en un produit de clauses, et en déduire la clause vide.

□

Exercice 35 (Preuve,*) Montrer, à l'aide d'une preuve par résolution, la correction du raisonnement suivant :

$$r+q \Rightarrow t, t.q \Rightarrow r, q \models t \Leftrightarrow r.$$

□

Exercice 36 (Formalisation et preuve,*) Montrer par résolution que le raisonnement suivant est contradictoire :

- Il fait beau à moins qu'il neige.
- Il pleut à moins qu'il neige.
- Il fait beau à moins qu'il pleuve.

□

Exercice 37 (∞ Définir une clause) Une clause est la clause vide ou une disjonction (non vide) de littéraux. Donner une définition formelle de ce qu'est une clause et définir par récurrence la fonction s telle que $s(C)$ est l'ensemble des littéraux de la clause C .

□

Exercice 38 (∞ Preuve) Prouver la propriété 2.1.14 page 44 sur la monotonie et composition.

□

Exercice 39 (∞ Propriété de la résolution) Montrer la propriété 2.1.8 page 43.

□

Exercice 40 (∞ La résolution n'ajoute aucun littéral,*) Soit Γ un ensemble de clauses. Un littéral de Γ est un littéral d'une clause de Γ . Montrer que toute clause déduite de Γ ne comporte que des littéraux de Γ .

□

Exercice 41 (Réduction, X1603 Andrews) Soit l'ensemble de clauses :

$$\{p + q, \bar{p} + r + \bar{q} + p, p + \bar{r}, q + \bar{p} + \bar{q}, q + \bar{r} + p, r + q + \bar{p} + \bar{r}, \bar{r} + q\}.$$

1. Réduire cet ensemble (la réduction est définie en 2.1.4 page 47)
2. Indiquer si l'ensemble réduit est ou non satisfaisable.

□

Exercice 42 (DPLL) Considérons l'ensemble de clauses suivant :

$$\{\bar{a} + \bar{b} + \bar{f}, a + b + f, e + \bar{a}, \bar{a} + \bar{b}, \bar{a} + c, d + a + \bar{d}, a + b, \bar{a} + \bar{c} + \bar{d}, d\}.$$

- Appliquer l'algorithme Algo_DPLL sur cet ensemble de clauses.
- Conclure si cet ensemble est satisfaisable ou non.
- Donner un modèle ou un contre-modèle obtenu à partir de la trace de l'algorithme.

Dans l'arbre d'appel vous étiquetterez les étapes comme suit :

- Suppression des clauses valides, en abrégé VAL.
- Réduction, en abrégé RE.
- Suppression des clauses ayant des littéraux isolés, en abrégé ELI.
- Résolution unitaire, abrégé en RU.

De plus, noter les affectations effectuées à chaque étape de l'algorithme afin de retrouver facilement le modèle construit.

□

Exercice 43 (DPLL) Utiliser l'algorithme Algo_DPLL pour déterminer si les ensembles suivants de clauses sont satisfaisables ou insatisfaisables :

- $\{a + b + c + d + e + f, \bar{a} + b, \bar{b} + a, \bar{c} + d, \bar{d} + c, \bar{b} + \bar{c}, \bar{b} + c, b + \bar{c}, \bar{e}, \bar{f}\}.$
- $\{a + b + c + d + f, \bar{a} + b, \bar{b} + a, \bar{c} + d, \bar{d} + c\}.$
- $\{b + j + \bar{a}, a + j + \bar{b}, b + a + j, a + j, j + b, \bar{b} + j, j + b, j + s, \bar{s} + \bar{b}\}.$
- $\{a + \bar{c} + d, \bar{b} + c + f, b + \bar{e} + f, \bar{c} + e + \bar{f}, e + f, c + d, \bar{a}, \bar{e} + \bar{f}\}.$

Donner une trace de l'algorithme.

□

Exercice 44 (Stratégie complète) Soit Γ le produit suivant de clauses :

$$(a + b + \bar{c}).(b + \bar{c}).(c + \bar{c}).(b + c).(\bar{a} + \bar{b} + c).(a + \bar{b} + c).$$

Déterminer par la stratégie complète de résolution si Γ est insatisfaisable ou possède un modèle. Donner la trace de l'algorithme. Indiquer la ou les preuves obtenues. Si Γ a un modèle indiquez-le.

□

Exercice 45 (Stratégie complète) Soient l'ensemble de clauses suivant :

$$\{p + q, \bar{p} + s, \bar{s} + t, \bar{t}, \bar{q} + r, \bar{r}, \bar{r} + p + t, q + z + \bar{z}, \bar{q} + r + s\}.$$

Appliquer l'algorithme de la stratégie complète sur cet ensemble de clauses et conclure si cet ensemble est satisfaisable ou non.

□

Exercice 46 (Stratégie complète) Considérons la fonction f telle que $f(x, y, z) = 0$ si et seulement si le nombre d'arguments valant 1 est pair. Exprimer f comme un produit de clauses suivant la méthode décrite dans la sous-section 1.6 page 31 puis simplifier f en utilisant la stratégie complète. □

Exercice 47 (Preuve) Prouver le lemme 2.3.2 page 55. □

Exercice 48 (Preuve) Prouver le lemme 2.3.5 page 56. □

Exercice 49 (De SAT à 3-SAT,*)** SAT est un problème de décision qui peut être énoncé comme suit : « étant donné un ensemble de clauses Γ , existe-t-il une assignation qui soit modèle de Γ ? » 3-SAT est une restriction de ce problème où toutes les clauses de Γ ont exactement trois littéraux.

SAT est un problème NP-complet [?]. Dans cet exercice, nous proposons d'étudier une réduction polynomiale de SAT vers 3-SAT, prouvant ainsi que 3-SAT est aussi NP-complet².

Une réduction polynomiale de SAT vers 3-SAT consiste à transformer en temps polynomial un ensemble de clauses Γ en un ensemble de clauses Γ' vérifiant les deux propriétés suivantes :

- (a) Γ' est uniquement constitué de clauses ayant trois littéraux distincts.
- (b) Γ a un modèle si et seulement si Γ' a un modèle.

Généralement, de telles réductions vérifient une propriété supplémentaire :

- (c) Tout modèle de Γ' est un modèle de Γ .

Le but de cet exercice est de montrer que la transformation polynomiale présentée ci-dessous vérifie les trois propriétés énoncées ci-dessus. Noter que cette transformation introduit des variables supplémentaires.

Soit $\Gamma = \{c_1, \dots, c_m\}$ un ensemble de clauses. La réduction consiste à remplacer toute clause $c_i = z_1 + \dots + z_k$ de Γ (ATTENTION : z_1, \dots, z_k sont des littéraux) par un ensemble de clauses C'_i construit en fonction de la valeur de k , comme suit :

$k = 1$: $C'_i = \{z_1 + y_1 + y_2, z_1 + y_1 + \bar{y}_2, z_1 + \bar{y}_1 + y_2, z_1 + \bar{y}_1 + \bar{y}_2\}$ où y_1 et y_2 sont des variables supplémentaires.

$k = 2$: $C'_i = \{z_1 + z_2 + y_1, z_1 + z_2 + \bar{y}_1\}$ où y_1 est une variable supplémentaire.

$k = 3$: $C'_i = \{c_i\}$.

$k > 3$: $C'_i = \{z_1 + z_2 + y_1, \bar{y}_1 + z_3 + y_2, \bar{y}_2 + z_4 + y_3, \bar{y}_3 + z_5 + y_4, \dots, \bar{y}_{k-3} + z_{k-1} + z_k\}$ où y_1, \dots, y_{k-3} sont des variables supplémentaires.

Ainsi, $\Gamma' = \bigcup_{i=1}^m C'_i$.

Par construction, la réduction proposée vérifie la propriété (a). Les réponses aux questions suivantes permettent de prouver les propriétés (b) et (c) :

1. Montrer (sans table de vérité) que toute assignation donne la même valeur à c_i et C'_i lorsque c_i est constituée d'un seul littéral.
2. Montrer (sans table de vérité) que toute assignation donne la même valeur à c_i et C'_i lorsque c_i est constituée de deux littéraux.
3. Soit c_i une clause de plus de 3 littéraux. Montrer que si c_i a un modèle, alors C'_i a aussi un modèle.
4. Soit c_i une clause de plus de 3 littéraux. Montrer que tout modèle de C'_i est modèle de c_i .
5. La réduction proposée maintient uniquement la satisfaisabilité. Montrer qu'elle ne maintient pas le sens.

□

2. Il faut noter qu'en général, un problème k -SAT n'est pas nécessairement NP-complet. Par exemple, 2-SAT a été prouvé polynomial.

Chapitre 3

Déduction Naturelle

Sommaire

3.1	Système formel de la déduction naturelle	65
3.1.1	Règles de conjonction	66
3.1.2	Règles de disjonction	66
3.1.3	Règles de l'implication	66
3.1.4	Deux règles spéciales	67
3.1.5	Preuves en déduction naturelle	67
3.2	Tactiques de preuve	71
3.3	Cohérence de la déduction naturelle	73
3.4	Complétude de la déduction naturelle	73
3.5	Outils	75
3.5.1	Logiciel de construction automatique de preuves	76
3.5.2	Dessiner des arbres de preuves	76
3.6	Exercices	77

UNE preuve dans les cours de Mathématiques est une décomposition du raisonnement en pas élémentaires évidents, nous pratiquons alors sans le savoir la *déduction naturelle*. Dans ce chapitre, nous présentons un système formel qui est une modélisation particulière de la déduction naturelle. La déduction naturelle et le calcul de séquent furent introduits en 1934 par Gerhard Gentzen (1909-1945) [?, ?]. Les preuves dans notre système ne sont pas des arbres comme dans le système original de Gerhard Gentzen mais ressemblent plus aux preuves des mathématiciens. Nous expliquons aussi comment faire le lien entre ces deux formalismes en construisant des arbres proches de ceux introduits originellement. Nous choisissons de ne parler que de la *logique classique*, par opposition à la *logique intuitionniste* qui est obtenue en omettant la règle de réduction à l'absurde (la dernière règle de la table 3.1 page 67).

Remarque préliminaire : Pour simplifier l'étude de la déduction naturelle, nous considérons que le vrai, la négation et l'équivalence sont des *abréviations* ainsi définies :

- \top abrège $\perp \Rightarrow \perp$.
- $\neg A$ abrège $A \Rightarrow \perp$.
- $A \Leftrightarrow B$ abrège $(A \Rightarrow B) \wedge (B \Rightarrow A)$.

Deux formules seront considérées comme *égales*, si les formules obtenues en éliminant les abréviations sont identiques. Par exemple, les formules $\neg\neg a$, $\neg a \Rightarrow \perp$ et $(a \Rightarrow \perp) \Rightarrow \perp$ sont égales. Il est clair que deux formules égales, au sens ci-dessus, sont équivalentes (voir exercice 53 page 77) et nous utilisons implicitement cette propriété quand nous remplaçons une formule par une autre formule égale aux abréviations près.

Plan : Nous décrivons tout d'abord notre système de règles de déduction naturelle et introduisons la notion de preuve dans ce contexte. Ensuite, nous présentons des tactiques pour aider le lecteur à la rédaction de preuve. Nous montrons la cohérence et la complétude de notre système. Enfin nous présentons un outil permettant à partir d'une formule de construire automatiquement une preuve ou bien d'exhiber un contre-exemple.

3.1 Système formel de la déduction naturelle

Nous présentons d'abord les règles du système de la déduction naturelle. Elles constituent les raisonnements élémentaires autorisés. Ensuite nous exposons la notion de preuve qui se construit par applications successives de ces règles.

Définition 3.1.1 (Règle) Une règle est constituée de formules dites hypothèses H_1, \dots, H_n et d'une conclusion. Les hypothèses sont écrites au-dessus d'un trait et l'unique formule en dessous de ce trait est la conclusion de la règle. Le nom d'une règle est écrit au même niveau que le trait séparant hypothèses et conclusion.

$$\frac{H_1 \dots H_n}{C} R$$

La déduction naturelle que nous présentons comporte une dizaine de règles de déduction qui sont regroupées dans la table 3.1 page 67. La règle fondamentale est la suivante : si nous pouvons déduire une formule B d'une hypothèse A , alors nous pouvons déduire $A \Rightarrow B$ en nous passant de cette hypothèse. Nous décrivons chacune des dix règles du système de déduction que nous considérons. L'application de ces règles et seulement de ces règles va nous permettre de prouver la correction des raisonnements. Il est donc important de bien comprendre le fonctionnement de chacune d'entre elles. Nous avons deux familles de règles pour chaque connecteur : les règles d'introduction et les règles d'élimination. L'intuition du rôle des règles d'introduction est de générer un symbole de la logique afin de bâtir progressivement la formule à prouver. Les règles d'élimination servent à simplifier un raisonnement ou à obtenir une nouvelle formule grâce aux raisonnements déjà réalisés et ainsi faire disparaître un connecteur logique. En plus nous avons deux règles spéciales, que nous décrirons en dernier, une pour éliminer deux occurrences successives de la négation et une autre pour déduire d'importe quoi à partir du faux.

3.1.1 Règles de conjonction

La règle d'introduction de la conjonction, notée $\wedge I$, signifie simplement que si nous avons une preuve de A et une preuve de B , alors nous pouvons déduire une preuve de la proposition $A \wedge B$. Nous pouvons aussi dire que la preuve de la proposition $A \wedge B$ peut se décomposer en une preuve de A et une preuve de B .

$$\frac{A \quad B}{A \wedge B} \wedge I$$

Les règles d'élimination de la conjonction, notées $\wedge E1$ et $\wedge E2$, permettent de déduire à partir de la conjonction de deux formules $A \wedge B$ soit la formule A soit la formule B . Ainsi nous avons éliminé le connecteur de la conjonction.

$$\frac{A \wedge B}{A} \wedge E1 \quad \frac{A \wedge B}{B} \wedge E2$$

3.1.2 Règles de disjonction

Les règles d'introduction de la disjonction sont les duales des règles de l'élimination de la conjonction. Les deux règles d'introduction, notées $\vee I1$ et $\vee I2$, permettent de créer une disjonction de deux formules à partir d'une des deux formules. Ces règles signifient simplement que si A est vraie alors $A \vee B$ et $B \vee A$ sont aussi vraies quelle que soit la proposition B .

$$\frac{A}{A \vee B} \vee I1 \quad \frac{A}{B \vee A} \vee I2$$

La règle d'élimination de la disjonction, notée $\vee E$, est plus complexe. Afin de déduire la formule C à partir de la disjonction $A \vee B$ il faut aussi prouver les prémisses suivantes : A implique C et B implique C . Cette règle formalise la notion de *preuve par cas* utilisée en mathématiques : supposons que dans un environnement donné, nous souhaitons prouver C alors que deux cas, A ou B , sont possibles ; nous nous plaçons alors dans le cas où A est vérifiée et nous prouvons C , puis nous nous plaçons dans le cas où B est vérifiée et nous prouvons C .

$$\frac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C} \vee E$$

3.1.3 Règles de l'implication

La règle de l'élimination de l'implication $\Rightarrow E$ dit que si nous sommes capables d'obtenir une preuve de A et une preuve de $A \Rightarrow B$ alors nous avons une preuve de B . Cette règle correspond au *modus ponens*.

$$\frac{A \quad A \Rightarrow B}{B} \Rightarrow E$$

La règle fondamentale de notre système est la règle d'*introduction de l'implication* $\Rightarrow I$: si nous pouvons déduire une formule B d'une hypothèse A , alors nous pouvons déduire $A \Rightarrow B$ en nous passant de cette hypothèse. Cette règle est résumée par le schéma ci-dessous, où la notation $[A]$ indique que si A est une hypothèse de la preuve de B , cette hypothèse est *enlevée* de la preuve de $A \Rightarrow B$, autrement dit ne sert plus pour prouver $A \Rightarrow B$.

$$\frac{\begin{array}{c} [A] \\ \vdots \\ B \end{array}}{A \Rightarrow B} \Rightarrow I$$

3.1.4 Deux règles spéciales

Enfin nous avons deux règles spéciales :

- La première permet de déduire n'importe quoi à partir du faux : il s'agit de la « règle du faux », notée *Efq*, pour la formule latine « ex falso, quodlibet », indiquant que du faux, on peut déduire ce qu'on veut.

$$\frac{\perp}{A} \text{Ef}q$$

- La seconde est la règle de réduction à l'absurde, notée *RAA* pour « reductio ad absurdum ». Elle élimine deux occurrences successives de la négation.

$$\frac{\neg \neg A}{A} \text{RAA}$$

Nous regroupons l'ensemble des règles de la déduction naturelle dans la table 3.1.

3.1.5 Preuves en déduction naturelle

Intuitivement une preuve est l'application successive de règles de la déduction naturelle. Dans le système originel introduit par Gerhard Gentzen, une preuve est représentée par un arbre de formules (certaines formules pouvant être marquées comme enlevées). Nous avons choisi de représenter une preuve par une succession d'étapes de raisonnement basées sur les règles de la déduction naturelle, ceci afin d'être plus proche des preuves faites en mathématiques. Dans le chapitre précédent, une preuve par résolution était constituée d'une liste de clauses. En déduction naturelle, au cours d'une preuve, nous pouvons ajouter des hypothèses et les enlever, ce qui rend plus complexe la définition de ce qu'est une preuve. Nous introduisons donc plusieurs notions intermédiaires afin de définir clairement la notion de preuve en déduction naturelle. Tout d'abord nous introduisons les composants d'une preuve, c'est-à-dire, les « lignes d'une preuve », puis la notion de brouillon de preuve, ensuite les contextes de ces lignes de preuve afin de définir une preuve en déduction naturelle.

3.1.5.1 Brouillon de preuve

Une preuve est constituée d'une succession de lignes qui sont soit une formule soit une formule précédée par le mot clef « Supposons » ou « Donc ».

Définition 3.1.2 (Ligne de preuve) Une ligne de preuve est de l'une des trois formes suivantes :

- *Supposons* formule,
- formule,
- *Donc* formule.

Un brouillon de preuve est une preuve en construction, intuitivement elle a au moins autant de *Supposons* que de *Donc*.

Introduction	Élimination
$\frac{\begin{array}{c} [A] \\ \vdots \\ B \\ A \Rightarrow B \end{array}}{A \Rightarrow B} \Rightarrow I$	$\frac{A \quad A \Rightarrow B}{B} \Rightarrow E$
$\frac{A \quad B}{A \wedge B} \wedge I$	$\frac{A \wedge B}{A} \wedge E1 \quad \frac{A \wedge B}{B} \wedge E2$
$\frac{A}{A \vee B} \vee I1 \quad \frac{A}{B \vee A} \vee I2$	$\frac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C} \vee E$
Règle du faux	Règle de réduction à l'absurde
$\frac{\perp}{A} Efq$	$\frac{\neg \neg A}{A} RAA$

TABLE 3.1 – Ensemble de règles de déduction naturelle.

Définition 3.1.3 (Brouillon de preuve) *Un brouillon de preuve est une suite de lignes telle que, dans tout préfixe de la suite, le nombre de lignes commençant par le mot Supposons est au moins égal à celui des lignes commençant par le mot Donc.*

Exemple 3.1.4 *Dans l'exemple ci-dessous, la suite des lignes de 1 à 3 est un brouillon de preuve. Par contre la suite des lignes de 1 à 5, n'est pas un brouillon de preuve car dans la suite des lignes 1 à 4, le nombre de lignes commençant par le mot Donc dépasse celui des lignes commençant par le mot Supposons.*

numéro	ligne
1	Supposons a
2	$a \vee b$
3	Donc $a \Rightarrow a \vee b$
4	Donc $\neg a$
5	Supposons b

3.1.5.2 Contexte des lignes d'un brouillon de preuve

Afin de pouvoir appliquer la règle d'introduction de l'implication nous avons besoin de connaître à chaque étape d'une preuve les formules qui sont des hypothèses utilisables. Pour ce faire, nous associons à chaque ligne d'un brouillon de preuve un *contexte*, qui est la suite des hypothèses introduites par les lignes Supposons et non enlevées par les lignes Donc. Puisque dans tout préfixe d'un brouillon de preuve, le nombre de lignes commençant par le mot Supposons est au moins égal à celui des lignes commençant par le mot Donc, le contexte de la ligne précédant une ligne Donc n'est pas vide et par suite le contexte de chaque ligne d'un brouillon de preuve est bien défini. Dans la suite, nous représentons les contextes de chaque ligne en remplaçant chaque hypothèse du contexte par le numéro de la ligne où cette hypothèse est introduite.

Définition 3.1.5 (Contexte) *Nous numérotions les lignes d'un brouillon de preuve de 1 à n . Pour i de 1 à n , la liste de formules Γ_i est le contexte de la ligne i . La liste Γ_0 est vide et les listes de formules Γ_i sont ainsi définies :*

- Si la ligne i est « Supposons A », alors $\Gamma_i = \Gamma_{i-1}, i$.
- Si la ligne i est « A » alors $\Gamma_i = \Gamma_{i-1}$
- Si la ligne i est « Donc A » alors Γ_i est obtenue en enlevant la dernière formule de Γ_{i-1}

La liste Γ_i est le contexte de la ligne i .

Nous présentons nos preuves sous forme de tableaux, ce qui nous permet de numéroté les lignes d'une preuve, les contextes utilisables, et les règles utilisées pour générer une nouvelle ligne.

Exemple 3.1.6 Nous illustrons ici la gestion des contextes sur un brouillon de preuve simple.

contexte	numéro	ligne	règle
1	1	Supposons a	
1,2	2	Supposons b	
1,2	3	$a \wedge b$	$\wedge I$ 1,2
1	4	Donc $b \Rightarrow a \wedge b$	$\Rightarrow I$ 2,3
1,5	5	Supposons e	

3.1.5.3 Preuves

Nous précisons la notion de conclusion et de formule utilisable afin de définir et manipuler des preuves en déduction naturelle.

Définition 3.1.7 (Conclusion, formule utilisable) Nous définissons la notion de conclusion et de formule utilisable :

- La formule figurant sur une ligne d'un brouillon de preuve est la conclusion de la ligne.
- La conclusion d'une ligne est utilisable tant que son contexte (c'est-à-dire les hypothèses qui ont permis de la déduire) est présent.

Autrement dit la conclusion de la ligne i est utilisable sur la ligne i et sur toutes les lignes suivantes dont le contexte a pour préfixe le contexte de la ligne i .

Exemple 3.1.8 Dans l'exemple ci-dessous, la conclusion de la ligne 2 est utilisable sur la ligne 2, et pas au delà, car à la ligne 3, l'hypothèse qui a permis de la déduire est enlevée.

contexte	numéro	ligne	règle
1	1	Supposons a	
1	2	$a \vee b$	$\vee I$ 1
	3	Donc $a \Rightarrow a \vee b$	$\Rightarrow I$ 1,2
4	4	Supposons c	
4	5	$c \vee a$	$\vee I$ 4
	6	Donc $c \Rightarrow c \vee a$	$\Rightarrow I$ 4,5
	7	$(a \Rightarrow a \vee b) \wedge (c \Rightarrow c \vee a)$	$\wedge I$ 3,6

Nous donnons la définition de preuve dans un environnement. Un environnement est un ensemble de formules supposées « vraies ». Intuitivement, réaliser une preuve dans un environnement consiste à pouvoir utiliser sans les prouver les formules de l'environnement.

Définition 3.1.9 (Preuve) Soit Γ un ensemble de formules, une preuve dans l'environnement Γ est un brouillon de preuve ayant les propriétés suivantes :

1. Pour toute ligne « Donc A », la formule A est égale à $B \Rightarrow C$, où B est la dernière formule du contexte de la ligne précédente et où C est une formule utilisable sur la ligne précédente ou égale à un élément de l'environnement Γ .
2. Pour toute ligne « A », la formule A est la conclusion d'une règle (autre que la règle d'introduction de l'implication) dont les prémisses sont utilisables sur la ligne précédente ou sont éléments de l'environnement Γ .

La ligne « Donc A » est une application de la règle d'introduction de l'implication. En effet, C est déduite de Γ ou d'hypothèses qui figurent dans la ligne précédente. La liste des hypothèses de la ligne précédente se terminant par B , nous pouvons en déduire $B \Rightarrow C$, en nous passant de l'hypothèse B .

Définition 3.1.10 (Preuve d'une formule) Une preuve de la formule A dans l'environnement Γ est soit la preuve vide lorsque A est élément de Γ , soit une preuve dont la dernière ligne est de conclusion A et de contexte vide.

Nous notons $\Gamma \vdash A$ le fait qu'il y a une preuve de A dans l'environnement Γ et $\Gamma \vdash P : A$ le fait que P est une preuve de A dans l'environnement Γ . Lorsque l'environnement est vide, nous l'omettons, autrement dit nous abrégeons $\emptyset \vdash A$ en $\vdash A$. Lorsque nous demandons une preuve d'une formule sans indiquer d'environnement, nous supposons que l'environnement est vide.

Exemple 3.1.11 Prouvons $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$.

contexte	numéro	preuve	règle
1	1	Supposons $a \Rightarrow b$	
$1,2$	2	Supposons $\neg b$	
$1,2,3$	3	Supposons a	
$1,2,3$	4	b	$\Rightarrow E\ 1,3$
$1,2,3$	5	\perp	$\Rightarrow E\ 2,4$
$1,2$	6	Donc $\neg a$	$\Rightarrow I\ 3,5$
1	7	Donc $\neg b \Rightarrow \neg a$	$\Rightarrow I\ 2,6$
	8	Donc $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$	$\Rightarrow I\ 1,7$

La preuve elle-même est ce qui figure dans la colonne preuve. Nous y avons ajouté la numérotation des lignes de preuves, les justifications qui indiquent les règles utilisées et les contextes de chaque ligne de la preuve.

Nous présentons la même preuve sous forme d'arbre, ce qui correspond à l'approche originelle de Gerhard Gentzen.

$$\begin{array}{c}
 \frac{\frac{\frac{(1)\cancel{a \Rightarrow b} \quad (3)\cancel{a}}{(2)\cancel{\neg b} \quad (4)b} \Rightarrow E}{(5)\perp} \Rightarrow I[3]}{(6)\neg a} \Rightarrow I[2]}{(7)\neg b \Rightarrow \neg a} \Rightarrow I[1]}{(8)(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)} \Rightarrow I[1]
 \end{array}$$

Nous remarquons que pour lever une hypothèse il est d'usage de barrer cette hypothèse.

Exemple 3.1.12 Nous donnons la preuve de $\neg A \vee B$ dans l'environnement $A \Rightarrow B$. Nous suggérons de numéroté par i, ii, iii, etc... les formules de l'environnement pour distinguer ces numéros et les numéros des lignes de la preuve.

environnement			
référence		formule	
i		$A \Rightarrow B$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg(\neg A \vee B)$	
1,2	2	Supposons A	
1,2	3	B	$\Rightarrow E$ i,2
1,2	4	$\neg A \vee B$	$\vee I2$ 3
1,2	5	\perp	$\Rightarrow E$ 1,4
1	6	Donc $\neg A$	$\Rightarrow I$ 2,5
1	7	$\neg A \vee B$	$\vee I1$ 6
1	8	\perp	$\Rightarrow E$ 1,7
	9	Donc $\neg\neg(\neg A \vee B)$	$\Rightarrow I$ 1,8
	10	$\neg A \vee B$	RAA 9

Nous présentons aussi la preuve précédente sous forme d'arbre.

$$\begin{array}{c}
 \frac{(i)A \Rightarrow B \quad (2)A'}{\frac{(3)B}{\Rightarrow E}} \\
 \frac{(1)\neg(\neg A \vee B) \quad \frac{(4)\neg A \vee B}{\vee I2}}{\Rightarrow E} \\
 \frac{(5)\perp}{\Rightarrow I[2]} \\
 \frac{(6)\neg A}{\vee I1} \\
 \frac{(1)\neg(\neg A \vee B) \quad (7)\neg A \vee B}{\Rightarrow E} \\
 \frac{(8)\perp}{\Rightarrow I[1]} \\
 \frac{(9)\neg\neg(\neg A \vee B)}{RAA} \\
 (10)\neg A \vee B
 \end{array}$$

Dans l'arbre de preuve, l'environnement est constitué des formules portées par les feuilles non enlevées.

Comme il a été rappelé dans les remarques préliminaires, nous considérons comme égales deux formules qui sont identiques quand nous éliminons leurs abréviations. Ainsi dans l'exemple précédent, nous avons identifié $A \Rightarrow \perp$ et $\neg A$ ainsi que $(\neg A \vee B) \Rightarrow \perp$ et $\neg(\neg A \vee B)$.

3.2 Tactiques de preuve

Nous donnons des conseils pour construire des preuves. Ces conseils sont la base du logiciel présenté dans le paragraphe 3.5 page 75. Pour prouver la formule A dans un environnement Γ , nous appliquons *dans l'ordre* les règles suivantes, où l'ordre a été choisi de façon à reculer le plus possible l'usage de la règle RAA :

1. Si $A \in \Gamma$, alors la preuve obtenue est vide.
2. Si A est la conséquence d'une règle dont les prémisses sont dans Γ , alors la preuve obtenue est « A ».
3. Si Γ comporte une contradiction, c'est-à-dire une formule B et une formule $\neg B$, alors la preuve obtenue est « \perp , A ».
4. Si $A = B \wedge C$, alors :
 - prouver B : Soit P la preuve obtenue pour B ,
 - prouver C : Soit Q la preuve obtenue pour C .
 La preuve obtenue pour A est « P, Q, A ».

Les preuves peuvent échouer (si l'on demande de prouver une formule improuvable dans l'environnement donné) : Si la preuve de B ou C échoue, il en est de même de la preuve de A . Pour simplifier la suite, nous ne signalons plus les cas d'échecs, sauf s'ils doivent être suivis d'une autre preuve.

5. Si $A = B \Rightarrow C$, alors prouver C sous l'hypothèse B (qui est ajoutée à l'environnement). Soit P , la preuve obtenue pour C , la preuve obtenue pour A est « Supposons B, P , Donc A ».
6. Si $A = B \vee C$, alors prouver B : Si P est la preuve obtenue pour B , alors « P, A » est la preuve obtenue pour A . Si la preuve de B échoue, alors prouver C : Si P est la preuve obtenue pour C , alors « P, A » est la preuve obtenue pour A . Si la preuve de C échoue, essayer les règles suivantes.
7. Si $B \wedge C$ est dans l'environnement, alors prouver A à partir des formules B, C , qui remplacent $B \wedge C$ dans l'environnement et soit P le résultat de cette preuve. Alors « B, C, P » est une preuve de A dans l'environnement initial.
8. Si $B \vee C$ est dans l'environnement, alors :
 - prouver A dans l'environnement où B remplace $B \vee C$: Soit P la preuve obtenue,
 - prouver A dans l'environnement où C remplace $B \vee C$: Soit Q la preuve obtenue.
 La preuve de A est alors « Supposons B, P , Donc $B \Rightarrow A$, Supposons C, Q , Donc $C \Rightarrow A, A$ ».
9. Si $\neg(B \vee C)$ est dans l'environnement, alors nous déduisons $\neg B$ par la preuve $P4$ et $\neg C$ par la preuve $P5$ (preuves demandées à l'exercice 57 page 78). Soit P la preuve de A dans l'environnement où $\neg B, \neg C$ remplacent la formule $\neg(B \vee C)$. La preuve de A est « $P4, P5, P$ ».
10. Si $A = B \vee C$, alors prouver C sous l'hypothèse $\neg B$: soit P la preuve obtenue. « Supposons $\neg B, P$, Donc $\neg B \Rightarrow C$ » est une preuve de la formule $\neg B \Rightarrow C$ qui est équivalente à A . Pour obtenir la preuve de A , il suffit d'ajouter la preuve $P1$, demandé à l'exercice 57 page 78, de A dans l'environnement $\neg B \Rightarrow C$. La preuve obtenue de A est donc « Supposons $\neg B, P$, Donc $\neg B \Rightarrow C, P1$ ».
11. Si $\neg(B \wedge C)$ est dans l'environnement, alors nous en déduisons $\neg B \vee \neg C$ par la preuve $P3$ demandée à l'exercice 57 page 78 puis nous raisonnons par cas comme ci-dessous :
 - prouver A dans l'environnement où $\neg B$ remplace $\neg(B \wedge C)$: Soit P la preuve obtenue,
 - prouver A dans l'environnement où $\neg C$ remplace $\neg(B \wedge C)$: Soit Q la preuve obtenue.
 La preuve de A est « $P3$, Supposons $\neg B, P$, Donc $\neg B \Rightarrow A$, Supposons $\neg C, Q$, Donc $\neg C \Rightarrow A, A$ ».
12. Si $\neg(B \Rightarrow C)$ est dans l'environnement, alors nous déduisons B par la preuve $P6$, $\neg C$ par la preuve $P7$ (preuves demandées à l'exercice 57 page 78). Soit P la preuve de A dans l'environnement où $B, \neg C$ remplacent la formule $\neg(B \Rightarrow C)$. La preuve de A est « $P6, P7, P$ ».
13. Si $B \Rightarrow C$ est dans l'environnement et si $C \neq \perp$, autrement dit si $B \Rightarrow C$ n'est pas égale à $\neg B$, alors, nous déduisons $\neg B \vee C$ dans l'environnement $B \Rightarrow C$ par la preuve $P2$ demandée à l'exercice 57 page 78 puis nous raisonnons par cas :
 - prouver A dans l'environnement où $\neg B$ remplace $B \Rightarrow C$: Soit P la preuve obtenue,
 - prouver A dans l'environnement où C remplace $B \Rightarrow C$: Soit Q la preuve obtenue.
 La preuve de A est « $P2$, Supposons $\neg B, P$, Donc $\neg B \Rightarrow A$, Supposons C, Q , Donc $C \Rightarrow A, A$ ».

Exemple 3.2.1 Nous appliquons ces tactiques à la preuve de la loi de Peirce¹ : $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$. Cette formule n'est pas prouvable en logique intuitionniste, qui est, ainsi qu'il est dit au début du chapitre, la logique classique privée de la règle de réduction à l'absurde. Nous pouvons montrer, qu'en logique intuitionniste, cette formule est équivalente à la loi du tiers-exclu.

Vu la forme de la formule, nous devons appliquer la tactique 5. La preuve de la formule de Peirce est donc de la forme suivante :

Preuve Q :

Supposons $(p \Rightarrow q) \Rightarrow p$

Q_1 preuve de p dans l'environnement $(p \Rightarrow q) \Rightarrow p$

Donc $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$

La preuve Q_1 utilise nécessairement la tactique 13. Donc cette preuve s'écrit : Dans l'environnement $B \Rightarrow C$ où $B = p \Rightarrow q$, $C = p$.

1. Charles Sanders Peirce était un logicien et philosophe américain (1839 - 1914).

<i>Preuve Q_1 :</i> <div> $Q_{11} = P_2$ où P_2 est la preuve de $\neg B \vee C$ dans l'environnement $B \Rightarrow C$, voir exercice 57 page 78 </div> Supposons $\neg B$ <div> Q_{12} preuve de $A = p$ dans l'environnement $\neg B$ </div> Donc $\neg B \Rightarrow A$ Supposons C <div> Q_{13} preuve de $A = p$ dans l'environnement C </div> Donc $C \Rightarrow A$ A

Q_{13} est la preuve vide, car $A = C = p$.

Q_{12} est la preuve de $A = p$ dans l'environnement $\neg(p \Rightarrow q)$. Cette preuve est la preuve P_6 demandée à l'exercice 57 page 78, où $B = p$ et $C = q$. En recollant les morceaux Q_1 , Q_{11} , Q_{12} , Q_{13} , nous obtenons la preuve Q , que nous ne recopions pas ici, car elle peut être obtenue automatiquement par le logiciel développé par Michel Levy, disponible à l'adresse suivante : <http://teachinglogic.liglab.fr/DN/>

Nous montrons ci-dessous comment retrouver la preuve Q_{12} sans utiliser les tactiques. La seule règle, ne conduisant pas à une impasse, est la réduction à l'absurde. Donc cette preuve est de la forme :

<i>Preuve Q_{12} de p dans l'environnement $\neg(p \Rightarrow q)$</i> Supposons $\neg p$ <div> Q_{121} preuve de \perp dans l'environnement $\neg(p \Rightarrow q), \neg p$ </div> Donc $\neg\neg p$ p
--

Pour obtenir une contradiction, donc une preuve de \perp , il faut déduire $p \Rightarrow q$. Donc la preuve Q_{121} est :

Supposons p \perp q Donc $p \Rightarrow q$ \perp
--

3.3 Cohérence de la déduction naturelle

La preuve de cohérence de la déduction naturelle consiste à montrer que chaque preuve construite à partir d'un ensemble de formules Γ donne une formule qui est déductible de Γ .

Théorème 3.3.1 (Cohérence de la déduction) *Si une formule A est déduite d'un environnement de formules Γ ($\Gamma \vdash A$), alors A est une conséquence de Γ ($\Gamma \models A$).*

Preuve : Soit Γ un ensemble de formules. Soit P une preuve de A dans cet environnement. Soient C_i la conclusion et H_i le contexte de la $i^{\text{ème}}$ ligne de la preuve P . Nous posons $H_0 = \emptyset$, la liste vide. Si la preuve P est vide, nous posons $C_0 = A$. Notons par Γ, H_i l'ensemble des formules de l'ensemble Γ et de la liste H_i . Nous montrons par induction que pour tout k nous avons $\Gamma, H_k \models C_k$, ce qui implique que pour la dernière ligne n de cette preuve, comme par définition H_n est la liste vide et $C_n = A$, nous obtenons $\Gamma \models A$.

Cas de base : Supposons que A est déduite de Γ par la preuve vide. Alors A est élément de Γ , donc $\Gamma \models A$. Puisque $H_0 = \emptyset$, nous pouvons conclure : $\Gamma, H_0 \models A$, donc $\Gamma, H_0 \models C_0$.

Induction : Supposons que pour toute ligne $i < k$ de la preuve nous avons $\Gamma, H_i \models C_i$. Montrons que $\Gamma, H_k \models C_k$. Supposons que la ligne k est :

- « Supposons C_k ». La formule C_k est la dernière formule de H_k , donc $\Gamma, H_k \models C_k$.
- « Donc C_k ». La formule C_k est égale à la formule $B \Rightarrow D$, et B est la dernière formule de H_{k-1} . Nous distinguons deux cas, soit D est élément de Γ , soit D est utilisable sur la ligne précédente.
 - Dans le premier cas, puisque D est égale à une formule de Γ , D est conséquence de Γ donc de Γ, H_k . Puisque $B \Rightarrow D$ est conséquence de D , il en résulte que $\Gamma, H_k \models C_k$.

- (b) Dans le deuxième cas, D est utilisable sur la ligne précédente. Donc il existe $i < k$ tel que $D = C_i$ et H_i est préfixe de H_{k-1} . Par *hypothèse de récurrence*, $\Gamma, H_i \models D$. Puisque H_i est préfixe de H_{k-1} , nous avons $\Gamma, H_{k-1} \models D$. Puisque B est la dernière formule de H_{k-1} , nous avons $H_{k-1} = H_k, B$ et donc $\Gamma, H_k, B \models D$, ce qui implique $\Gamma, H_k \models B \Rightarrow D$. Enfin puisque C_k est égale à $B \Rightarrow D$ et que deux formules égales sont équivalentes, nous avons $\Gamma, H_k \models C_k$.
3. « C_k ». Cette formule est la conclusion d'une règle de la table 3.1 page 67, appliquée à ses prémisses utilisables à la ligne précédente ou aux éléments de Γ . Considérons le seul cas de la règle $\wedge I$, les autres cas étant analogues. La formule C_k est égale à $(D \wedge E)$ et les prémisses de la règle sont D et E . Puisque D et E sont éléments de Γ ou utilisables à la ligne précédente, comme dans le cas précédent, en utilisant l'hypothèse de récurrence, nous avons : $\Gamma, H_{k-1} \models D$ et $\Gamma, H_{k-1} \models E$. Puisque la ligne k ne change pas les hypothèses, nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models D$ et $\Gamma, H_k \models E$. Puisque C_k est égale à $(D \wedge E)$, nous avons : $D, E \models C_k$. Par suite $\Gamma, H_k \models C_k$. \square

3.4 Complétude de la déduction naturelle

Nous prouvons la complétude des règles uniquement pour les formules avec les symboles logiques $\perp, \wedge, \vee, \Rightarrow$. La complétude pour les formules obtenues en ajoutant les symboles logiques \top, \neg et \Leftrightarrow résulte immédiatement du fait que ces symboles peuvent être considérés comme des abréviations.

Vu l'absence de négation, nous définissons un littéral comme étant une variable ou une implication entre une variable et \perp . Soit x une variable, x et $x \Rightarrow \perp$ (qui peut être abrégé en $\neg x$) sont des littéraux complémentaires. Dans la suite nous distinguons une liste de formules Γ et $s(\Gamma)$ l'ensemble des formules de la liste. Pour simplifier les notations, nous utilisons la virgule pour ajouter un élément en début ou en fin de liste ainsi que pour concaténer deux listes, que ces listes soit des listes de formules ou des preuves.

Théorème 3.4.1 Soient Γ un ensemble fini de formules et A une formule, si $\Gamma \models A$ alors $\Gamma \vdash A$.

Preuve : Soit m une mesure des formules et des listes de formules, ainsi définie :

- $m(\perp) = 0$,
- $m(x) = 1$ où x est une variable,
- $m(\Rightarrow) = 1$,
- $m(\wedge) = 1$,
- $m(\vee) = 2$,
- $m((A \circ B)) = m(A) + m(\circ) + m(B)$,
- la mesure d'une liste de formules est la somme des mesures des formules de la liste.

Puisque les formules $\neg A$ et $(A \Rightarrow \perp)$ sont égales aux abréviations près, nous avons : $m(\neg A) = m((A \Rightarrow \perp)) = m(A) + 1$. Par exemple soit $A = (a \vee \neg a)$, nous avons $m(\neg a) = 2$, $m(A) = 5$ et $m(A, (b \wedge b), A) = 13$.

Soit $P(n)$ la propriété suivante : soient Γ une liste de formules et A une formule telles que la mesure de la liste Γ, A est n , si $s(\Gamma) \models A$ alors $s(\Gamma) \vdash A$.

Supposons que pour tout $i < k$, la propriété $P(i)$ est vérifiée. Supposons que $m(\Gamma, A) = k$ et que $s(\Gamma) \models A$. Pour prouver $P(k)$, donc aussi $P(n)$ pour tout n , il suffit de montrer que : $s(\Gamma) \vdash A$. Nous posons que A est indécomposable si A est \perp ou une variable, et Γ est indécomposable si Γ est une liste de littéraux ou comprend la formule \perp . Nous étudions trois cas :

Cas 1 : ni A , ni Γ ne sont décomposables.

Cas 2 : A est décomposable. Nous décomposons A en deux sous-formules B et C , nous obtenons les inégalités suivantes : $m(\Gamma, B) < m(\Gamma, A)$ et $m(\Gamma, C) < m(\Gamma, A)$. Ainsi nous appliquons l'hypothèse de récurrence puis nous concluons.

Cas 3 : Γ est décomposable. Nous permutons Γ afin d'obtenir une liste qui commence par une formule B décomposable. Nous décomposons B , puis nous remplaçons B dans Γ par les formules obtenues en décomposant B . Ainsi, nous obtenons un nouvel environnement Γ' tel que $m(\Gamma', A) < m(\Gamma, A)$. Nous appliquons l'hypothèse de récurrence puis nous concluons.

Cas 1 : Lorsqu'il est impossible de décomposer A et Γ , alors A est \perp ou une variable et Γ est une liste de littéraux ou comprend la formule \perp . Nous distinguons deux cas :

- (a) Soit \perp est une formule de Γ . Alors, A peut se déduire de \perp par la règle Efq , donc $s(\Gamma) \vdash A$ (par la preuve « A »).
- (b) Soit \perp n'est pas une formule de Γ . Donc, Γ est une liste de littéraux et nous avons deux possibilités :

- $A = \perp$. Puisque $s(\Gamma) \models A$, la liste Γ comporte deux littéraux complémentaires, donc A (i.e., \perp) peut se déduire de Γ par la règle $\Rightarrow E$, et par suite $s(\Gamma) \vdash A$ (par la preuve « A »).
- A est une variable. Puisque $s(\Gamma) \models A$:
 - soit Γ comporte deux littéraux complémentaires et comme dans le cas précédent, on déduit \perp par la règle $\Rightarrow E$. Alors, A peut se déduire de \perp par la règle Efq , donc $s(\Gamma) \vdash A$ (par la preuve « \perp, A »).
 - soit A est élément de Γ et dans ce cas nous avons aussi $s(\Gamma) \vdash A$ (par la preuve vide).

Cas 2 : A est décomposable.

- (c) Supposons que $A = (B \wedge C)$. Nous savons que $s(\Gamma) \models B$ et $s(\Gamma) \models C$. Les mesures de B et C sont strictement inférieures à celles de A . Donc $m(\Gamma, B) < k$ et $m(\Gamma, C) < k$, et par *hypothèse de récurrence*, il existe deux preuves P et Q telles que $s(\Gamma) \vdash P : B$ et $s(\Gamma) \vdash Q : C$. Puisque A peut se déduire de B et C par la règle $\wedge I$: « P, Q, A » est une preuve de A dans l'environnement $s(\Gamma)$ donc $s(\Gamma) \vdash A$.
- (d) Supposons que $A = (B \Rightarrow C)$. Puisque $s(\Gamma) \models A$, il en résulte que $s(\Gamma, B) \models C$. La somme des mesures de B et C est strictement inférieure (de 1) à celle de A . Donc $m(\Gamma, B, C) < k$, et par *hypothèse de récurrence*, il existe une preuve P telle que $s(\Gamma, B) \vdash P : C$. Puisque nous pouvons déduire A de C en appliquant la règle $\Rightarrow I$ qui enlève l'hypothèse B : « Supposons B, P , Donc A » est une preuve de A dans l'environnement $s(\Gamma)$ donc $s(\Gamma) \vdash A$.
- (e) Supposons que $A = (B \vee C)$. Puisque $s(\Gamma) \models A$, il en résulte que $s(\Gamma, \neg B) \models C$.
Puisque $m(A) = m(B) + m(C) + 2$ et que $m(\neg B) = m(B) + 1$, la somme des mesures de $\neg B$ et de C est strictement inférieure (de 1) à celle de A . Donc $m(\Gamma, \neg B, C) < k$, et par *hypothèse de récurrence*, il existe une preuve P telle que $s(\Gamma, \neg B) \vdash P : C$.
Soit P_1 (cette preuve est demandée à l'exercice 57 page 78) une preuve de $B \vee C$ dans l'environnement $\neg B \Rightarrow C$.
Puisque nous pouvons déduire $\neg B \Rightarrow C$ de C en appliquant la règle $\Rightarrow I$ qui enlève l'hypothèse $\neg B$:
« Supposons $\neg B, P$, Donc $\neg B \Rightarrow C, P_1$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.

Cas 3 : Γ est décomposable. Par définition, toute permutation de Γ a la même mesure que Γ . La liste Γ est une permutation d'une des listes suivantes :

- (f) $(B \wedge C), \Delta$. Puisque $s(\Gamma) \models A$, il en résulte que $s(B, C, \Delta) \models A$. La somme des mesures de B et C est strictement inférieure (de 1) à celle de $B \wedge C$. Donc $m(B, C, \Delta, A) < m((B \wedge C), \Delta, A) = m(\Gamma, A) = k$. Par *hypothèse de récurrence*, il existe une preuve P telle que $s(B, C, \Delta) \vdash P : A$. Puisque B peut se déduire de $B \wedge C$ par la règle $\wedge E1$ et que C peut se déduire de $B \wedge C$ par la règle $\wedge E2$: « B, C, P » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.
- (g) $(B \vee C), \Delta$. Puisque $s(\Gamma) \models A$, il en résulte que $s(B, \Delta) \models A$ et $s(C, \Delta) \models A$. Les mesures des formules B et C sont strictement inférieures à celle de $B \vee C$. Donc $m(B, \Delta, A) < k$ et $m(C, \Delta, A) < k$. Par *hypothèse de récurrence*, il existe deux preuves P et Q telles que $s(B, \Delta) \vdash P : A$ et $s(C, \Delta) \vdash Q : A$.
Puisque A peut se déduire de $B \vee C, B \Rightarrow A, C \Rightarrow A$ par la règle $\vee E$:
« Supposons B, P , Donc $B \Rightarrow A$, Supposons C, Q , Donc $C \Rightarrow A, A$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.
- (h) $(B \Rightarrow C), \Delta$ où $C \neq \perp$. Puisque $s(\Gamma) \models A$, il en résulte que $s(\neg B, \Delta) \models A$ et $s(C, \Delta) \models A$. Comme $C \neq \perp$, la mesure de C n'est pas nulle et donc $m(\neg B) < m(B \Rightarrow C)$. Il est clair que : $m(C) < m(B \Rightarrow C)$. Par suite $m(\neg B, \Delta, A) < k$ et $m(C, \Delta, A) < k$, donc par *hypothèse de récurrence*, il existe deux preuves P et Q telles que $s(\neg B, \Delta) \vdash P : A$ et $s(C, \Delta) \vdash Q : A$.
Soit P_2 (la preuve P_2 est demandée à l'exercice 57 page 78) une preuve de $\neg B \vee C$ dans l'environnement $B \Rightarrow C$. Puisque A peut se déduire de $\neg B \vee C, \neg B \Rightarrow A, C \Rightarrow A$ par la règle $\vee E$:
« P_2 , Supposons $\neg B, P$, Donc $(\neg B \Rightarrow A)$, Supposons C, Q , Donc $(C \Rightarrow A), A$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.
- (i) $\neg(B \wedge C), \Delta$. Puisque $s(\Gamma) \models A$, il en résulte que $s(\neg B, \Delta) \models A$ et $s(\neg C, \Delta) \models A$. Les mesures de $\neg B$ et de $\neg C$ sont strictement inférieures à celle de $\neg(B \wedge C)$. Donc $m(\neg B, \Delta, A) < k$ et $m(\neg C, \Delta, A) < k$, par *hypothèse de récurrence*, il existe deux preuves P et Q telles que $s(\neg B, \Delta) \vdash P : A$ et $s(\neg C, \Delta) \vdash Q : A$.
Soit P_3 (la preuve P_3 est demandée à l'exercice 57 page 78) une preuve de $\neg B \vee \neg C$ dans l'environnement $\neg(B \wedge C)$.
Puisque A peut se déduire de $\neg B \vee \neg C, \neg B \Rightarrow A, \neg C \Rightarrow A$ par la règle $\vee E$:
« P_3 , Supposons $\neg B, P$, Donc $\neg B \Rightarrow A$, Supposons $\neg C, Q$, Donc $\neg C \Rightarrow A, A$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.
- (j) $\neg(B \vee C), \Delta$. Puisque $s(\Gamma) \models A$, il en résulte que $s(\neg B, \neg C, \Delta) \models A$.
Puisque la mesure de \vee est 2, la somme des mesures de $\neg B$ et de $\neg C$ est strictement inférieure (de 1) à celle

de $\neg(B \vee C)$. Donc $m(\neg B, \neg C, \Delta, A) < k$, par *hypothèse de récurrence*, il existe une preuve P telle que : $s(\neg B, \neg C, \Delta) \vdash P : A$.

Soit $P4$ une preuve de $\neg B$ dans l'environnement $\neg(B \vee C)$ et $P5$ une preuve de $\neg C$ dans ce même environnement (les preuves $P4$ et $P5$ sont demandées à l'exercice 57 page 78) :

« $P4, P5, P$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.

- (k) $\neg(B \Rightarrow C), \Delta$. Puisque $s(\Gamma) \models A$, il en résulte que $s(B, \neg C, \Delta) \models A$. La somme des mesures de B et de $\neg C$ est inférieure de 1 à celle de $\neg(B \Rightarrow C)$. Donc $m(B, \neg C, \Delta, A) < k$, et par *hypothèse de récurrence*, il existe une preuve P telle que : $s(B, \neg C, \Delta) \vdash P : A$.

Soient $P6$ une preuve de B dans l'environnement $\neg(B \Rightarrow C)$ et $P7$ une preuve de $\neg C$ dans ce même environnement (les preuves $P6$ et $P7$ sont demandées à l'exercice 57 page 78) :

« $P6, P7, P$ » est une preuve de A dans l'environnement $s(\Gamma)$, donc $s(\Gamma) \vdash A$.

□

Remarque 3.4.2 La preuve de complétude est constructive, c'est-à-dire qu'elle donne un ensemble complet de tactiques pour construire des preuves d'une formule dans un environnement. Cependant ces tactiques peuvent donner des preuves longues. En particulier si nous devons prouver une formule $(B \vee C)$, il vaut mieux en général suivre les tactiques données au paragraphe 3.2 page 71, c'est-à-dire essayer de prouver B , puis essayer de prouver C et seulement en cas d'échec, utiliser la tactique donnée dans la preuve de complétude, qui « réduit » cette preuve à une preuve de C en ajoutant l'hypothèse $\neg B$.

3.5 Outils

Nous indiquons deux logiciels pour se familiariser avec la déduction naturelle. Le premier construit automatiquement des preuves comme nous les avons présentées, le deuxième illustre sur quelques exemples de manière interactive comment dessiner des preuves sous forme d'arbres.

3.5.1 Logiciel de construction automatique de preuves

Pour produire automatiquement des preuves, nous recommandons d'utiliser le logiciel :

<http://teachinglogic.liglab.fr/DN/>

Ce logiciel permet grâce à une syntaxe intuitive de saisir une formule et de générer automatiquement :

- si la formule est (syntaxiquement) incorrecte, un message d'erreur (en rouge) est produit au dessus de la formule
- si la formule est prouvable, sa preuve (sans annotation) est générée par le logiciel.
- si la formule est correcte mais n'est pas prouvable, un contre-modèle est proposé au dessus de la formule.

Si la formule est prouvable, il est aussi possible d'obtenir une version de la preuve annotée.

3.5.2 Dessiner des arbres de preuves

Ceux qui affectionnent les preuves sous forme d'arbres peuvent utiliser le logiciel suivant développé Laurent Théry :

<http://www-sop.inria.fr/marelle/Laurent.Thery/peanoware/Nd.html>

Ce logiciel se présente comme un jeu, il propose des formules à prouver et permet d'appliquer les règles de la déduction naturelle, de manière interactive, pour en construire les arbres de preuve. Quand vous réussissez à construire la preuve d'une formule, vous voyez la photo de Dag Prawitz, un des principaux logiciens qui a étudié les propriétés de la déduction naturelle.

3.6 Exercices

Exercice 50 (Brouillon de preuve) La suite de lignes suivante n'est pas un brouillon de preuve. Indiquer le numéro i de la première ligne telle que les lignes 1 à $i - 1$ soient un brouillon de preuve et que les lignes 1 à i ne soient pas un brouillon de preuve. Préciser le contexte des lignes 1 à $i - 1$.

contexte	numéro	preuve
	1	Supposons a
	2	Supposons b
	3	c
	4	Donc d
	5	Supposons e
	6	f
	7	Donc g
	8	h
	9	i
	10	Donc j
	11	Donc k
	12	l

□

Exercice 51 (Preuves de formules avec les règles spéciales) Donner une preuve pour les formules suivantes :

1. $a \Rightarrow \neg \neg a$,
2. $\neg \neg a \Rightarrow a$,
3. $a \Leftrightarrow \neg \neg a$,
4. $a \vee \neg a$.

□

Exercice 52 (Preuves simples de formules) Donner une preuve pour les formules suivantes :

1. $a \Rightarrow c$ dans l'environnement $a \Rightarrow b, b \Rightarrow c$.
2. $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$.
3. $(a \Rightarrow b) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \Rightarrow c))$.

□

Exercice 53 (\Leftrightarrow Abréviations) Soient A une formule et $dpl(A)$ la formule obtenue en remplaçant dans A , le vrai, les négations et les équivalences (en forme abrégées) par leur définition. $dpl(A)$ est la formule obtenue en « dépliant » A , d'où le nom dpl choisi pour cette fonction de dépliage.

- Définir par récurrence la fonction dpl .
- Montrer que les formules A et $dpl(A)$ sont équivalentes.
- En déduire que deux formules, égales aux abréviations près, sont équivalentes.

□

Exercice 54 (Preuves de formules) Donner une preuve pour les formules suivantes :

1. $a \Rightarrow (b \Rightarrow a)$.
2. $a \wedge b \Rightarrow a$.
3. $\neg a \Rightarrow (a \Rightarrow b)$.
4. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$.
5. $a \wedge b \Rightarrow b \wedge a$.
6. $a \vee b \Rightarrow b \vee a$.
7. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow (a \wedge b \Rightarrow c)$.
8. $(a \wedge b \Rightarrow c) \Rightarrow (a \Rightarrow (b \Rightarrow c))$.

9. $(a \Rightarrow b) \wedge (c \Rightarrow d) \Rightarrow (a \wedge c \Rightarrow b \wedge d).$

□

Exercice 55 (À propos de l'implication) Donner une preuve pour les formules suivantes :

1. $a \Rightarrow b$ dans l'environnement $\neg a \vee b$.
2. $(\neg b \Rightarrow \neg a) \Rightarrow (a \Rightarrow b).$

□

Exercice 56 (Algèbre de Boole) Donner une preuve pour les formules suivantes :

1. $\neg(a \wedge \neg a).$
2. $a \vee a \Rightarrow a.$
3. $a \wedge a \Rightarrow a.$
4. $a \wedge (b \vee c) \Rightarrow a \wedge b \vee a \wedge c.$
5. $a \wedge b \vee a \wedge c \Rightarrow a \wedge (b \vee c).$
6. $a \vee b \wedge c \Rightarrow (a \vee b) \wedge (a \vee c).$
7. $(a \vee b) \wedge (a \vee c) \Rightarrow a \vee b \wedge c.$

□

Exercice 57 (\Leftrightarrow Preuves de formules avec environnement) Donner une preuve des formules suivantes :

1. $P1 : B \vee C$ dans l'environnement $\neg B \Rightarrow C.$
2. $P2 : \neg B \vee C$ dans l'environnement $B \Rightarrow C.$
3. $P3 : \neg B \vee \neg C$ dans l'environnement $\neg(B \wedge C).$
4. $P4 : \neg B$ dans l'environnement $\neg(B \vee C).$
5. $P5 : \neg C$ dans le même environnement $\neg(B \vee C).$
6. $P6 : B$ dans l'environnement $\neg(B \Rightarrow C).$
7. $P7 : \neg C$ dans le même environnement $\neg(B \Rightarrow C).$

□

Exercice 58 (Preuves de formules) Donner une preuve en déduction naturelle des formules suivantes :

1. $\neg(a \vee b) \Rightarrow (\neg a \wedge \neg b).$
2. $(\neg a \wedge \neg b) \Rightarrow \neg(a \vee b).$
3. (*) $\neg(a \wedge b) \Rightarrow (\neg a \vee \neg b).$
4. (**) $(\neg a \vee \neg b) \Rightarrow \neg(a \wedge b).$
5. (***) $(a \vee b) \wedge (\neg a \vee b) \Rightarrow b.$
6. (***) $(a \vee b) \wedge (\neg b \vee c) \Rightarrow a \vee c.$

□

Exercice 59 (Partiel 2011) Prouver les formules suivantes en utilisant la déduction naturelle sous forme de tableau :

1. $\neg p \wedge \neg(\neg p \wedge q) \Rightarrow \neg q.$
2. $((p \wedge q) \wedge (p \wedge r)) \wedge \neg p \Rightarrow q \vee r.$
3. (*) $\neg p \vee \neg(p \wedge q) \Rightarrow \neg q \vee \neg p.$

□

Exercice 60 (Partiel 2013) Donner une preuve des formules suivantes en utilisant la déduction naturelle sous forme de tableau :

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

□

Exercice 61 (Quelques questions posées en examen)

1. Démontrer la formule suivante par déduction naturelle : $(p \vee q) \wedge (p \Rightarrow r) \Rightarrow q \vee r$

2. Soient les hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Formaliser et démontrer qu'on peut en déduire la conclusion (C) : « Marie est la soeur de Jean ou Pierre est grand ou les deux à la fois » par déduction naturelle.

□

Deuxième partie

Logique du premier ordre

Chapitre 4

Logique du premier ordre

Sommaire

4.1	Syntaxe	83
4.1.1	Formules strictes	84
4.1.2	Formules à priorité	85
4.2	Être libre ou lié	87
4.2.1	Occurrences libres et liées	87
4.2.2	Variables libres et liées	88
4.3	Sens des formules	88
4.3.1	Déclaration de symbole	88
4.3.2	Signature	88
4.3.3	Interprétation	90
4.3.4	Sens des formules	90
4.3.5	Modèle, validité, conséquence, équivalence	93
4.3.6	Instanciation	94
4.3.7	Interprétation finie	94
4.3.8	Substitution et remplacement	98
4.4	Équivalences remarquables	98
4.4.1	Relation entre \forall et \exists	98
4.4.2	Déplacement des quantificateurs	99
4.4.3	Changement de variables liées	100
4.5	Exercices	102

Ce célèbre syllogisme¹ ne peut pas se formaliser en logique propositionnelle :

Tous les hommes sont mortels ;
or Socrate est un homme ;
donc Socrate est mortel.

Pour formaliser un tel raisonnement nous avons besoin d'enrichir la logique propositionnelle avec de nouveaux connecteurs appelés *quantificateurs*, cette logique étendue s'appelle logique du premier ordre. Elle permet de parler de structures comportant un seul domaine non vide (contrairement à la logique propositionnelle, ce domaine peut avoir plus de deux valeurs), des fonctions et relations sur ce domaine. Le langage de cette logique comporte plusieurs catégories : les termes qui représentent les éléments du domaine ou des fonctions sur ces éléments, des relations qui relient des termes entre eux et les formules qui décrivent les interactions entre les relations grâce aux connecteurs et aux quantificateurs. Par exemple, la relation *mortel*(x) désigne x est mortel, la relation *homme*(x) signifie que x est un homme. De plus, *Socrate* est une constante du domaine (c'est-à-dire, elle a pour valeur un élément du domaine) et *homme*(*Socrate*) signifie que Socrate est un homme. Enfin, la formule $\forall x (\text{homme}(x) \Rightarrow \text{mortel}(x))$ indique que tous les hommes sont mortels. À partir de ces deux hypothèses, il est possible de conclure que Socrate est mortel (*mortel*(*Socrate*)) par différentes méthodes

1. La notion de syllogisme fut introduite par Aristote et signifie littéralement « parole (qui va) avec (une autre) », aussi connu en latin par *modus ponendo ponens* = « manière d'affirmer, d'établir en affirmant » et plus brièvement *modus ponens*.

que nous détaillerons dans cette seconde partie. Ce même raisonnement s'applique également pour prouver le syllogisme suivant :

Un cheval bon marché est rare.
 Tout ce qui est rare est cher.
 Donc un cheval bon marché est cher.

Malgré la conclusion qui semble contradictoire, le raisonnement est correct. Afin d'obtenir une contradiction il faut ajouter l'hypothèse suivante : $\forall x(\text{bonmarché}(x) \Leftrightarrow \neg \text{cher}(x))$. Ceci traduit la relation communément admise entre la notion de bon marché et de cher, *i.e.*, tout ce qui est cher n'est pas bon marché et réciproquement. Avec cette hypothèse supplémentaire nous pouvons montrer que ce raisonnement est contradictoire, sans cette hypothèse le raisonnement est correct. Ceci élucide le paradoxe de ce syllogisme.

Notre objectif dans ce chapitre est d'introduire les concepts et notions élémentaires de la logique du premier ordre afin de pouvoir proposer des méthodes de raisonnement dans les chapitres suivants.

Plan : Nous commençons par décrire la syntaxe de la logique du premier ordre. Puis, nous définissons les notions de *libre* et *liée* qui sont essentielles pour pouvoir interpréter le sens des formules. Ensuite nous définissons le sens des formules que nous pouvons construire à partir de la syntaxe introduite précédemment. Enfin, nous énonçons des propriétés remarquables de la logique du premier ordre, qui pourront être utilisées dans les raisonnements.

4.1 Syntaxe

Nous ajoutons à la logique propositionnelle deux symboles : le symbole *existentiel* (\exists) et le symbole *universel* (\forall). Le symbole existentiel signifie qu'il existe un élément ayant une certaine propriété alors que le symbole universel permet de parler de tous les éléments ayant une propriété. Ces changements impliquent qu'un élément de ce langage peut désormais dépendre de plusieurs variables. Il faut donc rajouter dans la syntaxe un délimiteur de variables. Nous avons choisi classiquement la virgule. Ces changements introduisent de nouveaux symboles en plus des variables, ces symboles peuvent par exemple être des nombres sur lesquels nous pouvons créer des fonctions (comme l'addition) ou encore définir des relations (comme l'égalité). Toutes ces modifications rendent la présentation de la syntaxe légèrement plus subtile que celle de la logique propositionnelle.

4.1.1 Formules strictes

Pour écrire les formules de la logique du premier ordre nous étendons la syntaxe de la logique propositionnelle, ainsi nous disposons du vocabulaire suivant :

- Deux constantes propositionnelles : \perp et \top représentant respectivement le faux et le vrai.
- Variables : suite de lettres et de chiffres commençant par une des minuscules u, v, w, x, y, z .
- Connecteurs : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$.
- Quantificateurs : \forall le quantificateur universel et \exists le quantificateur existentiel.
- Ponctuations : la virgule « , » et les parenthèses ouvrantes « (» et fermantes «) ».
- Symbole :
 - ordinaire : suite de lettres et de chiffres ne commençant pas par une des minuscules u, v, w, x, y, z .
 - spécial : $+, -, *, /, =, \neq, <, \leq, >, \geq$

L'ensemble des symboles spéciaux peut être augmenté de nouveaux éléments, suivant les domaines mathématiques étudiés, à condition de respecter la contrainte que ces nouveaux éléments ne soient pas dans les ensembles des constantes propositionnelles, des variables, des connecteurs, des quantificateurs, des ponctuations ou des symboles ordinaires.

Exemple 4.1.1 *Ci-dessous, nous illustrons ces notions :*

- $x, x1, x2, y$ sont des variables,
- *homme, parent, succ, 12, 24, f1* sont des symboles ordinaires, les symboles ordinaires représenteront des fonctions (constantes numériques ou fonctions à plusieurs arguments) ou des relations (variables propositionnelles ou relations à plusieurs arguments).
- $x = y, z > 3$ sont des exemples d'application de symboles spéciaux.

Nous définissons en toute généralité la notion de terme qui est essentielle en logique du premier ordre. Nous étendons dans la suite cette notion en définissant ce qu'est un terme associé à un ensemble de symboles.

Définition 4.1.2 (Terme) Un terme est défini de manière inductive par :

- un symbole ordinaire est un terme,
- une variable est un terme,
- si t_1, \dots, t_n sont des termes et si s est un symbole (ordinaire ou spécial) alors $s(t_1, \dots, t_n)$ est un terme.

Exemple 4.1.3

$$x, a, f(x1, x2, g(y)), +(x, *(y, z)), +(5, 42)$$

sont des termes, par contre

$$f(\perp, 2, y)$$

n'est pas un terme. Notons que $42(1, y, 3)$ est aussi un terme, mais l'usage veut que les noms des fonctions et relations soient des symboles ordinaires commençant par des lettres.

Définition 4.1.4 (Formule atomique) Nous définissons une formule atomique de manière inductive par :

- \top et \perp sont des formules atomiques
- un symbole ordinaire est une formule atomique
- si t_1, \dots, t_n sont des termes et si s est un symbole (ordinaire ou spécial) alors $s(t_1, \dots, t_n)$ est une formule atomique.

Exemple 4.1.5

$$f(1, +(5, 42), g(z)), a, \text{ et } +(x, *(y, z))$$

sont des formules atomiques, notons également que

$$x \text{ et } A \vee f(4, 2, 6)$$

ne sont pas des formules atomiques.

Notons que l'ensemble des termes et l'ensemble des formules atomiques ne sont pas disjoints. Par exemple, $p(x)$ est à la fois un terme et une formule atomique. Lorsque t est à la fois un terme et une formule atomique, nous distinguons JtK le sens de t vu comme un terme, de $[t]$ le sens de t vu comme une formule (voir le paragraphe 4.3.4 page 90).

Définition 4.1.6 (Formule) Nous définissons une formule de manière inductive par :

- une formule atomique est une formule,
- si A est une formule alors $\neg A$ est une formule,
- si A et B sont des formules et si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule,
- si A est une formule et si x est une variable quelconque alors $\forall x A$ et $\exists x A$ sont des formules².

Exemple 4.1.7

$$\text{homme}(x), \text{parents}(\text{fils}(y), \text{mere}(\text{Alice})), = (x, +(f(x), g(y)))$$

2. Attention, x peut ne pas exister dans A .

sont des formules atomiques, donc des formules. Par contre

$$\forall x (\text{homme}(x) \Rightarrow \text{homme}(\text{Socrate}))$$

est une formule qui n'est pas atomique.

La notion de *sous-formule* (définition 1.1.4) s'étend naturellement à la logique du premier ordre. La notion de *taille de formule stricte* nécessite qu'en à elle une nouvelle définition :

Définition 4.1.8 (Taille d'une formule) La taille d'une formule A , notée $|A|$, est définie inductivement par :

- $|\top| = 0$ et $|\perp| = 0$.
- Si A est une formule atomique alors $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|Qx A| = 1 + |A|$ si Q est un des quantificateurs \forall ou \exists .
- $|(A \circ B)| = |A| + |B| + 1$.

4.1.2 Formules à priorité

Nous reprenons les priorités de la logique propositionnelle pour les connecteurs, et nous ajoutons une priorité identique à celle de la négation pour les quantificateurs. Dans le tableau 4.1 page suivante nous donnons les priorités des symboles et des connecteurs par ordre décroissant du haut vers le bas. Nous indiquons seulement deux règles de formation des formules à priorité, qui les distinguent des formules complètement parenthésées et permettent d'omettre des parenthèses ou d'ajouter des nouvelles parenthèses.

Pour abrégier l'écriture des termes, certains symboles de fonction $+$, $-$, $*$, $/$ et certains symboles de relations $=$, \neq , $<$, \leq , $>$, \geq peuvent être écrits de manière infixe, c'est-à-dire, de façon usuelle.

Exemple 4.1.9 Nous abrégeons le terme $+(x, *(y, z))$ en $x + y * z$ et $\leq (*(3, x), +(y, 5))$ en $3 * x \leq y + 5$. La transformation inverse est définie en donnant aux symboles $=$, \neq , $<$, \leq , $>$, \geq des priorités inférieures à celle des symboles $+$, $-$, $*$, $/$.

OPÉRATIONS	
— + unaire	
$*, /$	associatif gauche
$+, -$ binaire	associatif gauche
RELATIONS	
$=, \neq, <, \leq, >, \geq$	
NÉGATION, QUANTIFICATEURS	
\neg, \forall, \exists	
CONNECTEURS BINAIRES	
\wedge	associatif gauche
\vee	associatif gauche
\Rightarrow	associatif droit
\Leftrightarrow	associatif gauche

TABLE 4.1 – Priorités des connecteurs et symboles.

Définition 4.1.10 (Formule à priorité) Une formule à priorité est :

- Une formule atomique.
- Si A est une formule à priorité alors $\neg A$ est une formule à priorité.
- Si A et B sont des formules à priorité alors $A \circ B$ est une formule à priorité.
- Si A est une formule à priorité et si x une variable quelconque alors $\forall x A$ et $\exists x A$ sont des formules à priorité.
- Si A est une formule à priorité alors (A) est une formule à priorité.

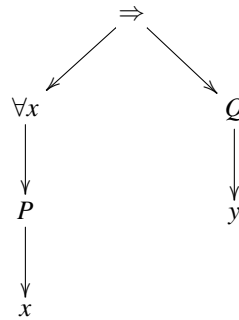
Exemple 4.1.11 La formule (à priorité) $\forall x P(x) \wedge \forall x Q(x) \Leftrightarrow \forall x (P(x) \wedge Q(x))$ peut être vue comme une abréviation de la formule

$$((\forall x P(x) \wedge \forall x Q(x)) \Leftrightarrow \forall x (P(x) \wedge Q(x))).$$

La formule (à priorité) $\forall x \forall y \forall z (x \leq y \wedge y \leq z \Rightarrow x \leq z)$ peut être vue comme une abréviation de la formule

$$\forall x \forall y \forall z ((\leq(x, y) \wedge \leq(y, z)) \Rightarrow \leq(x, z)).$$

Exemple 4.1.12 La priorité du \forall est plus forte que celle de \Rightarrow , dans la formule $\forall x P(x) \Rightarrow Q(y)$. Donc, l'opérande gauche de l'implication est $\forall x P(x)$. La structure de la formule sera ainsi représentée par l'arbre suivant :



De manière analogue à la logique propositionnelle, la taille d'une formule à priorité sera égale à la taille de la formule stricte dont elle est l'abréviation. De même, pour les sous-formules, nous considérerons toujours la formule stricte dont la formule à priorité est l'abréviation.

4.2 Être libre ou lié

Le sens de la formule $x + 2 = 4$ dépend de x : la formule n'est vraie (en arithmétique) que si $x = 2$. La variable x est libre dans cette formule.

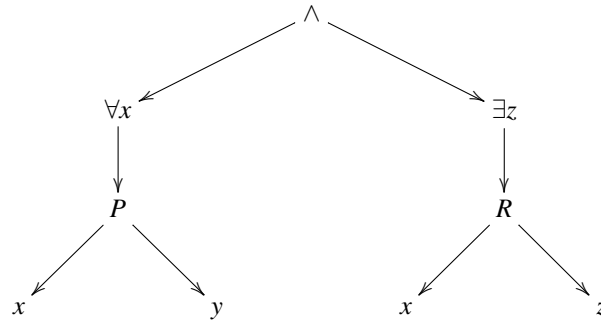
Par contre, toujours en arithmétique $\forall x (x + 2 = 4)$ est une formule fausse et $\exists x (x + 2 = 4)$ est une formule vraie car $2 + 2 = 4$. Pour chacune de ces deux formules, il n'y a pas à choisir de valeur pour x afin de déterminer leur valeur respective : ces deux formules n'ont pas de variables libres.

4.2.1 Occurrences libres et liées

Sur une représentation en arbre, où nous dessinons les structures des formules en faisant apparaître $\forall x$ et $\exists x$ comme des sommets de l'arbre, une *occurrence liée* de la variable x est une occurrence en dessous d'un sommet $\exists x$ ou $\forall x$. Une occurrence de x qui n'est pas sous un tel sommet est *libre*.

Définition 4.2.1 (Portée de liaison, occurrence libre, liée) Soient x une variable et A une formule. Dans une formule $\forall x A$ ou $\exists x A$, la portée de la liaison pour x est A . Une occurrence de x dans une formule est libre si elle n'est pas dans la portée d'une liaison pour x , sinon elle est dite liée.

Exemple 4.2.2 Pour voir les occurrences des variables, nous dessinons la structure de la formule $\forall x P(x, y) \wedge \exists z R(x, z)$, puisque la priorité de \forall est supérieure à celle du \wedge , nous obtenons :



L'occurrence de x présente dans $P(x,y)$ est liée, l'occurrence de x présente dans $R(x,z)$ est libre. L'occurrence de z est liée.

4.2.2 Variables libres et liées

Définition 4.2.3 (Variable libre, liée, formule fermée) La variable x est une variable libre d'une formule si et seulement s'il y a une occurrence libre de x dans la formule. Une variable x est une variable liée d'une formule si et seulement s'il y a une occurrence liée de x dans la formule. Une formule sans variable libre est aussi appelée une formule fermée.

Remarque 4.2.4 Une variable peut-être à la fois libre et liée. Par exemple, dans la formule $\forall x P(x) \vee Q(x)$, x est à la fois libre et liée.

Remarque 4.2.5 Par définition, une variable qui n'apparaît pas dans une formule (0 occurrence) est une variable NON libre de la formule.

Exemple 4.2.6 Les variables libres de la formule de l'exemple 4.2.2 page précédente sont x et y .

4.3 Sens des formules

Dans la suite, nous définissons en logique du premier ordre, le sens des connecteurs, des quantificateurs ainsi que de l'égalité. Aussi, pour définir le sens d'une formule, il suffit d'indiquer la valeur des variables libres et le sens de ses symboles utilisés. Dans un premier temps nous expliquons comment nous déclarons un symbole. Ensuite nous introduisons la notion de signature qui permet de définir les termes et formules associés. Une fois toutes ces notions établies nous définissons les interprétations afin de donner du sens aux formules de la logique du premier ordre.

4.3.1 Déclaration de symbole

Afin de donner un sens aux formules de la logique du premier ordre il nous faut déclarer les symboles utilisés. Nous distinguons en plus des variables, les symboles de fonctions, par exemple $g(x,y)$, et les symboles de relation comme $\text{parent}(x,y)$.

Définition 4.3.1 (Déclaration de symbole) Une déclaration de symbole est un triplet noté s^{gn} où s est un symbole, g est soit f (pour fonction) ou r (pour relation), et n est un entier naturel dénotant le nombre d'arguments de ce symbole, n est aussi appelé l'arité de s .

Exemple 4.3.2 Les fonctions d'arité 0 constituent les constantes, par exemple 1 et 2 sont des constantes d'arité nulle, mais parent^{r2} est une notation signifiant que parent est employé comme relation d'arité 2. De même $*^{f2}$ est une déclaration annonçant que $*$ est une fonction 2-aire. Le symbole A^{r0} est d'arité 0 et signifie que A est une variable propositionnelle. Le symbole homme^{r1} est d'arité 1 ou unaire et le symbole g^{f2} est d'arité 2 ou 2-aire.

Remarque 4.3.3 Lorsque le contexte ou les conventions usuelles comportent une déclaration implicite d'un symbole, nous omettons l'exposant. Par exemple, le symbole égal étant toujours employé comme relation à deux arguments, nous abrégons la déclaration de symbole $=^{r2}$ en $=$.

4.3.2 Signature

En logique du premier ordre nous pouvons choisir le nom des variables utilisées mais nous avons aussi la possibilité de construire nos propres constantes (fonctions sans arguments), fonctions, variables propositionnelles (relations d'arité nulle) et relations. Une signature est donc l'ensemble des déclarations de symboles autorisés pour construire des formules. Elle permet de définir les symboles dont le sens n'est pas fixé *a priori*, contrairement par exemple aux constantes \top et \perp dont le sens est toujours fixé respectivement à 1 et 0.

Définition 4.3.4 (Signature, constante, symbole de fonction, variable propositionnelle, et relation) Une signature Σ est un ensemble de déclarations de symboles de la forme s^{g^n} . Soient n un entier strictement positif et Σ une signature, le symbole s est :

1. une constante de la signature si et seulement si $s^{f^0} \in \Sigma$
2. un symbole de fonction à n arguments de la signature, si et seulement si $s^{f^n} \in \Sigma$
3. une variable propositionnelle de la signature si et seulement si $s^{r^0} \in \Sigma$
4. un symbole de relation à n arguments de la signature, si et seulement si $s^{r^n} \in \Sigma$

Au lieu de dire $0^{f^0}, 1^{f^0}, +^{f^2}, *^{f^2}, =^{r^2}$ est une signature pour l'arithmétique, nous disons plus simplement qu'une signature possible pour l'arithmétique comporte $0, 1, +$ (à deux arguments), $*$ et $=$. Sur cet exemple, nous devons préciser que le symbole *plus* est utilisé avec deux arguments (car nous pouvons aussi rencontrer le symbole *plus* avec un seul argument).

Exemple 4.3.5 Une signature possible pour la théorie des ensembles est $\in, =$. Notons que toutes les autres opérations sur les ensembles peuvent être définies à partir de ces deux symboles.

Définition 4.3.6 (Symbole surchargé) Un symbole est surchargé dans une signature, lorsque cette signature comporte deux déclarations distinctes du même symbole.

Exemple 4.3.7 Il est fréquent d'utiliser des signatures dans lesquelles le signe moins est surchargé. Il est utilisé simultanément pour obtenir l'opposé d'un nombre ou faire la soustraction de deux nombres.

Dans la suite, nous nous interdirons d'utiliser des signatures comportant des symboles surchargés. Cela permet de simplifier les notations, car une fois que nous avons précisé la signature utilisée, nous remplaçons une déclaration de symbole par le symbole lui-même. Nous définissons maintenant les termes associés à une signature. Nous considérons un ensemble dénombrable de variables disjoint de l'ensemble de symboles de la signature.

Définition 4.3.8 (Terme sur une signature) Soit Σ une signature, un terme sur Σ est :

- soit une variable,
- soit une constante s où $s^{f^0} \in \Sigma$,
- soit un terme de la forme $s(t_1, \dots, t_n)$, où $n \geq 1$, $s^{f^n} \in \Sigma$ et t_1, \dots, t_n sont des termes sur Σ .

L'ensemble des termes sur la signature Σ est noté T_Σ .

Définition 4.3.9 (Formule atomique sur une signature) Soit Σ une signature, une formule atomique sur Σ est :

- soit une des constantes \top, \perp ,
- soit une variable propositionnelle s où $s^{r^0} \in \Sigma$,
- soit de la forme $s(t_1, \dots, t_n)$ où $n \geq 1$, $s^{r^n} \in \Sigma$ et où t_1, \dots, t_n sont des termes sur Σ .

Notons que les variables ne sont pas des formules atomiques.

Définition 4.3.10 (Formule sur une signature) Une formule sur une signature Σ est une formule, dont les sous-formules atomiques sont des formules atomiques sur Σ (au sens de la définition 4.3.9).

Nous dénotons l'ensemble des formules sur la signature Σ par F_Σ .

Exemple 4.3.11 $\forall x (p(x) \Rightarrow \exists y q(x, y))$ est une formule sur la signature $\Sigma = \{p^{r^1}, q^{r^2}, h^{f^1}, c^{f^0}\}$. Mais c'est aussi une formule sur la signature $\Sigma' = \{p^{r^1}, q^{r^2}\}$, puisque les symboles h et c ne figurent pas dans la formule.

Définition 4.3.12 (Signature associée à une formule) La signature associée à une formule est la plus petite signature Σ telle que la formule est élément de F_Σ , c'est la plus petite signature permettant d'écrire la formule.

Exemple 4.3.13 La signature Σ associée à la formule $\forall x (p(x) \Rightarrow \exists y q(x, y))$ est

$$\Sigma = \{p^{r1}, q^{r2}\}.$$

Définition 4.3.14 (Signature associée à un ensemble de formules) La signature associée à un ensemble de formules est l'union des signatures associées à chaque formule de l'ensemble.

Exemple 4.3.15 La signature Σ associée à l'ensemble constitué des deux formules $\forall x(p(x) \Rightarrow \exists y q(x, y)), \forall u \forall v (u + s(v) = s(u) + v)$ est

$$\Sigma = \{p^{r1}, q^{r2}, +^{f2}, s^{f1}, =^{r2}\}.$$

4.3.3 Interprétation

En logique propositionnelle, le sens des formules est uniquement fixé par les valeurs des variables, en logique du premier ordre le sens des formules dépend aussi du sens des fonctions et des relations. Le sens des fonctions et des relations est fixé par une interprétation.

Définition 4.3.16 (Interprétation) Une interprétation I sur une signature Σ est définie par un domaine D non vide et une application qui à chaque déclaration de symbole $s^{gn} \in \Sigma$ associe sa valeur s_I^{gn} comme suit :

1. s_I^{f0} est un élément de D .
2. s_I^{fn} où $n \geq 1$ est une fonction de D^n dans D , autrement dit une fonction à n arguments.
3. s_I^{r0} vaut 0 ou 1.
4. s_I^{rn} où $n \geq 1$ est un sous-ensemble de D^n , autrement dit une relation à n arguments.

Exemple 4.3.17 Soit l'interprétation I de domaine $D = \{1, 2, 3\}$ où la relation binaire *ami* est vraie pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire, $\text{ami}_I^2 = \{(1, 2), (1, 3), (2, 3)\}$. *ami* $(2, 3)$ est vraie dans l'interprétation I . En revanche, *ami* $(2, 1)$ est fausse dans l'interprétation I .

Remarque 4.3.18 Dans toute interprétation I , la valeur du symbole $=$ est l'ensemble $\{(d, d) \mid d \in D\}$, autrement dit dans toute interprétation le sens de l'égalité est l'identité sur le domaine de l'interprétation.

Exemple 4.3.19 Considérons une signature suivante.

- Anne^{f0} , Bernard^{f0} et Claude^{f0} : les prénoms Anne, Bernard, et Claude dénotent des constantes,
- a^{r2} : la lettre a dénote une relation à deux arguments (nous lisons $a(x, y)$ comme x aime y) et
- c^{f1} : la lettre c dénote une fonction à un argument (nous lisons $c(x)$ comme le copain ou la copine de x).

Une interprétation possible sur cette signature est l'interprétation I de domaine $D = \{0, 1, 2\}$ où :

- $\text{Anne}_I^{f0} = 0$, $\text{Bernard}_I^{f0} = 1$, et $\text{Claude}_I^{f0} = 2$.
- $a_I^{r2} = \{(0, 1), (1, 0), (2, 0)\}$.
- $c_I^{f1}(0) = 1$, $c_I^{f1}(1) = 0$, $c_I^{f1}(2) = 2$. Notons que la fonction c_I^{f1} a comme domaine D , ce qui oblige à définir artificiellement $c_I^{f1}(2)$: Claude, dénoté par 2, n'a ni copain, ni copine.

Définition 4.3.20 (Interprétation d'un ensemble de formules) L'interprétation d'un ensemble de formules est une interprétation qui définit seulement le sens de la signature associée à l'ensemble des formules.

Définition 4.3.21 (État) Un état e d'une interprétation est une application de l'ensemble des variables dans le domaine de l'interprétation.

Définition 4.3.22 (Assignment) Une assignment est un couple (I, e) composé d'une interprétation I et d'un état e .

Exemple 4.3.23 Soient le domaine $D = \{1, 2, 3\}$ et l'interprétation I où la relation binaire *ami* est vraie uniquement pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire, $\text{ami}_I^2 = \{(1, 2), (1, 3), (2, 3)\}$. Soit e l'état qui associe 2 à x et 1 à y . L'assignment (I, e) rend la relation *ami* (x, y) fausse.

4.3.4 Sens des formules

Nous expliquons maintenant comment évaluer une formule à partir d'une assignation, c'est-à-dire, une interprétation et un état. Il faut noter que dans certains cas, l'état de l'assignation est inutile pour fixer le sens d'une formule.

Remarque 4.3.24 *La valeur d'une formule ne dépend que de ses variables libres et de ses symboles, aussi pour évaluer une formule sans variable libre, l'état des variables est inutile. Nous avons alors deux possibilités :*

- *Pour une formule sans variables libres, il suffit de donner une interprétation I des symboles de la formule. Dans ce cas, les assignations (I, e) et (I, e') donneront la même valeur à la formule pour tous états e et e' . Ainsi pour tout état e , nous identifierons (I, e) et I . Selon le contexte, I sera considéré comme soit une interprétation soit une assignation dont l'état est quelconque.*
- *Pour une formule avec des variables libres, nous avons donc besoin d'une assignation.*

Soient Σ une signature, I une interprétation sur Σ de domaine D et e un état de cette interprétation. Nous souhaitons connaître la valeur (0 ou 1) de toute formule A sur Σ dans l'assignation (I, e) . Cette valeur sera notée $[A]_{(I, e)}$. Pour cela, nous avons tout d'abord besoin de définir le sens de chaque terme t sur Σ dans l'assignation (I, e) , noté $JtK_{(I, e)}$. Ensuite nous devons fixer le sens de chaque formule atomique B sur Σ dans la même assignation, noté $[B]_{(I, e)}$.

4.3.4.1 Sens des termes sur une signature

Chacun sait intuitivement comment évaluer un terme : nous remplaçons les variables par leurs valeurs, les symboles de fonctions par les fonctions qui leur sont associées et nous appliquons les fonctions. Mais pour raisonner sur le sens des termes ou écrire un programme d'évaluation des termes, nous devons formaliser cette évaluation.

Définition 4.3.25 (Évaluation) *Nous donnons la définition inductive de l'évaluation d'un terme t :*

1. *si t est une variable, alors $JtK_{(I, e)} = e(t)$,*
2. *si t est une constante alors $JtK_{(I, e)} = t_I^{f0}$,*
3. *si $t = s(t_1, \dots, t_n)$ où s est un symbole et t_1, \dots, t_n sont des termes, alors $JtK_{(I, e)} = s_I^{fn}(Jt_1K_{(I, e)}, \dots, Jt_nK_{(I, e)})$.*

Dans les exemples, nous remplaçons les variables par leurs valeurs, nous confondons les symboles et leur sens.

Exemple 4.3.26 *Soit I l'interprétation de domaine \mathbb{N} qui donne aux déclarations de symboles $1^{f0}, *^{f2}, +^{f2}$ leur sens usuel sur les entiers. Soit e l'état tel que $x = 2, y = 3$. Calculons $Jx * (y + 1)K_{(I, e)}$.*

$$Jx * (y + 1)K_{(I, e)} = JxK_{(I, e)} * J(y + 1)K_{(I, e)} = JxK_{(I, e)} * (JyK_{(I, e)} + J1K_{(I, e)}) = e(x) * (e(y) + 1) = 2 * (3 + 1) = 8.$$

4.3.4.2 Sens des formules atomiques sur une signature

Définition 4.3.27 (Sens des formules atomiques) *Le sens des formules atomiques est donné par les règles inductives suivantes :*

1. $[\top]_{(I, e)} = 1, [\perp]_{(I, e)} = 0$. Dans les exemples, nous autorisons à remplacer \top par sa valeur 1 et \perp par sa valeur 0.
2. Soit s une variable propositionnelle, $[s]_{(I, e)} = s_I^{r0}$.
3. Soit $A = s(t_1, \dots, t_n)$ où s est un symbole et t_1, \dots, t_n sont des termes. Si $(Jt_1K_{(I, e)}, \dots, Jt_nK_{(I, e)}) \in s_I^{rn}$ alors $[A]_{(I, e)} = 1$ sinon $[A]_{(I, e)} = 0$.

Remarque 4.3.28 *Nous devons distinguer entre évaluer un terme JtK et évaluer une formule atomique $[]$, car dans notre présentation, ces deux catégories syntaxiques ne sont pas disjointes, car $s(t_1, \dots, t_n)$ peut être l'application d'une fonction (JK) ou d'une relation ($[]$). Mais quand le contexte est sans ambiguïté, nous pouvons omettre cette distinction.*

Exemple 4.3.29 *En reprenant l'exemple 4.3.19 page ci-contre, nous obtenons :*

$$— [a(\text{Anne}, \text{Bernard})]_I =$$

$$1 \text{ car } (J\text{Anne}K_I, J\text{Bernard}K_I) = (0, 1) \in a_I^{r2}.$$

$$— [a(\text{Anne}, \text{Claude})]_I =$$

$$0 \text{ car } (J\text{Anne}K_I, J\text{Claude}K_I) = (0, 2) \notin a_I^{r2}.$$

Soit e l'état $x = 0, y = 2$. Nous avons :

$$— [a(x, c(x))]_{(I,e)} =$$

$$1 \text{ car } (JxK_{(I,e)}, Jc(x)K_{(I,e)}) = (0, c_I^{f1}(JxK_{(I,e)})) = (0, c_I^{f1}(0)) = (0, 1) \in a_I^{r2}.$$

$$— [a(y, c(y))]_{(I,e)} =$$

$$0 \text{ car } (JyK_{(I,e)}, Jc(y)K_{(I,e)}) = (2, c_I^{f1}(JyK_{(I,e)})) = (2, c_I^{f1}(2)) = (2, 2) \notin a_I^{r2}.$$

Attention à distinguer (suivant le contexte), les éléments du domaine $0, 1$ et les valeurs de vérité $0, 1$. Les exemples suivants mettent en évidence l'interprétation de l'égalité comme une identité. Dans l'interprétation I , nous avons :

$$— [(Anne = Bernard)]_I =$$

$$0, \text{ car } (J\text{Anne}K_I, J\text{Bernard}K_I) = (0, 1) \notin a_I^{r2}.$$

$$— [(c(\text{Anne}) = \text{Anne})]_I =$$

$$0, \text{ car } (Jc(\text{Anne})K_I, J\text{Anne}K_I) = (c_I^{f1}(J\text{Anne}K_I), 0) = (c_I^{f1}(0), 0) = (1, 0) \notin a_I^{r2}.$$

$$— [(c(c(\text{Anne})) = \text{Anne})]_I =$$

$$1, \text{ car } (Jc(c(\text{Anne}))K_I, J\text{Anne}K_I) = (c_I^{f1}(Jc(\text{Anne})K_I), 0) = (c_I^{f1}(c_I^{f1}(0)), 0) = (c_I^{f1}(1), 0) = (0, 0) \in a_I^{r2}.$$

4.3.4.3 Sens des formules sur une signature

Nous pouvons maintenant ajouter le sens des quantificateurs et aussi celui des opérateurs qui reste identique à celui donné pour la logique propositionnelle (cf. sous-section 1.2.1 page 13).

Définition 4.3.30 (Sens des formules) Le sens des formules est donné par :

1. Les connecteurs propositionnels ont le même sens qu'en logique propositionnelle. Soient B et C des formules, rappelons uniquement le sens de l'implication : si $[B]_{(I,e)} = 0$ alors $[(B \Rightarrow C)]_{(I,e)} = 1$ sinon $[(B \Rightarrow C)]_{(I,e)} = [C]_{(I,e)}$.
2. Soient x une variable et B une formule. $[\forall x B]_{(I,e)} = 1$ si et seulement si $[B]_{(I,f)} = 1$ pour tout état f identique à e , sauf pour x . Soit $d \in D$. Notons $e[x=d]$ l'état identique à l'état e , sauf pour la variable x , auquel l'état $e[x=d]$ associe la valeur d . La définition ci-dessus peut être mise sous la forme suivante :

$$[\forall x B]_{(I,e)} = \min_{d \in D} [B]_{(I,e[x=d])} = \prod_{d \in D} [B]_{(I,e[x=d])},$$

où le produit est le produit booléen.

Cette définition permet de calculer la valeur de $\forall x B$ dans le cas où D est un domaine fini. Mais quand D n'est pas un domaine fini, c'est seulement une traduction du quantificateur universel d'un formalisme dans un autre.

3. $[\exists x B]_{(I,e)} = 1$ si et seulement s'il y a un état f identique à e , sauf pour x , tel que $[B]_{(I,f)} = 1$. La définition ci-dessus peut être mise sous la forme suivante :

$$[\exists x B]_{(I,e)} = \max_{d \in D} [B]_{(I,e[x=d])} = \sum_{d \in D} [B]_{(I,e[x=d])},$$

où la somme est la somme booléenne.

Exemple 4.3.31 Soit l'interprétation I de domaine $D = \{1, 2, 3\}$ où la relation binaire *ami* est vraie pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire, $\text{ami}_I^2 = \{(1, 2), (1, 3), (2, 3)\}$. La formule $\text{ami}(1, 2) \wedge \text{ami}(2, 3) \Rightarrow \text{ami}(1, 3)$ est vraie dans l'interprétation I , i.e., $[\text{ami}(1, 2) \wedge \text{ami}(2, 3) \Rightarrow \text{ami}(1, 3)]_I = 1$.

Exemple 4.3.32 Utilisons l'interprétation I donnée dans l'exemple 4.3.19 page 90. D'après le sens du quantificateur existentiel, nous devons évaluer $a(x, x)$ pour $x = 0$, $x = 1$, et $x = 2$. Nous simplifions ce calcul et les calculs suivants en remplaçant immédiatement les variables par leurs valeurs : cela nous évite d'utiliser les états.

$$- [\exists x a(x, x)]_I =$$

$$\max\{[a(0, 0)]_I, [a(1, 1)]_I, [a(2, 2)]_I\} = 0 \text{ car } (0, 0), (1, 1), (2, 2) \notin a_I^2.$$

$$D'après la définition, nous avons : $[\exists x a(x, x)]_I = [a(0, 0)]_I + [a(1, 1)]_I + [a(2, 2)]_I = 0 + 0 + 0 = 0.$$$

$$- [\forall x \exists y a(x, y)]_I =$$

$$\min\{\max\{[a(0, 0)]_I, [a(0, 1)]_I, [a(0, 2)]_I\}, \max\{[a(1, 0)]_I, [a(1, 1)]_I, [a(1, 2)]_I\}, \max\{[a(2, 0)]_I, [a(2, 1)]_I, [a(2, 2)]_I\}\} =$$

$$\min\{\max\{0, 1, 0\}, \max\{1, 0, 0\}, \max\{1, 0, 0\}\} = \min\{1, 1, 1\} = 1.$$

$$D'après la définition, nous avons : $[\forall x \exists y a(x, y)]_I = ([a(0, 0)]_I + [a(0, 1)]_I + [a(0, 2)]_I) \cdot ([a(1, 0)]_I + [a(1, 1)]_I + [a(1, 2)]_I) \cdot$

$$([a(2, 0)]_I + [a(2, 1)]_I + [a(2, 2)]_I) = (0 + 1 + 0) \cdot (1 + 0 + 0) \cdot (1 + 0 + 0) = 1 \cdot 1 \cdot 1 = 1.$$$$

$$- [\exists y \forall x a(x, y)]_I =$$

$$\max\{\min\{[a(0, 0)]_I, [a(1, 0)]_I, [a(2, 0)]_I\}, \min\{[a(0, 1)]_I, [a(1, 1)]_I, [a(2, 1)]_I\}, \min\{[a(0, 2)]_I, [a(1, 2)]_I, [a(2, 2)]_I\}\} =$$

$$\max\{\min\{0, 1, 1\}, \min\{1, 0, 0\}, \min\{0, 0, 0\}\} = \max\{0, 0, 0\} = 0.$$

$$D'après la définition, nous avons : $[\exists y \forall x a(x, y)]_I = [a(0, 0)]_I \cdot [a(1, 0)]_I \cdot [a(2, 0)]_I + [a(0, 1)]_I \cdot [a(1, 1)]_I \cdot [a(2, 1)]_I + [a(0, 2)]_I \cdot$

$$[a(1, 2)]_I \cdot [a(2, 2)]_I = 0 \cdot 1 \cdot 1 + 1 \cdot 0 \cdot 0 + 0 \cdot 0 \cdot 0 = 0 + 0 + 0 = 0.$$$$

Remarque 4.3.33 Dans l'interprétation ci-dessus, les formules $\forall x \exists y a(x, y)$ et $\exists y \forall x a(x, y)$ n'ont pas la même valeur. Donc en intervertissant un quantificateur existentiel et un quantificateur universel, nous ne préservons pas le sens des formules.

4.3.5 Modèle, validité, conséquence, équivalence

Ces notions sont définies comme en logique propositionnelle. Mais alors qu'en logique propositionnelle, une assignation est une application des variables propositionnelles dans 0, 1, en logique du premier ordre une assignation est un couple constitué d'une interprétation des symboles d'une part et de l'état des variables d'autre part.

4.3.6 Instanciation

Définition 4.3.34 (Instanciation) Soit x une variable, t un terme et A une formule.

1. $A\langle x := t \rangle$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t .
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x dans A .

Exemple 4.3.35 Le terme z est libre pour x dans la formule $\exists y p(x, y)$. Par contre le terme y , comme tout terme comportant la variable y , n'est pas libre pour x dans cette formule. Soit A la formule $(\forall x P(x) \vee Q(\mathbf{x}))$, la formule $A\langle x := b \rangle$ vaut

$(\forall x P(x) \vee Q(b))$ car seule l'occurrence en gras de x est libre.

Théorème 4.3.36 Soient A une formule et t un terme libre pour la variable x dans A . Soient I une interprétation et e un état de l'interprétation. Nous avons $[A\langle x := t \rangle]_{(I, e)} = [A]_{(I, e[x=d])}$, où $d = JtK_{(I, e)}$.

Autrement dit, la valeur de $A\langle x := t \rangle$ dans une assignation est la même que celle de A dans une assignation identique, sauf qu'elle donne à x la valeur du terme t . Ce théorème, dont le résultat est évident, peut être prouvé par une récurrence (que nous ne ferons pas) sur la taille des formules. Nous montrons seulement que la condition sur t est indispensable en observant un exemple où cette condition n'est pas respectée.

Exemple 4.3.37 Soient I l'interprétation de domaine $\{0, 1\}$ avec $p_I = \{(0, 1)\}$ et e , un état où $y = 0$. Soient A la formule $\exists y p(x, y)$ et t le terme y . Ce terme n'est pas libre pour x dans A . Nous avons :

— $A\langle x := t \rangle =$

$\exists y p(y, y)$

et $[A\langle x := t \rangle]_{(I, e)} =$

$[\exists y p(y, y)]_{(I, e)} = \max\{[p(0, 0)]_{(I, e)}, [p(1, 1)]_{(I, e)}\} = \max\{0, 0\} = 0.$

— Soit $d = JtK_{(I, e)} = JyK_{(I, e)} = 0$. Dans l'assignation $(I, e[x=d])$, nous avons $x = 0$. Donc, nous obtenons :

$[A]_{(I, e[x=d])} =$

$[\exists y p(x, y)]_{(I, e[x=d])} = \max\{[p(0, 0)]_{(I, e)}, [p(0, 1)]_{(I, e)}\} = \max\{0, 1\} = 1.$

Ainsi, $[A < x := t >]_{(I,e)} \neq [A]_{(I,e[x=d])}$, pour $d = JtK_{(I,e)}$.

Corollaire 4.3.38 Soient A une formule et t un terme libre pour x dans A . Les formules $\forall x A \Rightarrow A < x := t >$ et $A < x := t > \Rightarrow \exists x A$ sont valides.

Ce corollaire est une conséquence immédiate du théorème précédent.

4.3.7 Interprétation finie

Nous montrons comment rechercher des modèles finis de formules fermées (c'est-à-dire sans variables libres). Un modèle fini d'une formule fermée est une interprétation de la formule de domaine fini, qui rend vraie la formule. Il est clair que le nom des éléments du domaine est sans importance, aussi quand nous cherchons un modèle avec un domaine de n éléments, le domaine que nous utiliserons, sera celui des entiers (sous-entendu naturels) inférieurs à n . Pour savoir si une formule fermée a un modèle de domaine $\{0, \dots, n-1\}$, il suffit d'énumérer toutes les interprétations possibles de la signature associée à la formule et d'évaluer la formule pour ces interprétations. Mais cette méthode est inutilisable en pratique, car le nombre de ces interprétations est énorme. Soit une signature avec une constante, un symbole de fonction à un argument et un symbole de relation à deux arguments (plus éventuellement l'égalité de sens fixé), sur un domaine à 5 éléments, cette signature à $5 \times 5^5 \times 2^{25} = 524288000000$ interprétations. Aussi nous montrons d'abord, comment dans le cas où une formule n'a aucun symbole de fonction et aucune constante, sauf des représentations d'entiers inférieurs à n , nous pouvons rechercher ses modèles à n éléments par réduction au cas propositionnel. En présence de symboles de fonctions et de constantes, nous pouvons en énumérer leurs valeurs, puis appliquer les méthodes ci-dessous. Pour trouver des énumérations intelligentes, nous conseillons de lire le manuel de Prover9 [?].

4.3.7.1 Les entiers et leurs représentations

Nous distinguons dans la suite un entier n et sa représentation \underline{n} . Par exemple l'entier 3 peut être représenté en base 10 par 3, en base 2 par 11, en chiffres romains par III, en chiffre grec par γ . Par habitude, nous choisissons la représentation en base 10. Dans la suite, de façon implicite, toutes les interprétations considérées donnent à la représentation d'un entier, la valeur de l'entier représenté.

4.3.7.2 Expansion d'une formule

Dans certains cas il est suffisant de regarder les expansions de taille finie afin de trouver un modèle ou un contre-modèle pour une formule donnée.

Définition 4.3.39 (n -expansion) Soient A une formule et n un entier. La n -expansion de A est la formule qui consiste à remplacer toute sous-formule de A de la forme $\forall x B$ par la conjonction $(\prod_{i < n} B < x := \underline{i} >)$ et toute sous-formule de A de la forme $\exists x B$ par la disjonction $(\sum_{i < n} B < x := \underline{i} >)$ où \underline{i} est la représentation décimale de l'entier i .

Exemple 4.3.40 La 2-expansion de la formule $\exists x P(x) \Rightarrow \forall x P(x)$ est la formule

$$P(0) + P(1) \Rightarrow P(0).P(1).$$

Théorème 4.3.41 Soient n un entier et A une formule ne comportant que des représentations d'entiers de valeur inférieure à n . Soit A' la n -expansion de A . Toute interprétation de domaine $\{0, \dots, n-1\}$ attribue la même valeur à A et à A' .

La condition sur A est nécessaire car si A comporte une représentation d'un entier au moins égal à n , la valeur de cette représentation ne sera pas dans le domaine de l'interprétation. La preuve du théorème est une récurrence sur la taille des formules, que nous ne ferons pas. Nous nous contentons de montrer que l'élimination d'un quantificateur universel de la formule $\forall x B$ conserve la valeur de cette formule.

Soient (I, e) une interprétation et un état de domaine $\{0, \dots, n-1\}$ donnant à la représentation d'un entier, la valeur de l'entier représenté. Par définition du sens du quantificateur universel : $[\forall x B]_{(I,e)} = \prod_{i < n} [B]_{(I,e[x=i])}$. D'après le théorème 4.3.36 page précédente et le fait que la valeur de la représentation de l'entier i est i , nous avons : $[B]_{(I,e[x=i])} = [B < x := \underline{i} >]_{(I,e)}$. Par suite : $[\forall x B]_{(I,e)} = \prod_{i < n} [B < x := \underline{i} >]_{(I,e)} = [\prod_{i < n} B < x := \underline{i} >]_{(I,e)}$.

4.3.7.3 Interprétation et assignation propositionnelle

Soient n un entier et A une formule fermée, sans quantificateur, sans égalité, sans symbole de fonction, sans constante sauf des représentations d'entiers inférieurs à n . Soit P l'ensemble des formules atomiques de A (sauf \top et \perp dont le sens est fixé).

De l'assignation à l'interprétation.

Théorème 4.3.42 *Soit v une assignation propositionnelle de P dans $\{0, 1\}$ alors il existe une interprétation I de A telle que $[A]_I = [A]_v$.*

Preuve : Soit v une assignation propositionnelle de P dans $\{0, 1\}$. Soit I l'interprétation suivante de A :

1. Domaine : $\{0, \dots, n-1\}$
2. $s_I^r = v(s)$ si et seulement si $s \in P$
3. Soit $p \geq 1$ et s^{rp} un symbole de relation de A . Alors $s_I^{rp} = \{(k_1, \dots, k_p) \mid s(k_1, \dots, k_p) \in P \text{ et } v(s(k_1, \dots, k_p)) = 1\}$

Par définition de I , l'assignation v et l'interprétation I donnent la même valeur aux sous-formules atomiques de A , donc (par récurrence sur les formules) la même valeur à la formule A . \square

Exemple 4.3.43 *L'assignation v , définie par $p(0) = 1$ et $p(1) = 0$, donne la valeur 0 à la formule suivante :*

$$(p(0) + p(1)) \Rightarrow (p(0) \cdot p(1)).$$

Donc I définie par $p_I = \{0\}$ donne aussi la valeur 0 à cette même formule. Cet exemple montre que v et I sont deux façons analogues de présenter une interprétation, la deuxième étant souvent plus concise.

De l'interprétation à l'assignation.

Théorème 4.3.44 *Soit I une interprétation de A alors il existe une assignation v de P telle que*

$$[A]_I = [A]_v.$$

Preuve : Soient I une interprétation de A et v l'assignation propositionnelle suivante de P dans $\{0, 1\}$: pour tout $B \in P$, $v(B) = [B]_I$. Par définition de v , l'assignation v et l'interprétation I donnent la même valeur aux sous-formules atomiques de A , donc la même valeur à la formule A . \square

4.3.7.4 Recherche d'un modèle fini d'une formule fermée

Formule fermée sans symbole de fonction : Soit A une formule fermée sans symbole de fonction ni constantes, sauf des représentations d'entiers de valeur inférieure à n . Pour trouver une interprétation I modèle de A de domaine $\{0, \dots, n-1\}$ donnant à la représentation d'un entier, la valeur de l'entier représenté, nous procédons ainsi :

1. Nous remplaçons A par sa n -expansion B .
2. Dans la formule B , nous remplaçons les égalités par leur valeur, c'est-à-dire que $i = j$ est remplacée par 0 si $i \neq j$ et par 1 si $i = j$. De plus, nous recommandons d'éliminer ces valeurs de vérité par les identités $x + 0 = x$, $x + 1 = 1$, $x * 0 = 0$, $x * 1 = x$. Soit C la formule obtenue après ces remplacements et simplifications.
3. Nous cherchons une assignation propositionnelle v des formules atomiques de C , qui soit modèle de C : si une telle assignation n'existe pas, A n'a pas de modèle, sinon l'interprétation I déduite de v comme il est indiqué dans le théorème 4.3.42 page précédente est modèle de A .

Prouvons la correction de cette méthode :

1. Supposons qu'il n'y ait pas d'assignation propositionnelle modèle de C , mais que A ait un modèle I . D'après le théorème 4.3.41 page précédente, I est modèle de B , donc de C , et d'après le théorème 4.3.44, il y a une assignation propositionnelle modèle de C . De cette contradiction, nous déduisons que A n'a pas de modèle à n éléments.
2. Supposons que l'assignation propositionnelle v soit modèle de C . Donc, l'interprétation I construite comme il est indiqué dans le théorème 4.3.42 page précédente est modèle de C , donc elle est modèle de B , donc aussi d'après le théorème 4.3.41 page précédente, elle est modèle de A .

Exemple 4.3.45 Soit A la formule $\exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$. Il est clair que cette formule n'a pas de modèle à un élément car cet élément devrait vérifier à la fois la propriété P et sa négation. La 2-expansion de A est :

$$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)}).(P(0).P(0) \Rightarrow 0 = 0).(P(0).P(1) \Rightarrow 0 = 1).(P(1).P(0) \Rightarrow 1 = 0).(P(1).P(1) \Rightarrow 1 = 1).$$

En remplaçant les égalités par leurs valeurs, nous obtenons :

$$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)}).(P(0).P(0) \Rightarrow 1).(P(0).P(1) \Rightarrow 0).(P(1).P(0) \Rightarrow 0).(P(1).P(1) \Rightarrow 1).$$

Ce qui se simplifie en :

$$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)}).$$

L'assignation $P(0) = 1, P(1) = 0$ est un des modèles propositionnels de la formule ci-dessus, donc l'interprétation I de domaine $\{0, 1\}$ où $P_I = \{0\}$ est modèle de A .

Formule fermée avec symbole de fonction : Soit A une formule fermée pouvant comporter des représentations d'entiers de valeur inférieure à n . Comme dans le cas précédent, nous remplaçons A par son expansion, nous supprimons les égalités. Puis, nous énumérons les choix des valeurs des symboles comme dans l'algorithme $DPLL$, en propageant le plus possible chacun des choix effectués.

Exemple 4.3.46 Soit A la formule $\exists y P(y) \Rightarrow P(a)$, dont nous cherchons un contre-modèle à 2 éléments (cela revient au même que de trouver un modèle de la négation de A).

Nous remplaçons A par son expansion $P(0) + P(1) \Rightarrow P(a)$. La suite est analogue à l'algorithme de Davis et Putnam, nous devons trouver les valeurs de $P(0)$, $P(1)$ et a . Nous choisissons (arbitrairement) de poser $a = 0$, donc A devient $P(0) + P(1) \Rightarrow P(0)$. Cette dernière formule a un contre-modèle propositionnel $P(0) = 0, P(1) = 1$ que nous traduisons en une interprétation telle que $P = \{1\}$. Nous obtenons donc comme contre-modèle l'interprétation de domaine $\{0, 1\}$ telle que $P = \{1\}$ et $a = 0$.

Exemple 4.3.47 Nous cherchons un modèle à 2 éléments des formules $P(a), \forall x (P(x) \Rightarrow P(f(x))), \neg P(f(b))$.

Nous remplaçons les formules par leurs expansions $P(a), (P(0) \Rightarrow P(f(0))), (P(1) \Rightarrow P(f(1))), \neg P(f(b))$. Soit F l'ensemble obtenu. Il faut trouver les valeurs de $P(0), P(1), a, b, f(0)$ et $f(1)$ qui donnent un modèle de F . Nous avons donc : $P(a) = 1, (P(0) \Rightarrow P(f(0))) = 1, (P(1) \Rightarrow P(f(1))) = 1, P(f(b)) = 0$.

Nous choisissons (arbitrairement) de poser $a = 0$. Nous obtenons que :

- De $P(a) = 1$ et $a = 0$, nous déduisons : $P(0) = 1$.
- De $P(0) = 1$ et de $(P(0) \Rightarrow P(f(0))) = 1$, nous déduisons : $P(f(0)) = 1$.
- De $P(f(b)) = 0$ et de $P(f(0)) = 1$, nous déduisons $f(0) \neq f(b)$ donc que $b \neq 0$, donc : $b = 1$.
- De $P(f(b)) = 0, P(0) = 1$ et $b = 1$, nous déduisons $f(b) = f(1) \neq 0$ donc : $f(1) = 1$ et $P(1) = 0$.
- De $P(f(0)) = 1$ et $P(1) = 0$, nous déduisons : $f(0) = 0$.

Ainsi avec un seul choix propagé, nous avons trouvé l'interprétation modèle de domaine $\{0, 1\}$ suivante :

$$a = 0, b = 1, P = \{0\}, f(0) = 0, f(1) = 1.$$

4.3.8 Substitution et remplacement

La propriété de substitution et la propriété de remplacement déjà énoncées en logique propositionnelle (voir 1.3.1 page 20 et 1.3.10 page 22) s'étendent à la logique du premier ordre³. Plus précisément l'application d'une substitution à une formule *propositionnellement* valide donne une formule valide. Par exemple soit σ la substitution propositionnelle telle que $\sigma(p) = \forall x q(x)$. Puisque la formule $p \vee \neg p$ est valide, il en est de même de la formule $\sigma(p \vee \neg p) = \forall x q(x) \vee \neg \forall x q(x)$. Le principe de remplacement s'étend aussi avec le même énoncé de la logique propositionnelle à la logique du premier ordre car il est déduit des propriétés élémentaires suivantes. Pour toutes formules A et B et toute variable x :

- $(A \Leftrightarrow B) \models (\forall x A \Leftrightarrow \forall x B)$.
- $(A \Leftrightarrow B) \models (\exists x A \Leftrightarrow \exists x B)$.

4.4 Équivalences remarquables

Dans cette section, nous donnons les règles permettant de simplifier les formules de la logique du premier ordre. Remarquons que les règles de simplifications de la logique propositionnelle sont bien entendu encore applicables au premier ordre.

4.4.1 Relation entre \forall et \exists

Lemme 4.4.1 Soient A une formule et x une variable.

1. $\neg \forall x A \equiv \exists x \neg A$.
2. $\forall x A \equiv \neg \exists x \neg A$.
3. $\neg \exists x A \equiv \forall x \neg A$.
4. $\exists x A \equiv \neg \forall x \neg A$.

Prouvons les deux premières identités, les deux autres sont données en exercice 76 page 105 :

- 1.

3. Notons que dans le livre de Stephen Cole Kleene [?], cette notion de substitution sur les variables propositionnelles est étendue à une substitution plus générale applicable à tous les symboles de relation.

Soient I une interprétation de domaine D et e un état.

$$\begin{aligned}
 [\neg \forall x A]_{(I,e)} &\equiv 1 - [\forall x A]_{(I,e)} \\
 &\equiv 1 - \prod_{d \in D} [A]_{(I,e[x=d])} && \text{d'après le sens de } \forall \\
 &\equiv \sum_{d \in D} (1 - [A]_{(I,e[x=d])}) && \text{par les lois de De Morgan généralisées} \\
 &\equiv \sum_{d \in D} [\neg A]_{(I,e[x=d])} && \text{par définition du sens de } \neg \\
 &\equiv [\exists x \neg A]_{(I,e)} && \text{par définition du sens de } \exists
 \end{aligned}$$

En examinant de façon critique cette preuve, nous voyons qu'elle utilise la loi de De Morgan généralisée, qui est l'expression dans un autre formalisme d'une loi analogue. Il faut donc formaliser l'usage des quantificateurs pour échapper à ces preuves qui ne sont en fait que des changements de notations. Nous le ferons en ajoutant à la déduction naturelle des règles qui gouvernent les quantificateurs.

2.

Les autres équivalences sont prouvables à partir de la première par le principe de remplacement. Nous ne prouvons que la deuxième identité :

$$\begin{aligned}
 \forall x A &\equiv \neg \neg \forall x A && \text{identité de la double négation} \\
 &\equiv \neg \exists x \neg A && \text{par l'identité 1 page ci-contre}
 \end{aligned}$$

4.4.2 Déplacement des quantificateurs

Nous donnons, sans preuve, des équivalences qui s'appliquent à toutes les formules (avec ou sans variable libre) et qui permettent de retrouver les lois usuelles de déplacement des quantificateurs. Nous supposons ci-dessous que x, y sont des variables et que A, B sont des formules.

1. $\forall x \forall y A \equiv \forall y \forall x A$.
2. $\exists x \exists y A \equiv \exists y \exists x A$.
3. $\forall x (A \wedge B) \equiv (\forall x A \wedge \forall x B)$.
4. $\exists x (A \vee B) \equiv (\exists x A \vee \exists x B)$.
5. Soient Q un des quantificateurs \forall, \exists , et \circ une des opérations $\wedge, \vee, \Rightarrow$. Supposons que x ne soit pas une variable libre de A .
 - (a) $Qx A \equiv A$,
 - (b) $Qx (A \circ B) \equiv (A \circ Qx B)$.

Exemple 4.4.2 Nous éliminons de ces deux formules les quantificateurs inutiles :

$$- \forall x \exists x P(x) \equiv$$

$$\exists x P(x).$$

$$- \forall x (\exists x P(x) \vee Q(x)) \equiv$$

$$\exists x P(x) \vee \forall x Q(x).$$

4.4.3 Changement de variables liées

Théorème 4.4.3 Soit Q un des quantificateurs \forall, \exists . Supposons que y soit une variable qui ne figure pas dans QxA alors : $QxA \equiv QyA[x := y]$.

Exemple 4.4.4 La formule $\forall x p(x, z)$ est équivalente à la formule $\forall y p(y, z)$ (d'après le théorème ci-dessus) mais elle n'est pas équivalente à la formule $\forall z p(z, z)$, où le changement de variable ne respecte pas les conditions du théorème.

Définition 4.4.5 (Formules égales à un changement près de variables liées) Deux formules sont égales à un changement près de variables liées si nous pouvons obtenir l'une à partir de l'autre par des remplacements de sous-formules de la forme QxA par $QyA[x := y]$ où Q est un quantificateur et y est une variable qui ne figure pas dans QxA . Deux formules égales à un changement près de variables liées, sont dites aussi copies l'une de l'autre ou encore α -équivalentes.

Théorème 4.4.6 Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Ce résultat est clairement une conséquence du théorème 4.4.3. Nous n'en ferons pas la preuve, qui est longue et fastidieuse. Nous nous contentons de suggérer cette preuve à l'aide d'un exemple.

Exemple 4.4.7 Nous montrons que les formules $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont égales par changement des variables liées, au sens de la définition 4.4.3 donc sont équivalentes, en effet :

$$\begin{aligned} & \forall x \exists y P(x, y) \\ \equiv & \forall u \exists y P(u, y) \quad \text{d'après le théorème 4.4.3} \\ \equiv & \forall u \exists x P(u, x) \quad \text{d'après le théorème 4.4.3} \\ \equiv & \forall y \exists x P(y, x) \quad \text{d'après le théorème 4.4.3} \end{aligned}$$

Notre définition de l'égalité entre deux formules à un changement près de variables liées n'est pas pratique, car la définition ne donne pas un test. Or il est simple de voir si deux formules sont égales à un changement près de variables liées : nous traçons des traits entre chaque quantificateur et les variables qu'il lie et nous effaçons les noms des variables liées. Si après cette transformation, les deux formules deviennent identiques, c'est qu'elles sont égales à un changement près des variables liées.

Exemple 4.4.8 Les deux formules $\forall x \exists y P(y, x)$ et $\forall y \exists x P(x, y)$ sont transformées en $\forall | \exists | P(|, |)$.

Notons $=_\alpha$ la relation d'égalité entre deux formules à un changement près des variables liées. Nous donnons sans preuve des propriétés de cette relation.

Théorème 4.4.9 Les majuscules désignent des formules et les minuscules sont des variables.

1. Soit A une formule atomique, $A =_\alpha A'$ si et seulement si $A' = A$
2. $\neg B =_\alpha A'$ si et seulement si $A' = \neg B'$ et $B =_\alpha B'$
3. $(B \circ C) =_\alpha A'$ si et seulement si $A' = (B' \circ C')$ et $B =_\alpha B'$ et $C =_\alpha C'$. où \circ est l'un des connecteurs $\wedge, \vee, \Rightarrow, \Leftrightarrow$.
4. Si $\forall x B =_\alpha A'$ alors $A' = \forall x' B'$ et pour toute variable z absente des formules B et B' , nous avons : $B[x := z] =_\alpha B'[x' := z]$.
5. Si $\exists x B =_\alpha A'$ alors $A' = \exists x' B'$ et pour toute variable z absente des formules B et B' , nous avons : $B[x := z] =_\alpha B'[x' := z]$.

6. S'il existe une variable z absente des formules B et B' telle que $B \langle x := z \rangle =_\alpha B' \langle x' := z \rangle$ alors $\forall x B =_\alpha \forall x' B'$ et $\exists x B =_\alpha \exists x' B'$.

Ce théorème permet facilement d'écrire un algorithme pour tester cette relation.

Exemple 4.4.10 Nous donnons un algorithme pour le test de l'alpha-équivalence, ceci uniquement pour le cas où $A = \forall x B$. Les données du test sont deux formules A et A' . Le résultat est oui si $A =_\alpha A'$, non si $A \neq_\alpha A'$.

1. Si A' n'est pas de la forme $\forall x' B'$, alors, d'après le point (4) du théorème, la réponse est non.
2. Si $A' = \forall x' B'$ alors nous choisissons une variable z quelconque absente de B et B' .
 - (a) Si $B \langle x := z \rangle =_\alpha B' \langle x' := z \rangle$ alors, d'après point (6) du théorème, la réponse est oui.
 - (b) Si $B \langle x := z \rangle \neq_\alpha B' \langle x' := z \rangle$ alors, d'après le point (4) du théorème la réponse est non.

4.5 Exercices

Exercice 62 (Structure et variables libres)

Pour chaque formule ci-dessous, indiquer sa structure et ses variables libres.

1. $\forall x(P(x) \Rightarrow \exists y Q(x, y))$.
2. $\forall a \forall b(b \neq 0 \Rightarrow \exists q \exists r(a = b * q + r \wedge r < b))^4$.
3. $\text{Pair}(x) \Leftrightarrow \exists y(x = 2 * y)$.
4. $x \text{ Divise } y \Leftrightarrow \exists z(y = z * x)$.
5. $\text{Premier}(x) \Leftrightarrow \forall y(y \text{ Divise } x \Rightarrow y = 1 \vee y = x)$.

□

Exercice 63 (Formalisation, symbole de fonction et de relation) Nous considérons $\Sigma = \{f^{r2}, o^{r2}, c^{r2}, j^{r2}, r^{f0}, p^{f1}\}$ la signature ayant la sémantique donnée ci-dessous.

- $f(x, y)$: x est frère de y .
- $o(x, y)$: x est l'oncle de y .
- $c(x, y)$: x est le cousin de y .
- $j(x, y)$: x est plus jeune que y .
- r est le diminutif de Robert.
- $p(x)$ est le père de x .

Exprimer en logique du premier ordre et en utilisant la signature Σ les phrases suivantes :

1. Tout frère du père de Robert est un oncle de Robert.
2. Si les pères de deux enfants sont des frères alors ces deux enfants sont des cousins.
3. Robert a un cousin plus jeune qu'un des frères de Robert.

Exprimer en français les propositions logiques suivantes :

1. $\nexists x j(p(x), x)$
2. $\forall x \forall y(p(p(x)) = p(p(y)) \Rightarrow c(x, y))$
3. $\forall x \exists y(f(x, y) \wedge j(x, y))$
4. $\exists x \exists y(f(x, y) \wedge \neg(p(x) = p(y)))$

□

Exercice 64 (Formalisation) Considérons la signature $\Sigma = \{a^{f0}, f^{f0}, J^{r2}, G^{r2}\}$, où les symboles ont le sens donné ci-dessous.

- a : l'équipe d'Allemagne.
- f : l'équipe de France.
- $J(x, y)$: x a joué un match contre y .
- $G(x, y)$: x a gagné contre y .

Exprimer en logique du premier ordre en utilisant la signature Σ les assertions suivantes :

1. L'équipe de France a gagné un match et en a perdu un.
2. L'équipe de France et l'équipe d'Allemagne ont fait match nul.
3. Une équipe a gagné tous ses matchs.
4. Aucune équipe n'a perdu tous ses matchs.
5. Considérons l'assertion suivante : « Tous ceux qui ont joué contre une équipe qui a gagné tous ses matchs, ont gagné au moins un match ». Parmi les formules suivantes, lesquelles expriment la phrase ci-dessus, et lesquelles sont équivalentes ?

- (a) $\forall x \exists y(J(x, y) \wedge \forall z(J(y, z) \Rightarrow G(y, z)) \Rightarrow \exists v G(x, v))$.
- (b) $\forall x(\exists y(J(x, y) \wedge \forall z(J(y, z) \Rightarrow G(y, z))) \Rightarrow \exists v G(x, v))$.
- (c) $\exists x(\forall y(J(x, y) \Rightarrow G(x, y)) \Rightarrow \forall z(J(x, z) \Rightarrow \exists v G(x, v)))$.
- (d) $\forall x \forall y(J(x, y) \wedge \forall z(J(y, z) \Rightarrow G(y, z)) \Rightarrow \exists v G(x, v))$.
- (e) $\forall x(\forall y(J(x, y) \wedge \forall z(J(y, z) \Rightarrow G(y, z))) \Rightarrow \exists v G(x, v))$.

4. Afin de respecter la notation usuelle pour la division euclidienne nous prenons exceptionnellement a, b, q et r comme nom de variable.

□

Exercice 65 (Formaliser,)** Soient les constantes s pour Serge, t pour Toby et les symboles de relation $A(x,y)$, x aime y , $C(x)$, x est un chien, $D(x)$, x est un animal domestique, $E(x)$, x est un enfant, $O(x)$, x est un oiseau et $P(x,y)$, x a peur de y , donner la signature Σ associée à ces symboles. Formaliser en logique du premier ordre en utilisant Σ les énoncés suivants :

1. Les chiens et les oiseaux sont des animaux domestiques.
2. Toby est un chien qui aime les enfants.
3. Les oiseaux n'aiment pas les chiens.
4. Serge aime tous les animaux domestiques sauf les chiens.
5. Tous les enfants n'ont pas peur des chiens.
6. Certains chiens aiment les enfants.
7. Certains chiens aiment les enfants et réciproquement.
8. Les enfants aiment certains chiens.

□

Exercice 66 (Évaluer des prédicats unaires) Soit I l'interprétation de domaine $D = \{0, 1\}$ telle que $P_I = \{0\}$, $Q_I = \{1\}$.

1. Évaluer dans cette interprétation les formules $\forall x P(x)$ et $\forall x (P(x) \vee Q(x))$.
2. Les formules $\forall x P(x) \vee \forall x Q(x)$ et $\forall x (P(x) \vee Q(x))$ sont-elles équivalentes ?
3. Évaluer dans cette interprétation les formules $\exists x P(x)$ et $\exists x (P(x) \wedge Q(x))$.
4. Les formules $\exists x P(x) \wedge \exists x Q(x)$ et $\exists x (P(x) \wedge Q(x))$ sont-elles équivalentes ?
5. Évaluer dans cette interprétation les formules $\forall x (P(x) \Rightarrow Q(x))$ et $\forall x P(x) \Rightarrow \forall x Q(x)$.
6. Les deux formules $\forall x (P(x) \Rightarrow Q(x))$ et $\forall x P(x) \Rightarrow \forall x Q(x)$ sont-elles équivalentes ?

□

Exercice 67 (Interprétation,*) Nous donnons les formules suivantes :

1. $\forall x \exists y (y = x + 1)$.
2. $\exists y \forall x (y = x + 1)$.
3. $\forall x \exists y (y = x + 1) \Rightarrow \exists y \forall x (y = x + 1)$.
4. $\forall x \exists y (x = y + 1)$.
5. $\exists x \forall y (y = x + y)$.
6. $\exists x (x \neq 0 \wedge x + x = x)$.

Nous donnons les interprétations suivantes :

1. I_1 est l'algèbre de Boole sur $\{0, 1\}$.
2. I_2 est l'arithmétique usuelle sur les entiers naturels.
3. I_3 est l'arithmétique usuelle sur les entiers relatifs.
4. I_4 est l'algèbre de Boole de domaine $\mathcal{P}(X)$ où les constantes 0 et 1 dénotent les ensembles \emptyset et X et où l'addition est l'union d'ensembles.

Indiquer si ces interprétations sont des modèles ou des contre-modèles des formules ci-dessus.

□

Exercice 68 (Prédicat unaire et égalité) Soit I l'interprétation de domaine $D = \{0, 1, 2\}$ telle que : $P_I = \{0, 1\}$, $Q_I = \{1, 2\}$, $R_I = \{\}$. Évaluer dans cette interprétation les formules suivantes :

1. $\exists x R(x)$.
2. $\forall x (P(x) \vee Q(x))$.
3. $\forall x (P(x) \Rightarrow Q(x))$.
4. $\forall x (R(x) \Rightarrow Q(x))$.
5. $\exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$ ⁵.

5. Cette formule signifie qu'il y a un et un seul élément vérifiant P .

$$6. \exists x(P(x) \wedge Q(x) \wedge \forall y(P(y) \wedge Q(y) \Rightarrow x = y)).$$

□

Exercice 69 (Évaluation, égalité) Nous utilisons le symbole de fonction unaire f et la constante a . Nous abrégons $\neg(x = y)$ en $x \neq y$. Soient I_1, I_2 les interprétations suivantes de domaine $\{0, 1, 2\}$: $a_{I_1} = a_{I_2} = 0$.

x	$f_{I_1}(x)$	$f_{I_2}(x)$
0	0	1
1	0	2
2	2	0

Dans l'interprétation I_1 puis dans I_2 , évaluer les formules suivantes :

1. $f(a) = a$.
2. $f(f(a)) = a$.
3. $f(f(f(a))) = a$.
4. $\exists x(f(x) = x)$, f a un point-fixe.
5. $\forall x(f(f(f(x)))) = x$.
6. $\forall y \exists x(f(x) = y)$, f est surjective.
7. $\forall x \forall y(f(x) = f(y) \Rightarrow x = y)$, f est injective.
8. $\neg(\exists x \exists y(f(x) = f(y) \wedge x \neq y))$.

□

Exercice 70 (Formalisation et évaluation) Nous adoptons les notations suivantes :

- Anatoli, Boris, Catarina et Denka sont des constantes du domaine,
- $P(x)$ signifie que x a réussi son examen,
- $Q(x, y)$ signifie que x a téléphoné à y .

Donner la signature associée à ces notations et traduire en formules les énoncés suivants :

1. Quelqu'un a raté l'examen et n'a été appelé par personne.
2. Tous ceux qui ont réussi à l'examen ont été appelés.
3. Personne n'a appelé tous ceux qui ont réussi à l'examen.
4. Tous ceux qui ont appelé quelqu'un, ont appelé quelqu'un qui a réussi l'examen.

Soit l'interprétation ayant pour domaine $D = \{0, 1, 2, 3\}$. Anatoli, Boris, Catarina et Denka valent respectivement 0, 1, 2 et 3 dans l'interprétation. Anatoli et Boris sont des garçons et Catarina et Denka sont des filles. Dans cette interprétation seuls Boris et Catarina ont réussi l'examen, les garçons ont appelé les filles, Denka a appelé Boris, Catarina a appelé Denka et ce sont les seuls appels.

Nous demandons de définir formellement l'interprétation donnée ci-dessus, puis de donner la valeur des énoncés précédents dans cette interprétation.

Indication : pour faciliter le calcul de la valeur, nous suggérons de dessiner l'interprétation en entourant les prénoms des personnes qui ont réussi leurs examens, et en mettant une flèche de x vers y si x a téléphoné à y . □

Exercice 71 (Expansion et contre-modèle) Trouver, par la méthode des expansions, des contre-modèles des formules suivantes :

1. $\exists x P(x) \Rightarrow \exists x(P(x) \wedge Q(x))$.
2. $\forall x(P(x) \Rightarrow Q(x)) \Rightarrow \exists x Q(x)$.
3. $\forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\exists x P(x) \Rightarrow \forall x Q(x))$.
4. $(\exists x F(x) \Rightarrow \exists x G(x)) \Rightarrow \forall x(F(x) \Rightarrow G(x))$.
5. $\forall x \exists y R(x, y) \Rightarrow \exists x R(x, x)$.
6. $\forall x \forall y(R(x, y) \Rightarrow R(y, x)) \Rightarrow \forall x R(x, x)$.

Indication : il suffit de construire des 1 ou 2 expansions. □

Exercice 72 (Raisonnement incorrect) Considérons les hypothèses suivantes :

1. $\exists x P(x)$.
2. $\exists x Q(x)$.
3. $\forall x (P(x) \wedge Q(x) \Rightarrow R(x))$.

Montrer qu'il est incorrect de déduire à partir de ces trois hypothèses la conclusion suivante : $\exists x R(x)$. □

Exercice 73 (Contre-modèles avec relation) Construire des contre-modèles des formules suivantes, où F est une relation :

1. $\forall x \exists y (x = y) \Rightarrow \exists y \forall x (y = x)$.
 2. $F(a) \wedge (a \neq b) \Rightarrow \neg F(b)$.
 3. $\exists x \exists y (F(x) \wedge F(y) \wedge x \neq y) \Rightarrow \forall x F(x)$.
 4. $\forall x \forall y (F(x, y) \Rightarrow x = y) \Rightarrow \exists x F(x, x)$.
-

Exercice 74 (Contre-modèles avec fonction) Construire des contre-modèles des formules suivantes, où f est une fonction et P une relation :

1. $\forall y \exists x (f(x) = y)$.
 2. $\forall x \forall y (f(x) = f(y) \Rightarrow x = y)$.
 3. $\exists x \forall y (f(y) = x)$.
 4. $\forall x (P(x) \Rightarrow P(f(x)))$.
-

Exercice 75 (Équivalences) Prouver que :

1. $\neg \forall x \exists y P(x, y) \equiv \exists y \forall x \neg P(y, x)$.
2. $\exists x (P(x) \Rightarrow Q(x)) \equiv \forall x P(x) \Rightarrow \exists x Q(x)$.
3. La phrase « aucun malade n'aime les charlatans » a été traduite en logique du premier ordre par deux étudiants par les 2 formules suivantes :
 - $\forall x \forall y ((M(x) \wedge A(x, y)) \Rightarrow \neg C(y))$.
 - $\neg (\exists x (M(x) \wedge (\exists y (A(x, y) \wedge C(y))))$.

Montrer que ces étudiants disent la même chose, c'est-à-dire les deux formules associées aux traductions sont équivalentes. □

Exercice 76 (\Leftrightarrow , Preuve,*) Prouver les deux équivalences suivantes du lemme 4.4.1 page 98 :

- $\neg \exists x A \equiv \forall x \neg A$.
- $\exists x A \equiv \neg \forall x \neg A$.

Conseil : utiliser les propriétés 1 page 98 et 2 page 98 du lemme 4.4.1 page 98. □

Exercice 77 (Preuve) Nous savons que $(\forall x (A \wedge B)) \equiv (A \wedge (\forall x B))$ à la condition que x ne soit pas une variable libre de A comme il est dit au paragraphe 4.4.2 page 99. Montrer que cette condition est nécessaire en donnant une assignation qui donne des valeurs différentes aux deux formules $\forall x (P(x) \wedge Q(x))$ et $P(x) \wedge (\forall x Q(x))$. □

Chapitre 5

Base de la démonstration automatique

Sommaire

5.1	Méthodes de Herbrand	107
5.1.1	Domaine et base de Herbrand	107
5.1.2	Interprétation de Herbrand	108
5.1.3	Théorème de Herbrand	111
5.2	Skolémisation	113
5.2.1	Algorithme de skolémisation d'une formule	113
5.2.2	Propriétés de la forme de Skolem	117
5.2.3	Forme clausale	118
5.3	Unification	120
5.3.1	Unificateur	120
5.3.2	Algorithme d'unification	122
5.4	Résolution au premier ordre	124
5.4.1	Trois règles pour la résolution	124
5.4.2	Cohérence de la résolution	126
5.4.3	Complétude de la résolution	127
5.5	Exercices	131

Il est important de préciser que la logique du premier ordre est *indécidable*, cela signifie qu'il n'existe pas d'algorithme pour *déterminer* si une formule est valide ou non valide. Ce résultat fut établi par Alonzo Church et Alan Turing en 1936 et 1937 [?, ?]. Ils ont aussi montré que cette logique est *semi-décidable*, c'est-à-dire que nous pouvons écrire un programme, qui prend en entrée une formule et qui a le comportement suivant :

1. S'il termine alors il *décide correctement* si la formule est valide ou non. Lorsque la formule est valide, la décision est généralement accompagnée d'une preuve.
2. Si la formule est valide, alors il termine. Cependant, l'exécution peut être longue !

Notons que *si la formule n'est pas valide, la terminaison de ce programme n'est pas garantie*. En effet si le programme terminait pour toute formule, alors, d'après le premier point, ce serait un algorithme pour déterminer si une formule est valide ou non. Or, ainsi qu'il est dit ci-dessus, Church et Turing ont prouvé qu'un tel algorithme n'existait pas.

Plan : Dans ce chapitre nous commençons par présenter de méthodes introduites par Jacques Herbrand, permettant d'énumérer un ensemble fini de formules afin de trouver un modèle à une formule. Ensuite nous présentons la transformation due à Thoralf Albert Skolem qui transforme (skolémise) une formule en une « forme de Skolem » où l'existence d'un modèle est préservée. En transformant ensuite cette forme de Skolem en une « forme clausale » équivalente, nous pouvons appliquer la résolution. Nous reprenons alors le concept de résolution introduit dans le chapitre 2 page 41 et nous l'étudions en logique du premier ordre. En 1929, Kurt Gödel (1906 - 1978) prouva que la logique du premier ordre est complète et cohérente. Nous présentons ce résultat pour la résolution, c'est-à-dire qu'il existe une preuve par résolution en logique du premier ordre d'une formule F à partir d'un ensemble de formules Γ si et seulement si F est conséquence de Γ .

5.1 Méthodes de Herbrand

Jacques Herbrand (1908-1931) est un mathématicien et logicien français. Il a établi en 1930 un lien entre calcul des prédicats et calcul des propositions.

5.1.1 Domaine et base de Herbrand

Nous introduisons d'abord la notion de fermeture universelle qui permet de transformer toute formule en une formule fermée où la propriété de satisfiabilité est maintenue.

Définition 5.1.1 (Fermeture universelle) Soit C une formule ayant pour variables libres x_1, \dots, x_n . La fermeture universelle de C , notée $\forall(C)$, est la formule $\forall x_1 \dots \forall x_n C$. Cette notion est définie à l'ordre près des variables libres de C . Soit Γ un ensemble de formules, $\forall(\Gamma) = \{\forall(A) \mid A \in \Gamma\}$.

Exemple 5.1.2 La fermeture universelle de $P(x) \wedge R(x, y)$ est

$$\text{soit } \forall x \forall y (P(x) \wedge R(x, y)), \text{ soit } \forall y \forall x (P(x) \wedge R(x, y)).$$

Nous généralisons la définition 1.3.1 page 20 aux termes de la logique du premier ordre.

Définition 5.1.3 (Substitution) Une substitution est une application des variables dans les termes. Soient A une formule et σ une substitution. $A\sigma$ est la formule obtenue en remplaçant toute occurrence libre d'une variable par son image dans l'application. La formule $A\sigma$ est une instance de A .

Dans la suite de ce chapitre, nous ne considérons que des formules qui ne contiennent ni le symbole égal, ni les constantes propositionnelles \top et \perp , car leur sens est fixé dans toute interprétation : \top vaut 1, \perp vaut 0, et le symbole $=$ est l'identité sur le domaine de l'interprétation (voir Remarque 4.3.18 page 90). De plus nous considérons que toute signature comporte au moins une constante. Ainsi la signature d'un ensemble de formules ne comportant pas de constante, sera arbitrairement complétée par la constante a .

Définition 5.1.4 (Domaine et base de Herbrand)

1. Le domaine de Herbrand pour Σ est l'ensemble des termes fermés (i.e., sans variable) de cette signature, dénoté par D_Σ .
2. La base de Herbrand pour Σ est l'ensemble des formules atomiques fermées de cette signature, nous la notons B_Σ .

Notons que le domaine de Herbrand d'une signature sans constante est vide. Puisque toutes les signatures considérées dans ce chapitre doivent comporter au moins une constante, ce domaine n'est pas vide.

Exemple 5.1.5 Calculs de base de Herbrand :

1. Soit Σ la signature comportant uniquement les 2 constantes a, b et les 2 symboles de relations unaires P, Q . Nous avons $D_\Sigma = \{a, b\}$ et $B_\Sigma =$

$$\{P(a), P(b), Q(a), Q(b)\}.$$

2. Soit Σ la signature comportant uniquement la constante a , le symbole de fonction unaire f et le symbole de relation unaire P . Nous avons $D_\Sigma = \{f^n(a) \mid n \in \mathbb{N}\}$ et $B_\Sigma =$

$$\{P(f^n(a)) \mid n \in \mathbb{N}\}.$$

5.1.2 Interprétation de Herbrand

À partir de la base et du domaine de Herbrand d'une formule, nous sommes capables de construire une interprétation dite de « Herbrand ».

Définition 5.1.6 (Interprétation de Herbrand) Soient Σ une signature et $E \subseteq B_\Sigma$. L'interprétation de Herbrand $H_{\Sigma,E}$ a pour domaine D_Σ et donne aux symboles le sens suivant :

1. Si le symbole s est une constante de la signature, il vaut lui-même dans cette interprétation.
2. Si s est un symbole de fonction à $n \geq 1$ arguments de la signature et si $t_1, \dots, t_n \in D_\Sigma$ alors $s_{H_{\Sigma,E}}^{fn}(t_1, \dots, t_n) = s(t_1, \dots, t_n)$.
3. Si le symbole s est une variable propositionnelle, il vaut 1, autrement dit il est vrai, si et seulement si $s \in E$.
4. Si s est un symbole de relation de la signature à $n \geq 1$ arguments et si $t_1, \dots, t_n \in D_\Sigma$ alors $s_{H_{\Sigma,E}}^{rn} = \{(t_1, \dots, t_n) \mid t_1, \dots, t_n \in D_\Sigma \wedge s(t_1, \dots, t_n) \in E\}$.

Le résultat suivant montre qu'une interprétation de Herbrand peut être identifiée à l'ensemble des formules atomiques fermées dont elle est modèle.

Propriété 5.1.7 (Propriété d'une interprétation de Herbrand) Soient Σ une signature et $E \subseteq B_\Sigma$. Dans l'interprétation de Herbrand $H_{\Sigma,E}$:

1. La valeur d'un terme sans variable est lui-même.
2. L'interprétation est modèle d'une formule atomique fermée si et seulement si elle est élément de E .

La preuve est une conséquence directe de la définition d'une interprétation de Herbrand.

Notons ici, avec un exemple, pourquoi on a supposé que les formules ne contiennent pas les symboles de relation $\top, \perp, =$, dont le sens est fixé dans toutes les interprétations. Supposons au contraire que \top soit élément de la base mais non élément de E . D'après le point 2, l'interprétation $H_{\Sigma,E}$ donnerait \top la valeur 0, alors que \top vaut 1 dans toute interprétation.

Exemple 5.1.8 Soit Σ la signature comportant les 2 constantes a, b et les 2 symboles de relations unaires P, Q . L'ensemble $E = \{P(b), Q(a)\}$ définit l'interprétation de Herbrand $H_{\Sigma,E}$ de domaine

$$D_\Sigma = \{a, b\} \text{ où les constantes } a \text{ et } b \text{ ont pour valeur elles-mêmes et où } P_{H_{\Sigma,E}} = \{b\} \text{ et } Q_{H_{\Sigma,E}} = \{a\}.$$

Lemme 5.1.9 Soit I une interprétation de la signature Σ . Soit A une formule sur cette signature. L'interprétation I est modèle de $\forall(A)$ si et seulement si pour tout état e de cette l'interprétation, l'assignation (I, e) est modèle de A .

Preuve : Ce lemme est une conséquence directe du sens de \forall . Soit x_1, \dots, x_n une liste des variables libres de A . Nous rappelons que $\forall(A) = \forall x_1 \dots \forall x_n A$.

1. Supposons que l'interprétation I est modèle de $\forall(A)$. Montrons que pour tout état e , l'assignation (I, e) est modèle de A . Soit e un état quelconque. Nous prouvons, par récurrence sur i que (I, e) est modèle de $\forall x_i \dots \forall x_n A$.
 - (a) Cas $i = 1$. I étant modèle de la formule fermée $\forall(A)$, alors pour tout état e , (I, e) est modèle de $\forall(A)$, ce qui établit la propriété pour $i = 1$.
 - (b) Supposons que (I, e) est modèle de $\forall x_i \dots \forall x_n A$. Montrons que (I, e) est modèle de $\forall x_{i+1} \dots \forall x_n A$.
D'après l'hypothèse de récurrence ci-dessus et le sens de \forall , pour tout état f différent de e au plus par la valeur de x_i , (I, f) est modèle de $\forall x_{i+1} \dots \forall x_n A$. Puisque e est un tel état, (I, e) est modèle de $\forall x_{i+1} \dots \forall x_n A$.
En appliquant la propriété établie ci-dessus pour tout $i \geq 1$ au cas où $i = n + 1$, nous avons (I, e) modèle de $\forall x_{n+1} \dots \forall x_n A$, autrement dit (I, e) est modèle de A .
2. Supposons que pour tout état e , l'assignation (I, e) est modèle de A .
Nous prouvons, par récurrence sur i , que, pour tout état e , (I, e) est modèle de $\forall x_{n+1-i} \dots \forall x_n A$.
 - (a) Cas $i = 0$. Puisque $\forall x_{n+1} \dots \forall x_n A = A$, par hypothèse, la propriété est vérifiée pour $i = 0$.

- (b) Supposons que pour tout état e , (I, e) est modèle de $\forall x_{n+1-i} \dots \forall x_n A$. Montrons que pour tout état e , (I, e) est modèle de $\forall x_{n+1-(i+1)} \dots \forall x_n A$. Notons que $n+1-(i+1) = n-i$.

Soit e un état *quelconque*. D'après le sens de \forall , (I, e) est modèle de $\forall x_{n-i} \dots \forall x_n A$ si et seulement si, pour tout état f différent de e au plus par la valeur de x_{n-i} , (I, f) modèle de $\forall x_{n+1-i} \dots \forall x_n A$. Par hypothèse de récurrence, cette dernière propriété est vérifiée. Donc, puisque l'état e est quelconque, pour tout état g , (I, g) est modèle de $\forall x_{n-i} \dots \forall x_n A$.

En appliquant la propriété établie ci-dessus pour tout i , au cas où $i = n$, nous avons : pour tout état e , (I, e) est modèle de $\forall(A)$. Puisque cette formule est sans variable libre, sa valeur ne dépend pas de l'état e , donc I est modèle de $\forall(A)$. □

Lemme 5.1.10 *Soit A une formule sans quantificateur sur la signature Σ . Soit σ une substitution pour cette signature, autrement dit une application des variables dans les termes sur cette signature. Soit I une interprétation de la signature Σ et e un état de cette interprétation. Nous avons : $[A\sigma]_{(I,e)} = [A]_{(I,f)}$ où pour toute variable x , $f(x) = J\sigma(x)K_{(I,e)}$, autrement dit $f(x)$ est la valeur du terme $\sigma(x)$ dans l'assignation (I, e) .*

Preuve : La preuve de ce lemme est longue et sans intérêt, c'est pourquoi nous l'avons omise. En fait, elle consiste à prouver par récurrence les deux faits suivants :

1. Pour tout terme t , $Jt\sigma K_{(I,e)} = JtK_{(I,f)}$.
 2. Pour toute formule A sans quantificateur, $[A\sigma]_{(I,e)} = [A]_{(I,f)}$.
-

Lemme 5.1.11 *Soit A une formule sans quantificateur sur la signature Σ . Soit I une interprétation de cette signature. Si I est modèle de $\forall(A)$, alors, pour tout état e , l'assignation (I, e) est modèle de toute instance de A .*

Preuve : Soit B une instance de A sur Σ . Par définition des instances, il existe une substitution σ sur la signature Σ , telle que $B = A\sigma$. Soient e un état *quelconque* et f l'état où pour toute variable x , $f(x)$ est la valeur du terme $\sigma(x)$ dans l'assignation (I, e) . Supposons que I est modèle de $\forall(A)$. D'après le lemme 5.1.9 page précédente, (I, f) est modèle de A . D'après le lemme 5.1.10 page précédente (I, e) est modèle de B . □

Lemme 5.1.12 *Soit A une formule sans quantificateur ni variable sur la signature Σ . Soit I une interprétation de cette signature. Soit E l'ensemble des éléments de B_Σ , la base de Herbrand pour Σ , dont I est modèle. L'interprétation de Herbrand $H_{\Sigma,E}$ est modèle de A si et seulement si I est modèle de A .*

Preuve : Nous faisons la preuve par récurrence sur le nombre de connecteurs de A . Supposons que pour toute formule A sans quantificateur ni variable sur la signature Σ ayant moins de n connecteurs, l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle de A si et seulement si l'interprétation I est modèle de A .

Soit A une formule sans quantificateur ni variable sur la signature Σ avec n connecteurs.

1. Supposons $n = 0$. Alors A est dans la base de Herbrand pour Σ . Par définition, I est modèle de A si et seulement si $A \in E$. Et d'après la propriété 5.1.7 page 108, I est modèle de A si et seulement si $H_{\Sigma,E}$ est modèle de A .
 2. Supposons $n > 0$. Alors A est de l'une des formes $\neg B, (B \wedge C), (B \vee C), (B \Rightarrow C), (B \Leftrightarrow C)$ où B et C sont des formules avec moins de n connecteurs. Nous n'examinons que le cas $A = (B \wedge C)$, car les autres cas sont analogues. Supposons que I est modèle de A , alors I est modèle de B et de C . Par hypothèse de récurrence, $H_{\Sigma,E}$ est modèle de B et de C , donc aussi modèle de A . La réciproque est analogue.
-

Définition 5.1.13 (Assignation propositionnelle caractéristique) *Soit Σ une signature. Soit E un sous-ensemble de B_Σ la base de Herbrand pour Σ . L'assignation propositionnelle de domaine B_Σ caractéristique de E est l'application $v : B_\Sigma \rightarrow \{0, 1\}$ qui vérifie pour tout $A \in B_\Sigma$, $v(A) = 1$ si et seulement si $A \in E$.*

Lemme 5.1.14 *Soit A une formule sans quantificateur ni variable sur la signature Σ . Soit E un sous-ensemble de B_Σ la base de Herbrand pour Σ . Soit v l'assignation propositionnelle de domaine B_Σ caractéristique de E . L'interprétation de Herbrand $H_{\Sigma,E}$ est modèle du premier ordre de A si et seulement si v est modèle propositionnel de A .*

Preuve : La preuve est analogue à la précédente, par récurrence sur le nombre de connecteurs de A . Aussi nous ne traitons que le cas où A ne comporte *aucun connecteur*. La formule A est alors élément de la base de Herbrand pour Σ .

D'après la propriété 5.1.7 page 108, $H_{\Sigma,E}$ est modèle de A si et seulement si $A \in E$. Par définition de v , $A \in E$ si et seulement si $v(A) = 1$. Donc $H_{\Sigma,E}$ est modèle de A si et seulement si v est modèle de A . \square

Soit e un état de l'interprétation de Herbrand $H_{\Sigma,E}$. Par définition d'une telle interprétation, pour toute variable x , $e(x)$ est un élément de D_Σ l'ensemble des termes fermés sur la signature Σ . L'état e est aussi une substitution pour Σ . Ainsi, nous pouvons appliquer la substitution e à toute formule A sur la signature Σ . Nous rappelons qu'appliquer la substitution e à la formule A est notée Ae .

Lemme 5.1.15 *Soit e un état de l'interprétation de Herbrand $H_{\Sigma,E}$. Soit A une formule sans quantificateur sur Σ . L'interprétation $H_{\Sigma,E}$ est modèle de Ae si et seulement si l'assignation $(H_{\Sigma,E}, e)$ est modèle de A .*

Preuve : Soit f un état quelconque de $H_{\Sigma,E}$.

D'après le lemme 5.1.10 page précédente, l'assignation $(H_{\Sigma,E}, f)$ est modèle de Ae si et seulement si l'assignation $(H_{\Sigma,E}, g)$ est modèle de A où, pour toute variable x , $g(x)$ est la valeur de $e(x)$ dans l'assignation $(H_{\Sigma,E}, f)$. Puisque Ae est une formule sans variable, l'assignation $(H_{\Sigma,E}, f)$ est modèle de Ae si et seulement si l'interprétation $H_{\Sigma,E}$ est modèle de Ae . Pour toute variable x , $e(x)$ est un terme sans variable. Donc, d'après la propriété 5.1.7 page 108, la valeur de $e(x)$ dans une assignation de Herbrand est $e(x)$ et par suite $g = e$. Par suite l'interprétation $H_{\Sigma,E}$ est modèle de Ae si et seulement si l'assignation $(H_{\Sigma,E}, e)$ est modèle de A . \square

Théorème 5.1.16 (Fermeture universelle et modèle de Herbrand) *Soit Γ un ensemble de formules sans quantificateur sur la signature Σ . La fermeture universelle $\forall(\Gamma)$ a un modèle si et seulement si $\forall(\Gamma)$ a un modèle qui est une interprétation de Herbrand de Σ .*

Preuve : L'implication de droite à gauche est trivial. Prouvons l'implication de gauche à droite. Soit I une interprétation de Σ , qui est modèle de $\forall(\Gamma)$. Soit E l'ensemble des éléments de B_Σ , dont I est modèle.

Nous montrons que l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle $\forall(\Gamma)$. Soit A une formule quelconque de Γ . Puisque $A \in \Gamma$, I est modèle de $\forall(A)$. Soit e un état quelconque de $H_{\Sigma,E}$. D'après le lemme 5.1.11 page ci-contre et le fait que Ae est une formule sans quantificateur ni variable, I est modèle de Ae . D'après le lemme 5.1.12 page précédente, $H_{\Sigma,E}$ est aussi modèle de Ae . D'après le lemme 5.1.15 page ci-contre, l'assignation $(H_{\Sigma,E}, e)$ est modèle de A . Puisque e est un état quelconque de $H_{\Sigma,E}$, d'après le lemme 5.1.9 page 109, l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle de $\forall(A)$. Puisque A est une formule quelconque de Γ , $H_{\Sigma,E}$ est modèle de $\forall(\Gamma)$. \square

Ce théorème est appliqué dans la preuve du théorème 5.2.17 page 118.

5.1.3 Théorème de Herbrand

Le théorème de Herbrand établit un lien entre la logique du premier ordre et la logique propositionnelle.

Théorème 5.1.17 (Théorème de Herbrand) *Soit Γ un ensemble de formules sans quantificateur de signature Σ . $\forall(\Gamma)$ a un modèle si et seulement si tout ensemble fini d'instances fermées sur la signature Σ des formules de Γ a un modèle propositionnel application de la base de Herbrand B_Σ dans l'ensemble $\{0, 1\}$.*

Preuve :

\Rightarrow Supposons que $\forall(\Gamma)$ a un modèle I , qui est une interprétation de Σ . Soit E l'ensemble des éléments de B_Σ , dont I est modèle. Soit v l'assignation propositionnelle de domaine B_Σ caractéristique de E . Nous montrons que v est un modèle propositionnel des instances fermées des formules de Γ . Soit A une telle instance. D'après le lemme 5.1.11 page précédente, I est modèle de A . D'après le lemme 5.1.12 page ci-contre, l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle de A . D'après le lemme 5.1.14 page précédente, v est modèle de A .

\Leftarrow Supposons que tout ensemble fini d'instances fermées sur la signature Σ des formules de Γ a un modèle propositionnel de domaine B_Σ . D'après le théorème de compacité (théorème 1.2.30 page 18) l'ensemble de toutes les instances fermées sur la signature Σ a alors un modèle propositionnel v de domaine B_Σ . Ce modèle propositionnel peut être vu comme le modèle de Herbrand de $\forall(\Gamma)$ associé à l'ensemble des éléments de la base de Herbrand dont v est modèle. En effet, soit $E = \{A \in B_\Sigma \mid v(A) = 1\}$. Soit A une formule quelconque de Γ . D'après le lemme 5.1.14 page précédente, l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle des instances fermées de A , autrement dit, pour tout état e de cette interprétation, $H_{\Sigma,E}$ est modèle de Ae . D'après le lemme 5.1.15 page ci-contre, pour tout e , l'assignation $(H_{\Sigma,E}, e)$ est modèle de A . D'après le lemme 5.1.9 page 109, l'interprétation $H_{\Sigma,E}$ est modèle de $\forall(A)$. Puisque A est une formule quelconque de Γ , cette interprétation est modèle de $\forall(\Gamma)$.

□

Corollaire 5.1.18 Soit Γ un ensemble de formules sans quantificateur de signature Σ . La fermeture universelle $\forall(\Gamma)$ est insatisfaisable si et seulement s'il existe un ensemble fini insatisfaisable d'instances fermées sur la signature Σ des formules de Γ .

Preuve : Le corollaire est obtenu en remplaçant chaque côté de l'équivalence du théorème de Herbrand par sa négation. □

Dans le cas où Γ est un ensemble fini de formules, ce corollaire fonde une procédure de *semi-décision* pour savoir si $\forall(\Gamma)$ est ou n'est pas insatisfaisable : nous énumérons l'ensemble des instances fermées des formules de Γ sur la signature Σ et nous arrêtons cette énumération dès que nous obtenons un ensemble insatisfaisable, ou dès que cette énumération est terminée sans contradiction ou dès que nous sommes « fatigués » :

1. Dans le premier cas, la procédure répond (en appliquant le corollaire) que $\forall(\Gamma)$ est insatisfaisable.
2. Le deuxième cas ne peut se produire que si le domaine de Herbrand ne comprend que des constantes, la procédure répond (en appliquant le corollaire) que $\forall(\Gamma)$ est satisfaisable et en donne un modèle.
3. Dans le troisième cas, nous ne pouvons évidemment pas conclure : le corollaire nous dit que si $\forall(\Gamma)$ est insatisfaisable, et si nous avions été plus courageux, nous aurions obtenu une contradiction.

Exemple 5.1.19

1. Soit $\Gamma = \{P(x), Q(x), \neg P(a) \vee \neg Q(b)\}$. La signature Σ comporte 2 constantes a, b et 2 symboles de relation unaire P, Q .

Le domaine de Herbrand est $\{a, b\}$. La base de Herbrand est $\{P(a), P(b), Q(a), Q(b)\}$.

L'ensemble $\{P(a), Q(b), \neg P(a) \vee \neg Q(b)\}$ d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

2. Soit $\Gamma = \{P(x) \vee Q(x), \neg P(a), \neg Q(b)\}$. Nous avons la même signature que précédemment.

Le domaine de Herbrand est $\{a, b\}$. La base de Herbrand est $\{P(a), P(b), Q(a), Q(b)\}$.

L'ensemble de toutes les instances sur le domaine de Herbrand $\{P(a) \vee Q(a), P(b) \vee Q(b), \neg P(a), \neg Q(b)\}$ a un modèle propositionnel caractérisé par $E = \{P(b), Q(a)\}$. Donc l'interprétation de Herbrand associée à E est modèle de $\forall(\Gamma)$.

À partir de l'interprétation de Herbrand associée à E , nous pouvons fabriquer une interprétation modèle de $\forall(\Gamma)$, c'est-à-dire, un modèle de l'ensemble de formules $\{\forall x(P(x) \vee Q(x)), \neg P(a), \neg Q(b)\}$: soit I l'interprétation de domaine $\{0, 1\}$ où $a_I = 0, b_I = 1, P_I = \{1\}$ et $Q_I = \{0\}$; I est modèle de $\forall(\Gamma)$.

3. Soit $\Gamma = \{P(x), \neg P(f(x))\}$. La signature Σ comporte une constante a , un symbole de fonction unaire f et un symbole de relation unaire P . La constante a est ajoutée à la signature de Γ pour que le domaine de Herbrand ne soit pas vide.

Le domaine de Herbrand est $\{f^n(a) \mid n \in \mathbb{N}\}$. La base de Herbrand est $\{P(f^n(a)) \mid n \in \mathbb{N}\}$.

L'ensemble $\{P(f(a)), \neg P(f(a))\}$ d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

4. Soit $\Gamma = \{P(x) \vee \neg P(f(x)), \neg P(a), P(f(f(a)))\}$. Nous avons la même signature que précédemment.

Le domaine de Herbrand est $\{f^n(a) \mid n \in \mathbb{N}\}$. La base de Herbrand est $\{P(f^n(a)) \mid n \in \mathbb{N}\}$.

L'ensemble $\{P(a) \vee \neg P(f(a)), P(f(a)) \vee \neg P(f(f(a))), \neg P(a), P(f(f(a)))\}$ d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable. Observez qu'il a fallu prendre 2 instances de la première formule de Γ pour obtenir une contradiction.

5. Soit $\Gamma = \{R(x, s(x)), R(x, y) \wedge R(y, z) \Rightarrow R(x, z), \neg R(x, x)\}$. La signature Σ comporte une constante a , un symbole de fonction unaire s et un symbole de relation binaire R .

Le domaine de Herbrand est $D_\Sigma = \{s^n(a) \mid n \in \mathbb{N}\}$. Ce domaine est infini. La base de Herbrand est aussi infinie : $\{R(s^n(a), s^m(a)) \mid n, m \in \mathbb{N}\}$.

$\forall(\Gamma)$ a un modèle infini : l'interprétation I de domaine \mathbb{N} avec pour tout $n \in \mathbb{N}$, $s_I(n) = n + 1$ et $R_I = \{(n, p) \mid n < p\}$, en bref $R(x, y) = x < y$. Nous pouvons prouver que $\forall(\Gamma)$ n'a aucun modèle fini, autrement dit il est inutile de chercher des modèles finis avec les méthodes du chapitre précédent.

Puisque $\forall(\Gamma)$ a un modèle, nous sommes dans une situation où la procédure évoquée ci-dessus, ne pourra jamais donner de réponse, aussi longtemps que nous poursuivons l'énumération des instances des formules de Γ .

5.2 Skolémisation

Le théorème de Herbrand est applicable à la fermeture universelle d'un ensemble de formules sans quantificateur. Nous étudions une transformation, la *skolémisation*, qui change un ensemble de formules fermées en la fermeture universelle d'un ensemble de formules sans quantificateur et qui *préserve l'existence d'un modèle*. Cette transformation est due à Thoralf Albert Skolem (1887 - 1963), mathématicien et logicien norvégien. Elle est nécessaire pour pouvoir appliquer la résolution sur un ensemble de formules de la logique du premier ordre. Nous commençons par donner l'intuition de la skolémisation avant de regarder le résultat de cette transformation sur des exemples simples afin d'observer les propriétés de la skolémisation. La skolémisation sert à éliminer les quantificateurs existentiels et change une formule fermée A en une formule B telle que :

- B a pour conséquence A .
- tout modèle de A donne un modèle de B .

Par suite A a un modèle si et seulement si B a un modèle : la skolémisation préserve l'existence d'un modèle, nous disons aussi qu'elle préserve la satisfaisabilité. Regardons ce qu'il en est sur deux exemples simples.

Exemple 5.2.1 La formule $\exists x P(x)$ est skolémisée en $P(a)$. Nous observons les relations entre ces deux formules :

1. $P(a)$ a pour conséquence $\exists x P(x)$.
2. $\exists x P(x)$ n'a pas pour conséquence $P(a)$ mais un modèle de $\exists x P(x)$ « donne » un modèle de $P(a)$. En effet soit I un modèle de $\exists x P(x)$. Donc il existe $d \in P_I$. Soit J l'interprétation telle que $P_J = P_I$ et $a_J = d$. J est modèle de $P(a)$.

Exemple 5.2.2 La formule $\forall x \exists y Q(x, y)$ est skolemisée en $\forall x Q(x, f(x))$. Nous observons les relations entre ces deux formules :

1. $\forall x Q(x, f(x))$ a pour conséquence $\forall x \exists y Q(x, y)$.
2. $\forall x \exists y Q(x, y)$ n'a pas pour conséquence $\forall x Q(x, f(x))$ mais un modèle de $\forall x \exists y Q(x, y)$ « donne » un modèle de $\forall x Q(x, f(x))$. En effet soient I un modèle de $\forall x \exists y Q(x, y)$ et D le domaine de I . Pour tout $d \in D$, l'ensemble $\{e \in D \mid (d, e) \in Q_I\}$ n'est pas vide, donc il existe $g : D \rightarrow D$ une fonction¹ telle que pour tout $d \in D$, $g(d) \in \{e \in D \mid (d, e) \in Q_I\}$. Soit J l'interprétation J telle que $Q_J = Q_I$ et $f_J = g : J$ est modèle de $\forall x Q(x, f(x))$.

5.2.1 Algorithme de skolemisation d'une formule

Nous définissons d'abord les éléments nécessaires pour présenter la forme de Skolem, ensuite nous donnons un algorithme de skolemisation.

Définition 5.2.3 (Formule propre) Une formule fermée est dite propre, s'il elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemple 5.2.4 Les formules $\forall x P(x) \vee \forall x Q(x)$ et $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$

ne sont pas propres.

Les formules $\forall x P(x) \vee \forall y Q(y)$ et $\forall x (P(x) \Rightarrow \exists y R(x, y))$

sont propres.

Nous étendons la notion de forme normale de la logique des propositions à la logique du premier ordre en disant qu'une formule est en forme normale si elle est sans équivalence, ni implication, et dont les négations portent uniquement sur les formules atomiques.

Définition 5.2.5 (Forme de Skolem) Soient A une formule fermée et B la formule en forme normale sans quantificateur obtenue par la transformation ci-après : B est la forme de Skolem de A .

Il y a plusieurs façons de skolemiser une formule fermée. Nous proposons un compromis entre l'efficacité et la simplicité de la transformation. Notre algorithme est constitué de la succession des quatre étapes suivantes :

1. **Transformation en formule normale** : Transformation de la formule fermée en une autre formule fermée en forme normale équivalente.
2. **Transformation en formule propre** : Transformation de la formule fermée en forme normale en une formule fermée, en forme normale, propre et équivalente.
3. **Élimination des quantificateurs existentiels** : Cette transformation préserve *seulement l'existence de modèle*, comme nous l'avons vu sur les exemples.
4. **Transformation en forme de Skolem** : Transformation de la formule fermée, en forme normale, propre et sans quantificateur existentiel en une formule en forme normale sans quantificateur.

Nous détaillons chacune de ces quatre étapes.

5.2.1.1 Transformation en formule normale

Comme dans le cas propositionnel, nous enlevons les équivalences et les implications et déplaçons les négations vers les formules atomiques au moyen des équivalences suivantes utilisées de gauche à droite :

$$— A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A).$$

1. En affirmant l'existence de g , nous utilisons implicitement l'axiome du choix.

- $A \Rightarrow B \equiv \neg A \vee B$.
- $\neg\neg A \equiv A$.
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$.
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$.
- $\neg\forall x A \equiv \exists x \neg A$.
- $\neg\exists x A \equiv \forall x \neg A$.

Remarque 5.2.6 En combinant déplacement de la négation et élimination de l'implication, c'est-à-dire en remplaçant $\neg(A \Rightarrow B)$ par $A \wedge \neg B$, nous pouvons effectuer plus rapidement la transformation en formule normale équivalente, qu'en se limitant aux équivalences ci-dessus.

Exemple 5.2.7 La formule $\forall y(\forall x P(x, y) \Leftrightarrow Q(y))$ est transformée par élimination de l'équivalence et de l'implication en :

$$\forall y((\neg\forall x P(x, y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x, y))).$$

Puis par déplacement de la négation en

$$\forall y((\exists x \neg P(x, y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x, y))).$$

5.2.1.2 Transformation en formule propre

En utilisant les résultats sur les changements de variables présentés dans la section 4.4.3 page 100, nous changeons le nom des variables liées à deux quantificateurs, en choisissant par exemple de nouvelles variables à chaque changement de nom.

Exemple 5.2.8 La formule $\forall x P(x) \vee \forall x Q(x)$ est changée en

$$\forall x P(x) \vee \forall y Q(y).$$

La formule $\forall x(P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$ est changée en

$$\forall x(P(x) \Rightarrow \exists z Q(z) \wedge \exists y R(x, y))$$

5.2.1.3 Élimination des quantificateurs existentiels

Soit A une formule fermée normale et propre ayant une occurrence de la sous-formule $\exists y B$. Soient x_1, \dots, x_n l'ensemble des variables libres de $\exists y B$, où $n \geq 0$. Soit f un symbole ne figurant pas dans A . Nous remplaçons cette occurrence de $\exists y B$ par $B_{<y := f(x_1, \dots, x_n)>}$ ².

Théorème 5.2.9 Soit A' la formule obtenue en appliquant l'élimination d'un quantificateur existentiel sur la formule A . La formule A' est une formule fermée en forme normale et propre vérifiant :

1. A' a pour conséquence A ($A' \models A$).

2. Si $n = 0$, f est une constante.

2. Si A a un modèle alors A' a un modèle identique à celui de A sauf pour le sens de f .

Preuve :

1. Montrons que A' a pour conséquence A . Puisque la formule A est fermée et propre, les variables libres de $\exists yB$, qui sont liées à l'extérieur de $\exists yB$, ne sont pas liées par des quantificateurs dans B (sinon la propriété propre ne serait pas respectée), donc le terme $f(x_1, \dots, x_n)$ est libre pour y dans B et donc d'après le corollaire 4.3.38 page 94 : $B \langle y := f(x_1, \dots, x_n) \rangle$ a pour conséquence $\exists yB$. Nous en déduisons donc que A' a pour conséquence A .
2. Montrons que tout modèle de A donne un modèle de A' . Supposons que A a un modèle I où I est une interprétation de domaine D . Soit $c \in D$ et pour tous $d_1, \dots, d_n \in D$, soit E_{d_1, \dots, d_n} l'ensemble des éléments $d \in D$ tels que la formule B vaut 1 dans l'interprétation I et l'état $x_1 = d_1, \dots, x_n = d_n, y = d$ de ses variables libres. Soit $g : D^n \rightarrow D$ une fonction telle que si $E_{d_1, \dots, d_n} \neq \emptyset$ alors $g(d_1, \dots, d_n) \in E_{d_1, \dots, d_n}$ sinon $g(d_1, \dots, d_n) = c$. Soit J l'interprétation identique à I sauf que $f_J = g$. Nous avons :

- (a) $[\exists yB]_{(I,e)} = [B \langle y := f(x_1, \dots, x_n) \rangle]_{(J,e)}$, ceci d'après l'interprétation de f et le théorème 4.3.36 page 94, pour tout état e des variables,
- (b) $[\exists yB]_{(I,e)} = [\exists yB]_{(J,e)}$, puisque le symbole f est nouveau, la valeur de $\exists yB$ ne dépend pas du sens de f .
- (c) $\exists yB \Leftrightarrow B \langle y := f(x_1, \dots, x_n) \rangle \models A \Leftrightarrow A'$, d'après la propriété du remplacement 1.3.10 page 22, qui est aussi vraie en logique du premier ordre.

D'après ces trois points, nous obtenons $[A]_{(J,e)} = [A']_{(J,e)}$ et puisque f n'est pas dans A et que les formules A et A' n'ont pas de variables libres, nous avons $[A]_I = [A']_J$. Puisque I est modèle de A , J est modèle de A' .

□

Remarque 5.2.10 Dans le théorème 5.2.9 page précédente, il faut constater que la formule A' obtenue à partir de la formule A par élimination d'un quantificateur reste fermée, en forme normale et propre. Donc, en « appliquant » plusieurs fois le théorème, ce qui implique de choisir un nouveau symbole à chaque quantificateur éliminé, nous transformons une formule A fermée, en forme normale et propre en une formule B fermée, en forme normale, propre et sans quantificateur existentiel telle que :

- la formule A est conséquence de la formule B ,
- si A a un modèle, alors B a un modèle identique sauf pour le sens des nouveaux symboles.

Exemple 5.2.11 En éliminant les quantificateurs existentiels de la formule $\exists x \forall y P(x, y) \wedge \exists z \forall u \neg P(z, u)$ nous obtenons

$\forall y P(a, y) \wedge \forall u \neg P(b, u)$. Il est facile de voir que cette formule a un modèle. Mais si on fait l'erreur d'éliminer les deux quantificateurs existentiels avec la même constante a , nous obtenons la formule $\forall y P(a, y) \wedge \forall u \neg P(a, u)$, qui est insatisfaisable, puisqu'elle a pour conséquence $P(a, a)$ et $\neg P(a, a)$.

Donc, il faut impérativement utiliser un nouveau symbole lors de chaque élimination d'un quantificateur existentiel.

Exemple 5.2.12 En éliminant les quantificateurs existentiels de la formule $\exists x \forall y \exists z P(x, y, z)$ nous obtenons

deux solutions possibles :

- $\forall y P(a, y, f(a, y))$
- $\forall y P(a, y, f(y))$

Ceci dépend de l'ordre dans lequel nous appliquons l'élimination des quantificateurs existentiels. Or le théorème 5.2.9 montre que une des ces transformations a un modèle si et seulement si l'autre a également un modèle.

3. En affirmant l'existence de g , nous utilisons implicitement l'axiome du choix.

5.2.1.4 Transformation en fermeture universelle

Nous pouvons enfin enlever les quantificateurs universels afin d'obtenir une forme de Skolem.

Théorème 5.2.13 *Soit A une formule fermée, en forme normale, propre et sans quantificateur existentiel. Soit B la formule obtenue en enlevant de A tous les quantificateurs universels. La formule A est équivalente à la fermeture universelle de B .*

Preuve : Avec les conditions posées sur A , la transformation de A en $\forall(B)$ revient à effectuer tous les remplacements possibles des sous-formules de la forme suivante :

- $(\forall x C) \wedge D$ par $\forall x(C \wedge D)$, où x non libre dans D ,
- $(\forall x C) \vee D$ par $\forall x(C \vee D)$, où x non libre dans D ,
- $D \wedge (\forall x C)$ par $\forall x(D \wedge C)$, où x non libre dans D ,
- $D \vee (\forall x C)$ par $\forall x(D \vee C)$, où x non libre dans D .

Puisque chacun de ces remplacements change une formule en une autre équivalente, les formules A et $\forall(B)$ sont équivalentes. \square

5.2.2 Propriétés de la forme de Skolem

Propriété 5.2.14 *Soit A une formule fermée et B la forme de Skolem de A .*

- *La formule $\forall(B)$ a pour conséquence la formule A ,*
- *si A a un modèle alors $\forall(B)$ a un modèle.*

Donc A a un modèle si et seulement si $\forall(B)$ a un modèle.

Preuve : Soit C la formule fermée en forme normale et propre, obtenue au terme des deux premières étapes de la skolémisation de A . Soit D le résultat de l'élimination des quantificateurs existentiels appliquée à C . D'après la remarque 5.2.10 page précédente nous avons :

- la formule D a pour conséquence la formule C ,
- si C a un modèle alors D a un modèle.

Puisque les deux premières étapes changent des formules en des formules équivalentes, A et C sont équivalentes. D'après le théorème 5.2.13 page ci-contre, D est équivalent à $\forall(B)$. Donc nous pouvons remplacer ci-dessus D par $\forall(B)$ et C par A , ce qui prouve le théorème. \square

Exemple 5.2.15 *Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. Nous skolémisons $\neg A$.*

1. $\neg A$ est transformée en la formule en forme normale :

$$\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x).$$

2. La formule en forme normale est transformée en la formule propre :

$$\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z).$$

3. Le quantificateur existentiel est « remplacé » par une constante :

$$\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \neg Q(a).$$

4. Les quantificateurs universels sont enlevés :

$$(\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a).$$

Cette formule est la forme de Skolem de $\neg A$ suivant la définition 5.2.5 page 114.

Instancions la forme de Skolem de $\neg A$ en remplaçant x et y par a . Nous obtenons la formule $(\neg P(a) \vee Q(a)) \wedge P(a) \wedge \neg Q(a)$ qui est insatisfaisable. Donc $\forall((\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a))$ est insatisfaisable. Puisque, d'après la propriété 5.2.14 page précédente, la skolemisation préserve l'existence d'un modèle, $\neg A$ est insatisfaisable, donc A est valide.

Pour skolemiser un ensemble de formules, nous skolemisons chaque formule de l'ensemble comme il a été indiqué ci-dessus en choisissant impérativement un nouveau symbole pour chaque quantificateur existentiel éliminé durant la troisième étape de la skolemisation. Les raisons pour le choix d'un symbole nouveau apparaissent clairement dans la preuve du théorème 5.2.9 page 115 et dans l'exemple 5.2.11 page 116.

Corollaire 5.2.16 En skolemisant un ensemble de clauses fermées Γ , nous obtenons un ensemble Δ de formules sans quantificateurs tel que :

- Tout modèle de $\forall(\Delta)$ est modèle de Γ .
- Si Γ a un modèle alors $\forall(\Delta)$ en a un modèle qui ne diffère de celui de Γ que par le sens des nouveaux symboles.

La skolemisation a des conséquences pratiques : elle facilite la preuve d'insatisfiabilité d'une formule. Elle a aussi des conséquences théoriques, que nous examinons.

Théorème 5.2.17 Soit Γ un ensemble dénombrable de formules fermées. Si Γ a un modèle alors Γ a un modèle dénombrable.

Preuve : Nous skolemisons Γ en un ensemble Δ de formules sans quantificateurs. Supposons que Γ a un modèle. Alors $\forall(\Delta)$ a un modèle d'après le corollaire 5.2.16. D'après le théorème 5.1.16 page 110, $\forall(\Delta)$ a un modèle de Herbrand sur la signature de Δ . Puisque l'ensemble Γ est dénombrable, sa signature l'est aussi, donc aussi celle de Δ et par suite le domaine de ce modèle de Herbrand est dénombrable. \square

Remarque 5.2.18 Appelons théorie, un ensemble de formules. D'après le théorème, toute théorie dénombrable qui a un modèle, a un modèle dénombrable. Autrement dit avec une théorie du premier ordre, nous pouvons parler des propriétés des nombres réels ou des ensembles, mais nous ne pouvons pas obliger notre théorie à n'avoir que des modèles non dénombrables, bien que l'ensemble des réels ou la classe de tous les ensembles ne soient pas dénombrables.

5.2.3 Forme clause

Après la skolémisation des formules et avant de pouvoir appliquer une généralisation au premier ordre de la méthode de résolution il faut transformer les formes de Skolem en clauses comme nous l'avons fait en logique propositionnelle. Nous étendons donc les notions essentielles à cette transformation avant de donner les règles permettant cette transformation.

Définition 5.2.19 (Littéral positif, négatif et clause) *Un littéral positif est une formule atomique. Un littéral négatif est la négation d'une formule atomique.*

Nous remarquons que tout littéral est positif ou négatif, et nous rappelons qu'une clause est une somme de littéraux.

Définition 5.2.20 (Forme clause d'une formule) *Soit A une formule fermée, la forme clause de A est un ensemble de clauses obtenu en deux étapes :*

1. Skolémisation de A , autrement dire construire B la forme de Skolem de A .
2. Remplacement de B par un ensemble Γ équivalent de clauses obtenu par distributivité de la somme sur le produit.

Propriété 5.2.21 *La fermeture universelle de la forme clause d'une formule fermée a un modèle si et seulement la formule a un modèle. Plus précisément :*

- la fermeture universelle de la forme clause d'une formule a pour conséquence la formule,
- si la formule a un modèle alors la fermeture universelle de sa forme clause en a un.

Preuve : Soient A une formule fermée, B sa forme de Skolem et Γ sa forme clause. D'après les propriétés de la skolémisation :

- $\forall(B)$ a pour conséquence A .
- Si A a un modèle alors $\forall(B)$ a un modèle.

Puisque Γ est obtenu par distributivité, B et Γ sont équivalents donc $\forall(B)$ et $\forall(\Gamma)$ sont aussi équivalents. Par suite dans les deux propriétés ci-dessus, nous pouvons remplacer $\forall(B)$ par $\forall(\Gamma)$. \square

Définition 5.2.22 (Forme clause d'un ensemble de formules) *Soit Γ un ensemble de formules fermées. Nous définissons la forme clause de Γ comme l'union des formes clause de chacune des formules de Γ , en prenant soin au cours de la skolémisation d'éliminer chaque occurrence d'un quantificateur existentiel à l'aide d'un nouveau symbole.*

Corollaire 5.2.23 *Soient Γ un ensemble de formules fermées et Δ la forme clause de Γ . Nous avons :*

- $\forall(\Delta)$ a pour conséquence Γ , et
- si Γ a un modèle alors $\forall(\Delta)$ a un modèle.

Théorème 5.2.24 (Adaptation du théorème de Herbrand aux formes clause) *Soient Γ un ensemble de formules fermées et Δ la forme clause de Γ . L'ensemble Γ est insatisfaisable si et seulement s'il existe un sous-ensemble fini insatisfaisable d'instances des clauses de Δ sur la signature de Δ .*

Preuve : D'après 5.2.23, la skolémisation préserve la satisfaisabilité, donc : Γ est insatisfaisable si et seulement si $\forall(\Delta)$ est insatisfaisable. D'après le corollaire du théorème de Herbrand 5.1.18 page 111, $\forall(\Delta)$ est insatisfaisable si et seulement s'il existe un sous-ensemble fini insatisfaisable d'instances des clauses de Δ sur la signature de Δ . \square

Exemple 5.2.25 *Soit $A = \exists y \forall z (P(z, y) \Leftrightarrow \neg \exists x (P(z, x) \wedge P(x, z)))$. Calculons la forme clause de A .*

1. Mettre A sous forme en forme normale :

$$\exists y \forall z ((\neg P(z, y) \vee \forall x (\neg P(z, x) \vee \neg P(x, z))) \wedge \exists x (P(z, x) \wedge P(x, z)) \vee P(z, y)).$$

2. Rendre propre le résultat :

$$\exists y \forall z ((\neg P(z, y) \vee \forall x (\neg P(z, x) \vee \neg P(x, z))) \wedge \exists u (P(z, u) \wedge P(u, z)) \vee P(z, y)).$$

3. Éliminer les quantificateurs existentiels :

$$\forall z ((\neg P(z, a) \vee \forall x (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, f(z)) \wedge P(f(z), z)) \vee P(z, a)).$$

4. Supprimer les quantificateurs universels, nous obtenons la forme de Skolem de A :

$$((\neg P(z, a) \vee (\neg P(z, x) \vee \neg P(x, z))) \wedge (P(z, f(z)) \wedge P(f(z), z)) \vee P(z, a)).$$

5. Transformer en produit de sommes de littéraux, nous obtenons la forme clausale de A, qui est l'ensemble suivant de clauses :

- $C_1 = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z).$
- $C_2 = P(z, f(z)) \vee P(z, a).$
- $C_3 = P(f(z), z) \vee P(z, a).$

D'après la propriété ci-dessus, A n'a pas de modèle si et seulement s'il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses. Nous recherchons ces instances :

- Soit C'_1 obtenue avec $x := a, z := a$ dans C_1 : $C'_1 = \neg P(a, a).$
- Soit C''_1 obtenue avec $x := a, z := f(a)$ dans C_1 : $C''_1 = \neg P(f(a), a) \vee \neg P(a, f(a)).$
- Soit C'_2 obtenue avec $z := a$ dans C_2 : $C'_2 = P(a, f(a)) \vee P(a, a).$
- Soit C'_3 obtenue avec $z := a$ dans C_3 : $C'_3 = P(f(a), a) \vee P(a, a).$

L'ensemble de ces instances est insatisfaisable, donc A est insatisfaisable.

5.3 Unification

Après la mise en forme clausale d'un ensemble de formules en logique du premier ordre et avant de généraliser la résolution pour ces clauses, nous avons besoin d'un algorithme résolvant l'égalité entre deux termes. Étant donné deux termes, trouver une substitution qui appliquée à ces deux termes les rendent égaux est appelé *problème d'unification* de ces deux termes. Ce problème a intéressé de grands noms de l'informatique comme Alan Robinson [?], Gérard Huet [?, ?], Alberto Martelli et Ugo Montanari [?], ou encore plus récemment Franz Baader et Ayne Snyder [?].

5.3.1 Unificateur

Définition 5.3.1 (Expression, solution) Appelons expression, un terme ou un littéral. Une substitution σ (voir définition 5.1.3 page 108) est solution de l'équation $e_1 = e_2$ entre deux expressions, si les deux expressions $e_1\sigma$ et $e_2\sigma$ sont identiques. Une substitution est solution d'un ensemble d'équations si elle est solution de chaque équation de l'ensemble.

Définition 5.3.2 (Unificateur) Soient σ une substitution et E un ensemble d'expressions. $E\sigma = \{t\sigma \mid t \in E\}$. La substitution σ est un unificateur de E si et seulement si l'ensemble $E\sigma$ n'a qu'un élément. Soit $\{e_i \mid 1 \leq i \leq n\}$ un ensemble fini d'expressions. La substitution σ est un unificateur de cet ensemble si et seulement si elle est solution du système d'équations $\{e_i = e_{i+1} \mid 1 \leq i < n\}$.

Définition 5.3.3 (Support d'une substitution) Le support d'une substitution σ est l'ensemble des variables x telles que $x\sigma \neq x$. Dans la suite, nous nous intéressons seulement aux substitutions à support fini, c'est à dire qui ne changent qu'un nombre fini de variables. Une substitution σ à support fini est notée $\langle x_1 := t_1, \dots, x_n := t_n \rangle$ ou plus simplement $x_1 := t_1, \dots, x_n := t_n$ quand il n'y a pas de risque d'ambiguïté. Les variables x_1, \dots, x_n sont distinctes et la substitution vérifie :

- Pour i de 1 à n , $x_i\sigma = t_i$.
- Pour toute variable y telle que $y \notin \{x_1, \dots, x_n\}$, nous avons : $y\sigma = y$.

Exemple 5.3.4 L'équation $P(x, f(y)) = P(g(z), z)$ a pour solution :

La substitution $x := g(f(y)), z := f(y)$.

Le système d'équations $x = g(z), f(y) = z$ a pour solution :

La substitution $x := g(f(y)), z := f(y)$.

Définition 5.3.5 (Composition de substitutions) Soient σ et τ deux substitutions, nous notons $\sigma\tau$ la substitution telle que pour toute variable x , $x\sigma\tau = (x\sigma)\tau$. La substitution $\sigma\tau$ est une instance de σ . Deux substitutions sont équivalentes si chacune d'elles est une instance de l'autre.

Exemple 5.3.6 Considérons les substitutions suivantes :

- $\sigma_1 = \langle x := g(z), y := z \rangle$.
- $\sigma_2 = \langle x := g(y), z := y \rangle$.
- $\sigma_3 = \langle x := g(a), y := a, z := a \rangle$.

Nous avons les relations suivantes entre ces substitutions :

- $\sigma_1 = \sigma_2 < y := z >$.
- $\sigma_2 = \sigma_1 < z := y >$.
- $\sigma_3 = \sigma_1 < z := a >$.
- $\sigma_3 = \sigma_2 < y := a >$.

Les substitutions σ_1 et σ_2 sont équivalentes. La substitution σ_3 est une instance de σ_1 ainsi que de σ_2 , mais ne leur est pas équivalente.

Définition 5.3.7 (Solution la plus générale) Une solution d'un système d'équations est appelée la plus générale (en français nous disons aussi principale) si toute autre solution en est une instance. Notons que deux solutions « les plus générales » sont équivalentes.

Exemple 5.3.8 L'équation $f(x, g(z)) = f(g(y), x)$ a pour solutions (entre autres) :

$\sigma_1 = < x := g(z), y := z >$, $\sigma_2 = < x := g(y), z := y >$, $\sigma_3 = < x := g(a), y := a, z := a >$. σ_1 et σ_2 en sont les solutions « les plus générales ».

Définition 5.3.9 (Unificateur le plus général) Soit E un ensemble d'expressions. Nous rappelons qu'une expression est un terme ou un littéral. Un unificateur de E (voir définition 5.3.2 page ci-contre) est appelé le plus général (ou encore principal), si tout autre unificateur en est une instance.

Remarque 5.3.10 (Unificateur le plus général et solution la plus générale) Soit $E = \{e_i \mid 1 \leq i \leq n\}$ un ensemble d'expressions. Dans la définition d'un unificateur, nous avons indiqué que σ est un unificateur de E si et seulement si σ est solution du système $S = \{e_i = e_{i+1} \mid 1 \leq i < n\}$.
Donc l'unificateur le plus général de E est la solution la plus générale de S .

5.3.2 Algorithme d'unification

L'algorithme calculant la solution d'un système d'équations est appelé *algorithme d'unification* car la recherche d'un unificateur d'un ensemble d'expressions se réduit à la recherche d'une solution d'un système d'équations. L'algorithme sépare les équations en équations à résoudre, notées par une égalité et équations résolues, notées par le signe $:=$. Initialement, il n'y a pas d'équations résolues. L'algorithme applique les règles énoncées ci-dessous. Il s'arrête quand il n'y a plus d'équations à résoudre ou quand il a déclaré que le système à résoudre n'a pas de solution. Quand il s'arrête sans avoir déclaré l'absence de solution, la liste des équations résolues est la solution la plus générale du système initial d'équations.

5.3.2.1 Les règles de l'algorithme

Notre algorithme d'unification consiste à appliquer les six règles suivantes :

1. **Supprimer.** Si les deux membres d'une équation sont identiques, l'équation est supprimée⁴.
2. **Décomposer.** Si les 2 membres d'une équation sont distincts et commencent par le même symbole alors :
 - Une équation de la forme $\neg A = \neg B$ est remplacée par $A = B$.
 - Une équation de la forme $f(s_1, \dots, s_n) = f(t_1, \dots, t_n)$ est remplacée par les équations $s_1 = t_1, \dots, s_n = t_n$. Pour $n = 0$, cette décomposition supprime l'équation.
3. **Échec de la décomposition.** Si une équation à résoudre est de la forme $f(s_1, \dots, s_n) = g(t_1, \dots, t_p)$ avec $f \neq g$ ou $n \neq p$ alors l'algorithme déclare qu'il n'y a pas de solution. En particulier il y a évidemment un échec si nous cherchons à résoudre une équation entre un littéral positif et un littéral négatif ou bien si nous avons une constante et une fonction.

4. Nous pouvons nous limiter à la suppression des équations de la forme $x = x$ où x est une variable.

4. **Orienter.** Si une équation est de la forme $t = x$ où t est un terme qui n'est pas une variable et x une variable, alors nous remplaçons l'équation par $x = t$.
5. **Élimination d'une variable :** Si une équation à résoudre est de la forme $x = t$ où x est une variable et t un terme *ne contenant pas* x alors :
 - (a) Enlever des équations à résoudre $x = t$.
 - (b) Remplacer x par t dans toutes les équations (non résolues *et résolues*)⁵.
 - (c) Ajouter $x := t$ à la partie résolue.
6. **Échec de l'élimination :** Si une équation à résoudre est de la forme $x = t$ où x est une variable et t un terme distinct de x et *contenant* x alors l'algorithme déclare qu'il n'y a pas de solution.

Exemple 5.3.11 Nous appliquons cet algorithme afin de résoudre les équations suivantes :

1. $f(x, g(z)) = f(g(y), x)$.

Par décomposition, nous obtenons : $x = g(y), g(z) = x$.

Par élimination de x , nous obtenons : $x := g(y), g(z) = g(y)$.

Par décomposition, nous obtenons : $x := g(y), z = y$.

Par élimination de z , nous obtenons la solution : $x := g(y), z := y$.

2. $f(x, x, a) = f(g(y), g(a), y)$.

Par décomposition, nous obtenons : $x = g(y), x = g(a), a = y$.

Par élimination de x , grâce à la première équation, nous obtenons : $x := g(y), g(y) = g(a), a = y$.

Par décomposition, nous obtenons : $x := g(y), y = a, a = y$.

Par élimination de y , nous obtenons : $x := g(a), y := a, a = a$.

Par suppression de l'identité, nous obtenons : $x := g(a), y := a$.

3. $f(x, x, x) = f(g(y), g(a), y)$.

Par décomposition, nous obtenons : $x = g(y), x = g(a), x = y$.

Par élimination de x , nous obtenons : $x := g(y), g(y) = g(a), g(y) = y$.

Par orientation des équations, nous obtenons : $x := g(y), g(y) = g(a), y = g(y)$.

L'équation $y = g(y)$ engendre un échec.

Donc l'équation $f(x, x, x) = f(g(y), g(a), y)$ n'a pas de solution.

5. Un algorithme efficace évite ce remplacement systématique.

Nous montrons d'abord que notre algorithme est correct, c'est-à-dire qu'il calcule bien un unificateur. Ensuite nous démontrons qu'il termine toujours.

5.3.2.2 Correction de l'algorithme

Nous observons facilement les trois faits suivants :

- (1) : les règles de suppression, de décomposition, d'orientation et d'élimination des variables préservent l'ensemble des solutions.
- (2) : initialement et après toute application de règle, l'ensemble des équations résolues forme une substitution $x_1 := t_1, \dots, x_n := t_n$ où x_1, \dots, x_n sont des variables distinctes ne figurant pas dans les termes t_1, \dots, t_n . Soit σ cette substitution. Il est clair que $t_i \sigma = t_i$ ainsi σ est solution des équations résolues.
- (3) : Les règles d'échec déclarent correctement l'absence de solution.

De (1) et (2), il résulte que si l'algorithme termine sans échec, la substitution σ trouvée est la solution la plus générale du système *initial*. En effet :

- σ est une solution, puisque, d'après (2), elle est solution du système *terminal* et que, d'après (1), l'ensemble des solutions est préservé par les règles.
- σ est une solution la plus générale, puisque, d'après (1), toute autre solution est une solution du système terminal, donc une instance de la substitution définie par ce système.

De (1) et de (3), il résulte que si l'algorithme déclare un échec, alors le système initial n'a pas de solution. En effet, dans ce cas :

- si le système initial avait une solution, d'après (1), elle serait solution du système terminal,
- d'après (3), le système terminal n'a pas de solution.

5.3.2.3 Terminaison de l'algorithme

- Les règles de suppression et de décomposition font décroître strictement la somme des longueurs des termes qui sont de part et d'autre des équations.
 - La règle d'orientation fait décroître strictement la somme des longueurs des termes qui sont à gauche des équations.
 - La règle d'élimination fait décroître strictement le nombre de variables des équations non résolues.
- Il résulte que l'algorithme d'unification termine.

5.4 Résolution au premier ordre

Nous avons maintenant tous les ingrédients pour présenter la résolution au premier ordre. Bien entendu nous transformons un ensemble de formules en forme clausale. Ensuite nous pouvons appliquer trois règles permettant de « factoriser » des termes, de « copier » une clause et enfin de faire une « résolution binaire ». Ces différentes règles utilisent l'algorithme d'unification présenté précédemment. Par la suite, nous montrons la cohérence et la complétude de ce système formel. La complétude est basée sur le théorème de Herbrand dans la version donnée dans le corollaire 5.2.24 page 119.

5.4.1 Trois règles pour la résolution

Soit Γ un ensemble de clauses, nous supposons que $\forall(\Gamma)$ n'a pas de modèle, nous proposons alors un système formel permettant de déduire \perp de Γ avec les trois règles suivantes :

1. La *factorisation* qui de la prémisse $P(x, f(y)) \vee P(g(z), z) \vee Q(z, x)$ déduit $P(g(f(y)), f(y)) \vee Q(f(y), g(f(y)))$. La clause déduite est obtenue en calculant la solution la plus générale $x := g(f(y)), z := f(y)$ de $P(x, f(y)) = P(g(z), z)$.
2. La règle de *copie* qui permet de renommer les variables d'une clause.
3. La *résolution binaire* (RB) qui des deux prémisses sans variable commune $P(x, a) \vee Q(x)$ et $\neg P(b, y) \vee R(f(y))$ déduit le résolvant $Q(b) \vee R(f(a))$, en calculant la solution plus générale $x := b, y := a$ de $P(x, a) = P(b, y)$.

Pour simplifier la présentation des règles, nous identifierons une clause, qui est une somme de littéraux, avec l'ensemble de ses littéraux. Nous formalisons ces trois règles.

5.4.1.1 Factorisation

Propriété 5.4.1 Soient A une formule sans quantificateur et B une instance de A . Nous avons : $\forall(A) \models \forall(B)$.

Preuve : Soit I une interprétation quelconque. Puisque $\forall(A)$ et $\forall(B)$ n'ont pas de variables libres, il suffit de montrer que si I est modèle de $\forall(A)$ alors I est modèle de $\forall(B)$. Supposons I modèle de $\forall(A)$. D'après le lemme 5.1.11 page 110, pour tout état e , (I, e) est modèle de B . D'après le lemme 5.1.9 page 109, I est modèle de $\forall(B)$. \square

Définition 5.4.2 La clause C' est un facteur de la clause C si $C' = C$ ou s'il existe un sous-ensemble E de C tel que E a au moins deux éléments, E est unifiable et $C' = C\sigma$ où σ est l'unificateur le plus général de E .

Exemple 5.4.3 La clause $\underline{P(x)} \vee Q(g(x, y)) \vee \underline{P(f(a))}$ a deux facteurs :

elle-même et le facteur $P(f(a)) \vee Q(g(f(a), y))$ obtenu en appliquant à la clause, l'unificateur le plus général $x := f(a)$ des deux littéraux soulignés.

Nous obtenons immédiatement la propriété suivante.

Propriété 5.4.4 (Cohérence de la règle de factorisation) Soit C' un facteur de la clause C . Nous avons : $\forall(C) \models \forall(C')$.

Preuve : Puisque C' est une instance de C , la propriété est une conséquence immédiate de la propriété 5.4.1. \square

5.4.1.2 Copie d'une clause

Définition 5.4.5 Soient C une clause et σ une substitution, qui ne change que les variables de C et dont la restriction aux variables de C est une bijection entre ces variables et celles de la clause $C\sigma$. La clause $C\sigma$ est une copie de la clause C . La substitution σ est aussi appelée un renommage de C .

Définition 5.4.6 Soient C une clause et σ un renommage de C . Soit f la restriction de σ aux variables de C et f^{-1} l'application réciproque de f . Soit σ_C^{-1} la substitution ainsi définie pour toute variable x :

- Si x est une variable de $C\sigma$ alors $x\sigma_C^{-1} = xf^{-1}$ (par soucis de clarté, nous avons préféré utiliser la notation postfixée xf^{-1} plutôt que la notation préfixée $f^{-1}(x)$, plus usuelle).
- Sinon $x\sigma_C^{-1} = x$.

Cette substitution est appelée l'inverse du renommage σ de C .

Exemple 5.4.7 Soit $\sigma = \langle x := u, y := v \rangle$. σ est un renommage du littéral $P(x, y)$. Le littéral $P(u, v)$, où $P(u, v) = P(x, y)\sigma$, est une copie de $P(x, y)$. Soit $\tau = \langle u := x, v := y \rangle$. τ est l'inverse du renommage σ de $P(x, y)$. Notons que $P(u, v)\tau = P(x, y)$: le littéral $P(x, y)$ est une copie de $P(u, v)$ par le renommage τ .

Propriété 5.4.8 Soient C une clause et σ un renommage de C .

1. σ_C^{-1} est un renommage de $C\sigma$.
2. Pour toute expression ou clause E , dont les variables sont celles de C , $E\sigma\sigma_C^{-1} = E$.

Donc $C\sigma\sigma_C^{-1} = C$ et par suite C est une copie de $C\sigma$.

Preuve : Soit f la restriction de σ aux variables de C . Par définition du renommage, f est une bijection entre les variables de C et celles de $C\sigma$.

1. Par définition de σ_C^{-1} , cette substitution ne change que les variables de $C\sigma$ et sa restriction aux variables de $C\sigma$ est la bijection f^{-1} . Donc, σ_C^{-1} est un renommage de $C\sigma$.
2. Soit x une variable de C . Par définition de f , $x\sigma\sigma_C^{-1} = xf^{-1} = x$. Donc, par une récurrence omise et sans intérêt sur les termes, littéraux et clauses, pour toute expression ou clause E , dont les variables sont celles de C , nous avons $E\sigma\sigma_C^{-1} = E$.

\square

Propriété 5.4.9 Soient deux clauses copies l'une de l'autre, leurs fermetures universelles sont équivalentes.

Preuve : Soit C' une copie de C . Par définition, C' est une instance de C et par la propriété précédente, C est une copie de C' , donc une instance de C' . Donc par la propriété 5.4.1 page précédente, la fermeture universelle de C est conséquence de celle de C' et inversement. Par suite, ces deux fermetures universelles sont équivalentes. \square

5.4.1.3 Résolution binaire

Définition 5.4.10 Soient C et D deux clauses n'ayant pas de variable commune. La clause E est un résolvant binaire de C et D s'il y a un littéral $L \in C$ et un littéral $M \in D$ tels que L et M^c (le littéral opposé à M) sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution la plus générale de l'équation $L = M^c$.

Exemple 5.4.11 Soient $C = P(x, y) \vee P(y, k(z))$ et $D = \neg P(a, f(a, y_1))$.

$\langle x := a, y := f(a, y_1) \rangle$ est la solution la plus générale de $P(x, y) = P(a, f(a, y_1))$, donc $P(f(a, y_1), k(z))$ est un résolvant binaire des clauses C et D .

Propriété 5.4.12 (Cohérence de la règle de résolution binaire) Soit E un résolvant binaire des clauses C et D , nous avons : $\forall(C), \forall(D) \models \forall(E)$.

Preuve : Soit I une interprétation. Puisque les trois formules $\forall(C), \forall(D), \forall(E)$ sont sans variable libre, il suffit de montrer que si I est modèle de $\forall(C)$ et de $\forall(D)$ alors I est modèle de $\forall(E)$. Supposons que I est modèle de $\forall(C), \forall(D)$ et montrons que c'est un modèle de $\forall(E)$. Soit e un état *quelconque* de cette interprétation. D'après le lemme 5.1.11 page 110, (I, e) est modèle de $C\sigma$ et $D\sigma$, pour toute substitution σ . Par définition du résolvant binaire, il y a un littéral $L \in C$ et un littéral $M \in D$ tels que L et M^c (le littéral opposé à M) sont unifiables. Ainsi, $L\sigma$ et $M\sigma$ sont deux littéraux opposés, donc (I, e) est modèle soit de l'un, soit de l'autre de ces deux littéraux.

1. Supposons (I, e) modèle de $L\sigma$. Puisque (I, e) est contre-modèle de $M\sigma$ et modèle de $D\sigma$, c'est un modèle de $(D - \{M\})\sigma$ donc de E car $(D - \{M\})\sigma \subset E$.
2. Supposons (I, e) contre-modèle de $L\sigma$. Puisque c'est un modèle de $C\sigma$, c'est un modèle de $(C - \{L\})\sigma$ donc de E car $(C - \{L\})\sigma \subset E$.

Donc (I, e) est modèle de E . Puisque l'état e est quelconque, d'après le lemme 5.1.9 page 109, I est modèle de $\forall(E)$. \square

5.4.1.4 Preuve par factorisation, copie et résolution binaire

Définition 5.4.13 Soient Γ un ensemble de clauses et C une clause. Une preuve de C à partir de Γ est une suite de clauses se terminant par C , toute clause de la preuve étant un élément de Γ , un facteur d'une clause la précédant dans la preuve, une copie d'une clause la précédant dans la preuve ou un résolvant binaire de deux clauses la précédant dans la preuve.

C est déduite de Γ au premier ordre par les 3 règles de factorisation, copie et résolution binaire est dénoté par $\Gamma \vdash_{1fcb} C$. Cela signifie qu'il y a une preuve de C à partir de Γ . Quand il n'y a pas d'ambiguïté sur le système formel utilisé, nous remplaçons \vdash_{1fcb} par \vdash .

5.4.2 Cohérence de la résolution

Propriété 5.4.14 (Cohérence) Soient Γ un ensemble de clauses et C une clause. Si $\Gamma \vdash_{1fcb} C$ alors $\forall(\Gamma) \models \forall(C)$.

Preuve : Cette propriété est une conséquence immédiate de la cohérence de la factorisation, de la copie et de la résolution binaire. Cette preuve est demandée dans l'exercice 91 page 133. \square

Exemple 5.4.15 Soient les deux clauses :

1. $C_1 = P(x, y) \vee P(y, x)$.
2. $C_2 = \neg P(u, z) \vee \neg P(z, u)$.

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

numéro	conclusion	justification
(1)	$P(x, y) \vee P(y, x)$	Hyp C_1
(2)	$P(y, y)$	Facteur de 1 par $\langle x := y \rangle$
(3)	$\neg P(u, z) \vee \neg P(z, u)$	Hyp C_2
(4)	$\neg P(z, z)$	Facteur de 3 par $\langle u := z \rangle$
(5)	\perp	RB 2, 4 par $\langle y := z \rangle$

Cet exemple montre, a contrario, que la résolution binaire seule est incomplète, sans la factorisation, nous ne pouvons pas déduire la clause vide.

Exemple 5.4.16 Soient les trois clauses :

1. $C_1 = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$.
2. $C_2 = P(z, f(z)) \vee P(z, a)$.
3. $C_3 = P(f(z), z) \vee P(z, a)$.

Nous donnons une preuve par résolution que $\forall (C_1, C_2, C_3)$ n'a pas de modèle. Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^e littéral de la clause 1 et le 1^e littéral de la clause 3.

numéro	conclusion	justification	unificateur
(1)	$\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$	Hyp C_1	
(2)	$P(z, f(z)) \vee P(z, a)$	Hyp C_2	
(3)	$P(v_0, f(v_0)) \vee P(v_0, a)$	Copie 2	$\text{par} \langle z := v_0 \rangle$
(4)	$\neg P(f(v_0), a) \vee \neg P(f(v_0), v_0) \vee P(v_0, a)$	RB 1(3), 3(1)	$\text{par} \langle z := f(v_0); x := v_0 \rangle$
(5)	$\neg P(f(a), a) \vee P(a, a)$	Fact 4	$\text{par} \langle v_0 := a \rangle$
(6)	$\neg P(a, a)$	Fact 1	$\text{par} \langle x := a; z := a \rangle$
(7)	$P(f(z), z) \vee P(z, a)$	Hyp C_3	
(8)	$P(f(a), a)$	RB 6(1), 7(2)	$\text{par} \langle z := a \rangle$
(9)	$P(a, a)$	RB 5(1), 8(1)	
(10)	\perp	RB 6(1), 9(1)	

5.4.3 Complétude de la résolution

Nous définissons une *nouvelle* règle, la résolution au 1^o ordre, qui est une combinaison des trois règles de factorisation, copie et résolution binaire. Ceci nous permettra de prouver la complétude de la résolution.

Définition 5.4.17 La clause E est un résolvant au 1^o ordre des clauses C et D si E est un résolvant binaire de C' et D' où C' est un facteur de C et D' est une copie sans variable commune avec C' d'un facteur de D . La règle qui de C et D déduit E est appelée la résolution de 1^o ordre.

Exemple 5.4.18 Soient $C = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$ et $D = P(z, f(z)) \vee P(z, a)$.

$C' = \neg P(a, a)$ est un facteur de C . La clause $P(a, f(a))$ est un résolvant binaire de C' et de D (qui est facteur de lui-même) donc c'est un résolvant au premier ordre de C et D .

Soient Γ un ensemble de clauses et C une clause, nous avons défini jusqu'à présent trois notions de preuve par résolution que nous distinguons en notant par :

1. $\Gamma \vdash_p C$ le fait qu'il y existe une preuve de C à partir de Γ obtenue par résolution propositionnelle, autrement dit sans substitution.
2. $\Gamma \vdash_{fcb} C$ le fait qu'il y existe une preuve de C à partir de Γ par factorisation, copie et résolution binaire.
3. $\Gamma \vdash_{1r} C$ le fait qu'il y existe une preuve de C à partir de Γ obtenue par résolution de 1^o ordre.

Puisque la résolution du premier ordre est une combinaison des règles factorisation, copie et résolution binaire, nous avons immédiatement $\Gamma \vdash_{1r} C$ implique $\Gamma \vdash_{fcb} C$.

Théorème 5.4.19 (Théorème du relèvement) Soient C et D deux clauses. Soient C' une instance de C et D' une instance de D . Soit E' un résolvant propositionnel de C' et D' , il existe E un résolvant au premier ordre de C et D qui a pour instance E' .

Preuve : La preuve consiste essentiellement à construire E .

Puisque C' est une instance de C , il existe une substitution σ telle que $C' = C\sigma$.

Puisque D' est une instance de D , il existe une substitution τ telle que $D' = D\tau$.

Puisque E' est un résolvant propositionnel de C' et D' , il y a deux littéraux $l' \in C'$ et $m' \in D'$ tels que :

- l' et m' sont des littéraux opposés, autrement dit $l' = m'^c$.
- $E' = (C' - \{l'\}) \cup (D' - \{m'\})$.

Soit L l'ensemble de tous les littéraux de C tels que $L\sigma = \{l'\}$.

Soit M l'ensemble de tous les littéraux de D tels que $M\tau = \{m'\}$.

Puisque L est unifiable, il existe σ' un unificateur le plus général de L . Puisque σ' unifie L , il existe un littéral l tel que $L\sigma' = \{l\}$. Puisque σ' est l'unificateur le plus général de L , il existe σ'' tel que $\sigma = \sigma'\sigma''$, d'après la définition 5.3.9 page 121.

Puisque M est unifiable, il existe τ' un unificateur le plus général de M . Puisque τ' unifie M , il existe un littéral m tel que $M\tau' = \{m\}$. Puisque τ' est l'unificateur le plus général de M , il existe τ'' tel que $\tau = \tau'\tau''$, d'après la définition 5.3.9 page 121.

Soit ρ un renommage de $D\tau'$ tel que $D\tau'\rho$ est une copie de $D\tau'$ sans variable commune avec $C\sigma'$. Nous montrons qu'il y a un résolvant binaire E de $C\sigma'$ et $D\tau'\rho$ dont E' est une instance.

Montrons que l et $m^c\rho$ sont unifiables. Nous définissons de la façon suivante un unificateur π de ces deux littéraux. Soit x une variable quelconque :

- si x est une variable de $C\sigma'$ alors $x\pi = x\sigma''$,
- si x est une variable de $D\tau'\rho$ alors $x\pi = x\rho_{D\tau'}^{-1}\tau''$,
- sinon $x\pi = x$.

Notons tout d'abord que π est bien défini, car $C\sigma'$ et $D\tau'\rho$ n'ont pas de variable commune.

- Puisque les variables de $L\sigma'$ sont des variables de $C\sigma'$, nous avons $L\sigma'\pi = L\sigma'\sigma'' = L\sigma = \{l\}$. Puisque $L\sigma' = \{l\}$, nous avons $\{l\}\pi = \{l'\}$.
- Puisque les variables de $M\tau'\rho$ sont des variables de $D\tau'\rho$, nous avons $M\tau'\rho\pi = M\tau'\rho\rho_{D\tau'}^{-1}\tau''$. Puisque les variables de $M\tau'$ sont des variables de $D\tau'$, alors d'après la propriété du renommage inverse 5.4.8 page 125, nous avons $M\tau'\rho\rho_{D\tau'}^{-1} = M\tau'$. Par suite $M\tau'\rho\pi = M\tau'\tau'' = M\tau = \{m'\}$. Puisque $M\tau' = \{m\}$, nous avons $\{m\}\rho\pi = \{m\rho\}\pi = \{m'\}$.

Puisque $l' = m'^c$, la substitution π est bien, comme il a été annoncé ci-dessus, un unificateur de l et de $m^c\rho$. Par suite, il existe π' un unificateur le plus général de ces deux littéraux et il existe π'' tel que $\pi = \pi'\pi''$. Soit $E = ((C\sigma' - \{l\}) \cup (D\tau'\rho - \{m\rho\}))\pi'$.

- E est un résolvant binaire de $C\sigma'$ et de $D\tau'\rho$ sans variable commune avec $C\sigma'$. En effet $l \in C\sigma'$, $m\rho \in D\tau'\rho$ et $l\pi' = m^c\rho\pi'$. Puisque $C\sigma'$ est un facteur de C , que $D\tau'$ est un facteur de D et que $D\tau'\rho$ est une copie de $D\tau'$ sans variable commune avec $C\sigma'$, alors, d'après la définition 5.4.17 page précédente, E est un résolvant du premier ordre de C et D .

- Montrons que $E\pi'' = E'$, donc que E' est une instance de E . En effet par définition de E et le fait que $\pi = \pi'\pi''$, nous avons : $E\pi'' = ((C\sigma' - \{l\}) \cup (D\tau'\rho - \{m\rho\}))\pi = ((C\sigma'\pi - \{l\}\pi) \cup (D\tau'\rho\pi - \{m\rho\}\pi))$. Comme nous l'avons vu ci dessus : $\{l\}\pi = \{l'\}$ et $\{m\rho\}\pi = \{m'\}$. Puisque π est identique à σ'' sur les variable de $C\sigma'$, nous avons $C\sigma'\pi = C\sigma'\sigma'' = C\sigma = C'$. Puisque π est identique à $\rho_{D\tau'}^{-1}\tau''$ sur les variable de $D\tau'\rho$, nous avons $D\tau'\rho\pi = D\tau'\rho\rho_{D\tau'}^{-1}\tau''$. D'après les propriétés du renommage inverse 5.4.8 page 125, nous avons : $D\tau'\rho\rho_{D\tau'}^{-1}\tau'' = D\tau'\tau'' = D\tau = D'$. Donc $E\pi'' = (C' - \{l'\}) \cup (D' - \{m'\}) = E'$.

□

Exemple 5.4.20 Soient $C = P(x) \vee P(y) \vee R(y)$ et $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$.

- Les clauses $C' = P(a) \vee R(a)$ et $D' = \neg Q(a) \vee P(a) \vee \neg R(a)$ sont des instances respectivement de C et D .
- La clause $E' = P(a) \vee \neg Q(a)$ est un résolvant propositionnel de C' et D' .
- La clause $E = P(x) \vee \neg Q(x)$ est un résolvant au 1^o ordre de C et D qui a pour instance E' .

Théorème 5.4.21 Soient Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ , et C_1, \dots, C_n une preuve par résolution propositionnelle à partir de Δ , il existe une preuve D_1, \dots, D_n par résolution au 1^o ordre à partir de Γ telle que pour i de 1 à n , la clause C_i est une instance de D_i .

Preuve : Nous effectuons une récurrence sur n . Soit C_1, \dots, C_n, C_{n+1} une preuve par résolution propositionnelle à partir de Δ . Par récurrence, il existe une preuve D_1, \dots, D_n par résolution 1^o ordre à partir de Γ telle que pour i de 1 à n , la clause C_i est une instance de D_i . Nous distinguons deux cas :

1. Supposons $C_{n+1} \in \Delta$. Il existe $E \in \Gamma$ dont C_{n+1} est une instance donc nous prenons $D_{n+1} = E$.
2. Supposons que C_{n+1} soit un résolvant propositionnel de C_j et C_k où $j, k \leq n$. D'après le résultat précédent, il existe E résolvant au 1^o ordre de D_j et D_k : nous prenons $D_{n+1} = E$.

□

Nous en déduisons immédiatement le corollaire suivant.

Corollaire 5.4.22 Soient Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ . Supposons que $\Delta \vdash_p C$. Il existe D telle que $\Gamma \vdash_{1r} D$ et C est une instance de D .

Exemple 5.4.23 Soit l'ensemble de clauses $P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$. La fermeture universelle de cet ensemble de clauses est insatisfaisable et nous le montrons de trois manières.

1. Par instanciation sur le domaine de Herbrand $\{f^n(a), n \in \mathbb{N}\}$:

- $P(f(x)) \vee P(u)$ est instanciée par $x := a, u := f(a)$ en $P(f(a))$
- $\neg P(x) \vee Q(z)$ est instanciée par $x := f(a), z := a$ en $\neg P(f(a)) \vee Q(a)$
- $\neg Q(x) \vee \neg Q(y)$ est instanciée par $x := a, y := a$ en $\neg Q(a)$

L'ensemble de ces 3 instanciations est insatisfaisable, comme le montre la preuve par résolution propositionnelle ci-dessous :

$$\frac{\frac{P(f(a)) \quad \neg P(f(a)) \vee Q(a)}{Q(a)} \quad \neg Q(a)}{\perp}$$

2. Cette preuve par résolution propositionnelle est relevée en une preuve par la règle de résolution au premier ordre :

$$\frac{\frac{P(f(x)) \vee P(u) \quad \neg P(x) \vee Q(z)}{Q(z)} \quad \neg Q(x) \vee \neg Q(y)}{\perp}$$

3. Chaque règle de résolution au premier ordre est décomposée en factorisation, copie et résolution binaire :

$$\frac{\frac{\frac{P(f(x)) \vee P(u)}{P(f(x))} \text{ Fact} \quad \frac{\neg P(x) \vee Q(z)}{\neg P(y) \vee Q(z)} \text{ Copie}}{Q(z)} \text{ RB} \quad \frac{\neg Q(x) \vee \neg Q(y)}{\neg Q(x)} \text{ Fact}}{\perp} \text{ RB}$$

Théorème 5.4.24 (Complétude réfutationnelle de la résolution au 1^o ordre) Soit Γ un ensemble de clauses. Les propositions suivantes sont équivalentes :

1. $\Gamma \vdash_{1r} \perp$.
2. $\Gamma \vdash_{1fcb} \perp$.
3. $\forall(\Gamma) \models \perp$.

Preuve :

- Nous savons que (1) implique (2), car la résolution au 1^o ordre est une combinaison de factorisation, copie et résolution binaire.
- Nous savons que (2) implique (3), car la factorisation, la copie et la résolution binaire sont cohérentes.
- Il nous reste à montrer que (3) implique (1). Supposons que $\forall(\Gamma) \models \perp$, autrement dit $\forall(\Gamma)$ est insatisfaisable. D'après le théorème de Herbrand, il y a Δ un ensemble fini d'instances sans variable de clauses de Γ qui n'a pas de modèle propositionnel. Par complétude de la résolution propositionnelle, nous avons : $\Delta \vdash_p \perp$. D'après le corollaire au relèvement 5.4.22 page 128, il existe D telle que $\Gamma \vdash_{1r} D$ et \perp est instance de D . Dans ce cas, nous avons $D = \perp$.

□

5.5 Exercices

Exercice 78 (Domaine de Herbrand) Soient Σ la signature composée de la constante a et des symboles de fonctions f et g respectivement à un et deux arguments.

- Donner 5 éléments différents du domaine de Herbrand de cette signature.
- Définir inductivement ce domaine.

□

Exercice 79 (Signature, domaine et base de Herbrand) Pour chacun de ces ensembles de formules :

- $\Gamma_1 = \{P(x) \vee Q(x) \vee R(x), \neg P(a), \neg Q(b), \neg R(c)\}.$
- $\Gamma_2 = \{P(x), \neg Q(x), \neg P(f(x)) \vee Q(f(x))\}.$
- $\Gamma_3 = \{P(x), Q(f(x)), \neg R(f(f(x))), \neg P(f(f(x))) \vee \neg Q(x) \vee R(f(x))\}.$

1. Donner la signature, le domaine de Herbrand ainsi que la base de Herbrand.
2. Prouver si leur fermeture universelle a un modèle ou pas.

□

Exercice 80 (Modèle de Herbrand,*)** Soit Δ l'ensemble de formules suivant :

1. $x < y \wedge y < z \Rightarrow x < z.$
2. $\neg(x < x).$
3. $x < y \Rightarrow x < f(x, y) \wedge f(x, y) < y.$
4. $a < b.$

- Indiquer un modèle de $\forall(\Delta).$
- $\forall(\Delta)$ a-t-il un modèle sur le domaine de Herbrand engendré par a, b, f ?

□

Exercice 81 (Skolémisation et forme clausale) Skolémiser les formules suivantes (attention aux négations !) puis les mettre en forme clausale.

1. $\neg(\exists x P(x) \vee \exists x Q(x) \Rightarrow \exists x (P(x) \vee Q(x))).$
2. $\neg(\forall x \forall y \forall z (e(x, y) \wedge e(y, z) \Rightarrow \neg e(x, z)) \Rightarrow \neg \exists x \forall y e(x, y)).$
3. $\neg(\neg \forall x P(x) \vee \neg \forall x Q(x) \Rightarrow \neg(\forall x P(x) \wedge \forall x Q(x))).$
4. $\forall x ((\exists y P(x, y) \Rightarrow \exists x Q(x)) \wedge \exists y P(x, y) \wedge \neg \exists x Q(x)).$
5. $\neg(\exists x \forall y \forall z ((P(y) \Rightarrow Q(z)) \Rightarrow (P(x) \Rightarrow Q(x))))$

□

Exercice 82 (Unification) Les termes suivants sont-ils unifiables ? Si la réponse est oui, donner leur unificateur le plus général, sinon justifier la réponse négative.

- $h(g(x), f(a, y), z)$ et $h(y, z, f(u, x)).$
- $h(g(x), f(a, y), z)$ et $h(y, z, f(u, g(x))).$

□

Exercice 83 (Unification) Donner les unificateurs les plus généraux des termes suivants s'ils existent :

1. $\text{pair}(a, \text{crypt}(z, b))$ et $\text{pair}(x, y).$
2. $\text{pair}(\text{crypt}(x, b), \text{crypt}(y, b))$ et $\text{pair}(\text{crypt}(a, b), z).$
3. $\text{crypt}(\text{pair}(z, a), x)$ et $\text{crypt}(\text{pair}(y, \text{crypt}(x, b)), b).$
4. $\text{crypt}(\text{pair}(a, z), x)$ et $\text{crypt}(\text{pair}(y, \text{crypt}(x, b)), b).$
5. $f(x, y, g(a, a))$ et $f(g(y, y), z, z)$
6. $f(x, y, a)$ et $f(y, g(z, z), x)$

□

Exercice 84 (Unification avec plusieurs solutions) L'équation $f(g(y), y) = f(u, z)$ a deux solutions « les plus générales » (rappel : elles sont donc équivalentes). Indiquer ces deux solutions.

□

Exercice 85 (Forme clausale et instanciation,*) Montrer que la formule suivante est valide en mettant sa négation sous la forme clausale et en trouvant un ensemble contradictoire d'instances des clauses obtenues :

$$\forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall xP(x) \Rightarrow \forall xQ(x)).$$

□

Exercice 86 (Forme clausale et preuve par résolution,)** Considérons les formules suivantes :

1. $H_1 = \exists xP(x) \Rightarrow \forall xP(x)$.
2. $H_2 = \forall x(P(x) \vee Q(x))$.
3. $C = \exists x\neg Q(x) \Rightarrow \forall xP(x)$.

Nous voulons montrer que C est conséquence de H_1 et H_2 par instanciation et par résolution.

1. Mettre en forme clausale l'ensemble des trois formules H_1 , H_2 , $\neg C$.
2. Trouver des instances contradictoires des clauses obtenues et montrer par résolution propositionnelle que ces instances sont contradictoires.
3. Donner une preuve directe de cette contradiction par factorisation, copie et résolution binaire. Il est possible que seule la dernière règle soit utilisée.

□

Exercice 87 (Preuve par résolution,)**

- En utilisant une preuve par factorisation, copie et résolvant binaire prouver que la fermeture universelle de l'ensemble de clauses suivant est insatisfaisable :

1. $P(f(x)) \vee \neg Q(y, a)$.
2. $Q(a, a) \vee R(x, x, b) \vee S(a, b)$.
3. $S(a, z) \vee \neg R(x, x, b)$.
4. $\neg P(f(c)) \vee R(x, a, b)$.
5. $\neg S(y, z) \vee \neg S(a, b)$.

- En utilisant une preuve par factorisation, copie et résolvant binaire prouver que la fermeture universelle de l'ensemble de clauses suivant est insatisfaisable :

1. $P(x)$.
2. $\neg P(y) \vee Q(y, x)$.
3. $\neg Q(x, a) \vee \neg Q(b, y) \vee \neg Q(b, a) \vee \neg P(f(y))$.

- En utilisant une preuve par factorisation, copie et résolvant binaire prouver que la fermeture universelle de l'ensemble de clauses suivant est insatisfaisable :

1. $x = a$.
2. $y = b$.
3. $z = a + b$.
4. $\neg(b < x + b) \vee (a + b < a)$.
5. $b < z$.
6. $\neg(a + b < x) \vee \neg(y = b)$.

Rappel : le symbole $+$ est plus prioritaire que le symbole $<$, qui est lui-même plus prioritaire que les connecteurs logiques binaires.

□

Exercice 88 (Unification, résolution) Soient $\Gamma_1 = \{P(x, f(x, b), u), \neg P(g(a), z, h(z))\}$ l'ensemble des deux clauses unitaires et Γ_2 l'ensemble des deux clauses unitaires $\{P(x, f(x, b), u), \neg P(g(z), z, h(z))\}$. Les ensembles $\forall(\Gamma_1)$ et $\forall(\Gamma_2)$ sont-ils satisfaisables ou insatisfaisables ?

□

Exercice 89 (Skolémisation et preuve par résolution) Le but de cet exercice est de démontrer le syllogisme suivant :

$$\forall x(\text{homme}(x) \Rightarrow \text{mortel}(x)) \wedge \text{homme}(\text{Socrate}) \Rightarrow \text{mortel}(\text{Socrate})$$

- Skolémiser la **négarion** du syllogisme.
- Transformer la forme de Skolem obtenue en forme clause.
- Démontrer par instanciation que la **négarion** du syllogisme est une contradiction.
- Démontrer en utilisant une preuve par factorisation, copie et résolution binaire que la négation du syllogisme est une contradiction.

□

Exercice 90 (Forme clause et preuve par résolution,)** Considérons les formules suivantes :

1. $A_1 = \exists u \forall v (P(u) \wedge (R(v) \Rightarrow Q(u, v)))$.
2. $A_2 = \forall u \forall v (\neg P(u) \vee \neg S(v) \vee \neg Q(u, v))$.
3. $A_3 = \exists v (R(v) \wedge S(v))$.

Montrer que la liste de ces trois formules est contradictoire par résolution :

1. Mettre en forme clause les trois formules A_1, A_2, A_3 .
2. Trouver des instances contradictoires des clauses obtenues et montrer cette contradiction par résolution propositionnelle.
3. Faire une preuve directe de \perp par factorisation, copie et résolution binaire.

□

Exercice 91 (\Leftrightarrow Cohérence de la résolution au premier ordre) Prouver la cohérence de la résolution au premier ordre, c'est-à-dire :

Soit Γ un ensemble de clauses et C une clause. Si $\Gamma \vdash_{1fc} C$ alors $\forall(\Gamma) \models \forall(C)$.

□

Exercice 92 (Forme clause et preuve par résolution,)** Considérons les formules suivantes :

1. $E_1 = \forall x (x = x)$.
2. $E_2 = \forall x \forall y (x = y \Rightarrow y = x)$.
3. $E_3 = \forall x \forall y \forall z \forall t (x = y \wedge z = t \Rightarrow (x \in z \Rightarrow y \in t))$.
4. $C = \forall y \forall z (y = z \Rightarrow \forall x (x \in y \Leftrightarrow x \in z))$.

E_1, E_2, E_3 sont des axiomes de l'égalité : E_1 exprime que l'égalité est réflexive, E_2 qu'elle est symétrique, E_3 que c'est une congruence relativement à l'appartenance. La transitivité de l'égalité ne sert pas dans cet exercice. C exprime que deux ensembles égaux ont les mêmes éléments. Nous notons que dans cet exercice, le symbole égal est traité comme un symbole de prédicat « ordinaire » dont le sens n'est pas fixé comme étant l'identité, mais que nous « définissons » par ses propriétés. Autrement dit l'interdiction de l'égalité dans les formules, c'est en réalité l'interdiction de l'égalité comme symbole de sens fixé. Montrer que C est conséquence des axiomes de l'égalité.

1. Mettre en forme clause les trois formules $E_1, E_2, E_3, \neg C$.
2. Trouver des instances contradictoires des clauses obtenues.
3. Faire une preuve directe de \perp par factorisation, copie et résolution binaire.

□

Exercice 93 (Forme clause et preuve par résolution,)** Considérons les formules suivantes :

1. $H_1 = \forall u (\exists v R(u, v) \Rightarrow R(u, f(u)))$.
2. $H_2 = \forall u \exists v R(u, v)$.
3. $H_3 = \exists u R(f(f(u)), u)$.
4. $C = \exists u \exists v \exists w (R(u, v) \wedge R(v, w) \wedge R(w, u))$.

Montrer que C est conséquence de H_1, H_2, H_3 .

1. Mettre en forme clause les trois formules $H_1, H_2, H_3, \neg C$.
2. Trouver des instances contradictoires des clauses obtenues (montrez la contradiction par résolution propositionnelle).
3. Faire une preuve directe de cette contradiction par factorisation, copie et résolution binaire.

□

Exercice 94 (Forme clausale et preuve par résolution,)** Soit $P(x,y)$ une formalisation de « x est père de y ». Soit $A(x,y)$ une formalisation de « x est aïeul de y ». Soient les trois formules :

1. $H_1 = \forall x \exists y P(x,y)$.
2. $H_2 = \forall x \forall y \forall z (P(x,y) \wedge P(y,z) \Rightarrow A(x,z))$.
3. $C = \forall x \exists y A(x,y)$.

Montrons que C est conséquence de H_1, H_2 , de la façon suivante, transformer $H_1, H_2, \neg C$ en leur forme clausale, puis trouver des instances contradictoires des clauses obtenues (montrez la contradiction par résolution propositionnelle). Enfin, dériver la clause vide par factorisation, copie, résolution binaire. \square

Exercice 95 (Formule de Bernays-Schonfinkel,*)** Une formule BS (de Bernays-Schonfinkel) est une formule fermée, sans symbole de fonction, de la forme $\exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_p B$ où B est sans quantificateur. Expliquer pourquoi nous pouvons décider la satisfaisabilité d'une telle formule. \square

Chapitre 6

Déduction naturelle au premier ordre : quantificateurs, copie et égalité

Sommaire

6.1 Règles pour la logique du premier ordre	135
6.1.1 Règles des quantificateurs	135
6.1.2 Copie	139
6.1.3 Les règles de l'égalité	139
6.2 Tactiques de preuves	140
6.2.1 Raisonner en avant avec une hypothèse d'existence	140
6.2.2 Raisonner en arrière pour généraliser	140
6.2.3 Un exemple d'application des tactiques	140
6.3 Cohérence du système	142
6.4 Exercices	144

EN plus des règles du chapitre 3, nous ajoutons les règles concernant les quantificateurs, la copie et l'égalité. Notre système comporte toujours une seule règle pour enlever les hypothèses (la règle d'introduction de l'implication). Les définitions de brouillon de preuve, environnement, contexte, formule utilisable restent inchangées. Nous montrons la cohérence des règles de notre système, mais nous admettons sans preuve, que ce système est complet : des preuves de complétude pour des systèmes de règles similaires peuvent être trouvées dans les livres [?, ?]. Contrairement au cas propositionnel, *il n'y a aucun algorithme* pour décider si une formule est valide ou non valide (preuve donnée par Alonzo Church et Alan Turing en 1936 et 1937 [?, ?]). Autrement dit, en admettant l'équivalence entre prouvable (sans environnement) et valide, il n'y a pas d'algorithme qui, étant donné une formule, puisse nous en construire la preuve, ou nous avertir que cette formule n'a pas de preuve.

Plan : Nous étendons d'abord les règles de la déduction naturelle proposées pour la logique propositionnelle. Nous présentons ensuite deux nouvelles tactiques pour aider à la rédaction de preuves. Enfin, nous prouvons la cohérence de notre système.

6.1 Règles pour la logique du premier ordre

Notre système de règles se divise en plusieurs familles de règles :

- les règles « propositionnelles »,
- les règles d'introduction et d'élimination de quantificateur,
- les règles régissant l'égalité,
- et une règle de copie.

L'ensemble des règles définies dans la section 3 page 65 peuvent être utilisées dans notre système de déduction. Nous renvoyons le lecteur à la table 3.1 page 67, qui récapitule l'ensemble de ces règles. Dans la suite de cette section, nous présentons uniquement les règles de déduction naturelle dédiées à la logique du premier ordre.

6.1.1 Règles des quantificateurs

L'ensemble des règles d'introduction et d'élimination de quantificateur est présenté dans la table 6.1. Dans cette table, A et B sont des formules, x est une variable, t est un terme. Notez que contrairement aux règles « propositionnelles », l'usage de ces nouvelles règles est contraint par des conditions d'emploi, qui utilisent les notions de *variable libre* (Définition 4.2.3 page 88) et de *terme libre pour une variable* (Définition 4.3.34 page 94).

Dans la suite, nous détaillons ces règles et nous illustrons leur usage sur des exemples et des contre-exemples montrant les erreurs occasionnées par le non respect des conditions d'emploi.

Règles	Conditions d'emploi
$\frac{\forall xA}{A<x:=t>} \forall E$	t est libre pour x dans A .
$\frac{A}{\forall xA} \forall I$	x ne doit être libre ni dans l'environnement de la preuve, ni dans le contexte de la prémisse de la règle.
$\frac{\exists xA \quad (A \Rightarrow B)}{B} \exists E$	x ne doit être libre ni dans l'environnement, ni dans B , ni dans le contexte de la prémisse droite de la règle.
$\frac{A<x:=t>}{\exists xA} \exists I$	t est libre pour x dans A .

TABLE 6.1 – Règles d'introduction et d'élimination des quantificateurs.

6.1.1.1 Règles pour le quantificateur universel

La règle d'élimination du quantificateur universel, notée $\forall E$, signifie que si la formule A est vraie pour toute valeur de x alors toute instance de A où x est remplacé par **un terme t libre pour x** est vraie. Notez que cette règle est une simple application du corollaire 4.3.38, page 94.

$$\frac{\forall xA}{A<x:=t>} \forall E$$

Nous justifions ci-dessous les conditions d'emploi de cette règle par un exemple montrant qu'un usage incorrect de la règle peut conduire à prouver une formule non-valide.

Exemple 6.1.1 Nous montrons qu'un usage incorrect de la **règle** $\forall E$ conduit à « prouver » une formule non valide.

contexte	numéro	ligne	règle
1	1	Supposons $\forall x \exists y P(x, y)$	
1	2	$\exists y P(y, y)$	$\forall E$ 1, y ERREUR
	3	Donc $\forall x \exists y P(x, y) \Rightarrow \exists y P(y, y)$	

À la ligne 2, nous n'avons pas respecté les conditions d'applications de la règle $\forall E$ car le terme y n'est pas libre pour x dans la formule $\exists y P(x, y)$. Nous donnons une interprétation qui est un contre-modèle de la « conclusion » : soit I

l'interprétation de domaine $\{0, 1\}$ avec $P_I =$

$\{(0, 1), (1, 0)\}$, I rend la « conclusion » fausse.

La règle d'introduction du quantificateur universel, notée $\forall I$, signifie que si nous avons pu déduire la formule A indépendamment de la valeur de x , nous pouvons généraliser en déduisant $\forall xA$. Ainsi, il est nécessaire que x **ne soit libre ni dans l'environnement de la preuve, ni dans le contexte de la ligne où nous avons déduit A .**

$$\frac{A}{\forall xA} \quad \forall I$$

Nous illustrons maintenant l'emploi de la règle d'introduction du quantificateur universel avec un exemple d'usage correct (exemple 6.1.2) et un exemple d'usage incorrect (exemple 6.1.3) de la règle.

Exemple 6.1.2 *Nous prouvons $\forall yP(y) \wedge \forall yQ(y) \Rightarrow \forall x(P(x) \wedge Q(x))$.*

contexte	numéro	ligne	règle
1	1	Supposons $\forall yP(y) \wedge \forall yQ(y)$	
1	2	$\forall yP(y)$	$\wedge E1$ 1
1	3	$\forall yQ(y)$	$\wedge E2$ 1
1	4	$P(x)$	$\forall E$ 2, x
1	5	$Q(x)$	$\forall E$ 3, x
1	6	$P(x) \wedge Q(x)$	$\wedge I$ 4,5
1	7	$\forall x(P(x) \wedge Q(x))$	$\forall I$ 6
	8	Donc $\forall yP(y) \wedge \forall yQ(y) \Rightarrow \forall x(P(x) \wedge Q(x))$	$\Rightarrow I$ 1,7

Notez que dans l'usage de la règle d'instanciation aux lignes 4 et 5, nous avons précisé que y est remplacé par x .

Exemple 6.1.3 *Nous montrons qu'un usage incorrect de la règle $\forall I$ conduit à « prouver » une formule non valide.*

contexte	numéro	ligne	règle
1	1	Supposons $P(x)$	
1	2	$\forall xP(x)$	$\forall I$ 1 ERREUR
	3	Donc $P(x) \Rightarrow \forall xP(x)$	$\Rightarrow I$ 1,2

À la ligne 2, nous n'avons pas respecté les conditions d'applications de la règle $\forall I$ car la prémisse $P(x)$ est établie dans le contexte $P(x)$, ce qui interdit de généraliser sur x . Nous donnons un contre-modèle de la « conclusion » :

soient I l'interprétation de domaine $\{0, 1\}$ avec $P_I = \{0\}$ et e un état où $x = 0$, l'assignation (I, e) rend la « conclusion » fausse.

6.1.1.2 Règles pour le quantificateur existentiel

La règle d'élimination du quantificateur existentiel, notée $\exists E$, signifie que si $\exists xA$ est vraie et que nous pouvons déduire $A \Rightarrow B$ indépendamment de la valeur de x , alors nous pouvons déduire B , si la formule B ne dépend, elle non plus, de la

valeur de x . Ainsi, il est nécessaire que x **ne soit libre ni dans l'environnement de la preuve, ni dans B , et ni dans le contexte de la ligne $A \Rightarrow B$.**

$$\frac{\exists xA \quad (A \Rightarrow B)}{B} \exists E$$

La condition d'emploi de cette règle étant complexe, nous illustrons l'emploi de la règle avec deux exemples d'usage incorrect.

Exemple 6.1.4 Nous montrons qu'un usage incorrect de la règle $\exists E$ conduit à « prouver » une formule non valide.

contexte	numéro	ligne	règle
1	1	Supposons $\exists xP(x) \wedge (P(x) \Rightarrow \forall yQ(y))$	
1	2	$\exists xP(x)$	$\wedge E1$ 1
1	3	$P(x) \Rightarrow \forall yQ(y)$	$\wedge E2$ 1
1	4	$\forall yQ(y)$	$\exists E$ 2,3 ERREUR
	5	Donc $\exists xP(x) \wedge (P(x) \Rightarrow \forall yQ(y)) \Rightarrow \forall yQ(y)$	$\Rightarrow I$ 1,4

Nous n'avons pas respecté la condition que le contexte de la prémisses $P(x) \Rightarrow \forall yQ(y)$ ne doit pas dépendre de x . Il est clair que la conclusion obtenue n'est pas valide. Nous donnons une assignation (I, e) qui est un contre-modèle de cette « conclusion » :

soient I l'interprétation de domaine $\{0, 1\}$ avec $P_I = Q_I = \{0\}$ et l'état e où $x = 1$, (I, e) rend la « conclusion » fausse.

Exemple 6.1.5 Nous montrons qu'un usage incorrect de la règle $\exists E$ conduit à « prouver » une formule non valide.

contexte	numéro	ligne	règle
1	1	Supposons $\exists xP(x)$	
1,2	2	Supposons $P(x)$	
1	3	Donc $P(x) \Rightarrow P(x)$	$\Rightarrow I$ 2,2
1	4	$P(x)$	$\exists E$ 1,3 ERREUR
1	5	$\forall xP(x)$	$\forall I$ 4
	6	Donc $\exists xP(x) \Rightarrow \forall xP(x)$	

La conclusion de la règle $\exists E$ est $P(x)$, contrairement à la condition d'application de cette règle qui impose que la conclusion ne doit pas dépendre de x . Nous donnons une interprétation qui est un contre-modèle de la cette « conclusion » :

soit I l'interprétation de domaine $\{0, 1\}$ avec $P_I = \{0\}$, I rend fausse la « conclusion ».

La règle d'introduction du quantificateur existentiel, notée $\exists I$, signifie que si une instance de la formule A où x est remplacé par **un terme t libre pour x** est vraie, alors la formule $\exists xA$ est vraie. Notez que cette règle est une simple application du corollaire 4.3.38, page 94. Ce qui justifie la condition d'emploi. Pour s'en convaincre, il suffit d'étudier l'exemple 4.3.37, qui précède le corollaire 4.3.38.

Ci-dessous, nous donnons un exemple d'usage correct de la règle d'introduction du quantificateur existentiel en démontrant l'une des lois de De Morgan.

Exemple 6.1.6 (Lois de De Morgan) Nous montrons que pour toute formule A , $\neg \forall xA \Rightarrow \exists x\neg A$.

contexte	numéro	ligne	règle
1	1	Supposons $\neg\forall xA$	
1,2	2	Supposons $\neg\exists x\neg A$	
1,2,3	3	Supposons $\neg A$	
1,2,3	4	$\exists x\neg A$	$\exists I\ 3, x$
1,2,3	5	\perp	$\Rightarrow E\ 2,4$
1,2	6	Donc $\neg\neg A$	$\Rightarrow I\ 3,5$
1,2	7	A	$RAA\ 6$
1,2	8	$\forall xA$	$\forall I\ 7$
1,2	9	\perp	$\Rightarrow E\ 1,8$
1	10	Donc $\neg\neg\exists x\neg A$	$\Rightarrow I\ 2,9$
1	11	$\exists x\neg A$	$RAA\ 10$
	12	Donc $\neg\forall xA \Rightarrow \exists x\neg A$	$\Rightarrow I\ 1,11$

Nous vérifions que les conditions d'application des règles sont respectées.

- À la ligne 4, nous avons utilisé le fait que $\neg A$ peut être vue comme le résultat de la substitution de x par x dans $\neg A$ et qu'une variable est toujours libre pour elle-même dans une formule.
- À la ligne 7, le contexte de la prémisse ne dépend pas de x , puisque dans ce contexte A et $\neg A$ sont dans la portée d'un quantificateur liant x .

6.1.2 Copie

La règle de *copie* consiste à déduire d'une formule A , une autre formule A' égale au changement près des variables liées, au sens de la définition 4.4.5, page 100. Par exemple, $\forall x\exists yP(x,y)$ est une copie de $\forall y\exists xP(y,x)$.

$$\frac{A'}{A} \text{ Copie}$$

Notez qu'il n'y pas de condition d'emploi pour cette règle.

6.1.3 Les règles de l'égalité

Deux règles caractérisent l'égalité : un terme est égal à lui-même (règle de la *réflexivité*) et si deux termes sont égaux, nous pouvons les remplacer l'un par l'autre (règle de la *congruence*).

$\frac{}{t=t}$ <i>Réflexivité</i>	t est un terme
$\frac{s=t \quad A\langle x:=s \rangle}{A\langle x:=t \rangle}$ <i>Congruence</i>	s et t sont deux termes libres pour la variable x dans la formule A

Nous remarquons que la première règle n'a pas de prémisse. C'est ce que nous appelons aussi un *axiome*. Nous remarquons aussi que les conditions d'emploi de la deuxième règle sont similaires à celles des règles $\forall E$ et $\exists I$. Ces conditions d'emploi se justifient de la même manière que précédemment.

Nous donnons maintenant deux exemples d'application des règles d'égalités.

Exemple 6.1.7 Prouvons que $s = t \Rightarrow t = s$, autrement dit prouvons que l'égalité est symétrique.

contexte	numéro	ligne	règle
1	1	Supposons $s = t$	
1	2	$s = s$	<i>Réflexivité</i>
1	3	$t = s$	<i>Congruence</i> 1,2, $(s = s) = (x = s)\langle x:=s \rangle$ et $(t = s) = (x = s)\langle x:=t \rangle$
	4	Donc $s = t \Rightarrow t = s$	$\Rightarrow I\ 1,3$

Notons que la variable x ne figure pas dans la preuve, elle ne fait que marquer l'endroit où fait le remplacement de s par t . Dans les prochains exemples, nous nous contenterons de souligner cet endroit.

Exemple 6.1.8 Prouvons que $s = t \wedge t = u \Rightarrow s = u$, autrement dit prouvons que l'égalité est transitive.

I	I	Supposons $s = t \wedge t = u$	
I	2	$s = t$	$\wedge E1\ I$
I	3	$t = u$	$\wedge E2\ I$
I	4	$s = \underline{u}$	Congruence 3,2
	5	Donc $s = t \wedge t = u \Rightarrow s = u$	$\Rightarrow I\ 1,4$

6.2 Tactiques de preuves

Nous introduisons deux nouvelles tactiques pour l'application des règles $\forall I$ et $\exists E$ et nous illustrons ces tactiques avec un exemple.

6.2.1 Raisonner en avant avec une hypothèse d'existence

Soient Γ un ensemble de formules, x une variable, A et C des formules. Supposons que nous cherchons une preuve de C dans l'environnement $\Gamma, \exists xA$.

1. Supposons que x n'est libre ni dans Γ , ni dans C . Dans ce cas, la preuve peut toujours s'écrire :

Supposons A

preuve de C dans l'environnement Γ, A

Donc $A \Rightarrow C \Rightarrow I\ 1, _$

$C \quad \exists E$

2. Supposons que x soit libre dans Γ ou C . Nous choisissons une variable y « nouvelle », c'est-à-dire non libre dans Γ, C et absente de A , puis nous nous ramenons au cas précédent, via la règle de copie. La preuve s'écrit alors :

$\exists yA \langle x := y \rangle \quad \text{Copie de } \exists xA$

Supposons $A \langle x := y \rangle$

preuve de C dans l'environnement $\Gamma, A \langle x := y \rangle$

Donc $A \langle x := y \rangle \Rightarrow C \Rightarrow I\ 1, _$

$C \quad \exists E$

La recherche de la preuve initiale a été réduite à la recherche d'une preuve dans un environnement plus simple. C'est exactement le mode de raisonnement appliqué dans les cours de mathématiques quand nous cherchons une preuve d'une formule C avec l'hypothèse $\exists xP(x)$. Nous introduisons une constante « nouvelle » a vérifiant $P(a)$ et nous prouvons C sous l'hypothèse $P(a)$.

6.2.2 Raisonner en arrière pour généraliser

Nous reprenons les notations du paragraphe précédent. Supposons que nous cherchons une preuve de $\forall xA$ dans l'environnement Γ .

1. Supposons que x n'est pas libre dans Γ . Dans ce cas, la preuve peut toujours s'écrire :

preuve de A dans l'environnement Γ

$\forall xA \quad \forall I$

2. Supposons que x est libre dans Γ . Nous choisissons une variable y « nouvelle », c'est-à-dire non libre dans Γ , puis nous nous ramenons au cas précédent, via la règle de copie. La preuve s'écrit alors :

preuve de $A \langle x := y \rangle$ dans l'environnement Γ	
$\forall y A \langle x := y \rangle$	$\forall I$
$\forall x A$	Copie de la formule précédente

La recherche de la preuve initiale a été réduite à la recherche d'une preuve d'une formule plus simple dans le même environnement. C'est exactement le mode de raisonnement appliqué dans les cours de mathématiques quand nous cherchons une preuve de $\forall x P(x)$. Nous introduisons une constante « nouvelle » a et nous prouvons $P(a)$. Puis nous ajoutons : puisque le choix de a est arbitraire, nous avons $\forall x P(x)$.

6.2.3 Un exemple d'application des tactiques

Nous notons « il existe un et un seul x » par $\exists!x$. Formellement, $\exists!x P(x)$ signifie $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$. En séparant l'existence de x et son unicité, nous pouvons aussi définir $\exists!x P(x)$ par $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$. Ces deux définitions sont bien sûr équivalentes et nous montrons formellement que la première implique la deuxième. Comme la preuve est longue, il faut apprendre à décomposer les preuves.

Plan de la preuve Nous appliquons les deux tactiques suivantes :

- Pour prouver $A \Rightarrow B$, supposer A et déduire B
- Pour prouver $A \wedge B$, prouver A et prouver B .

1	Supposons $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
	preuve de $\exists x P(x)$ dans l'environnement de (1)	
	preuve de $\forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$ dans l'environnement de (1)	
	$\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$	$\wedge I$
	Donc $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow I$

Application de la tactique utilisant une hypothèse d'existence

Nous cherchons une preuve de $\exists x P(x)$ dans l'environnement de $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$. Nous appliquons la tactique « raisonner en avant en présence d'une hypothèse existentielle ».

référence	formule	
i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	Supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
2	$P(x)$	$\wedge E 1$
3	$\exists x P(x)$	$\exists I 2, x$
4	Donc $P(x) \wedge \forall y(P(y) \Rightarrow x = y) \Rightarrow \exists x P(x)$	$\Rightarrow I 1, 2$
5	$\exists x P(x)$	$\exists E i, 3$

Application de la tactique pour obtenir une conclusion générale

Nous cherchons une preuve de $\forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$ dans l'environnement de $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$. Nous appliquons, dans l'ordre ci-dessous, les tactiques suivantes :

- « raisonner en avant en utilisant d'une hypothèse existentielle ».
- Pour prouver $A \Rightarrow B$, supposer A et déduire B
- « raisonner en arrière pour obtenir une conclusion générale ».

environnement			
référence		formule	
i		$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
contexte	numéro	preuve	règle
1	1	Supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	Supposons $P(u) \wedge P(y)$	
1,2	3	$\forall y(P(y) \Rightarrow x = y)$	$\wedge E$ 1
1,2	4	$P(u)$	$\wedge E$ 2
1,2	5	$P(u) \Rightarrow x = u$	$\forall E$ 3, u
1,2	6	$x = u$	$\Rightarrow E$ 4,5
1,2	7	$P(y)$	$\wedge E$ 2
1,2	8	$P(y) \Rightarrow x = y$	$\forall E$ 3, y
1,2	9	$x = y$	$\Rightarrow E$ 7,8
1,2	10	$\underline{u} = y$	Congruence 6,9
1	11	Donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow I$ 2,10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall I$ 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall I$ 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	Copie de 13
1	15	Donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow I$ 1,14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists E$ i, 15

La tactique « utiliser une hypothèse d'existence » sert à produire la ligne 16. La tactique « introduire une implication » sert à produire la ligne 15 : elle introduit l'hypothèse (1) dans laquelle x est une variable libre. Donc, pour appliquer la tactique « obtenir une conclusion générale », il faut changer de variable dès la ligne 2. Observer aussi que la formule, ligne 3, est instanciée aux lignes 5 et 8.

Comme sur cet exemple, toute la difficulté des preuves est concentrée autour des règles $\forall E$ et $\exists I$:

- dans le raisonnement en avant, il faut trouver les bonnes instanciations des formules commençant par un quantificateur existentiel.
- dans le raisonnement en arrière, il faut trouver la bonne instance permettant de déduire une formule commençant par un quantificateur universel.

6.3 Cohérence du système

Nous commençons par montrer deux propriétés à propos des quantificateurs existentiel et universel. Enfin nous montrons la cohérence de notre système de déduction.

Propriété 6.3.1 Soient Γ un ensemble de formules, x une variable et A une formule. Supposons que x ne soit pas libre dans Γ , alors nous avons : $\Gamma \models A$ si et seulement si $\Gamma \models \forall x A$.

Preuve :

- \Rightarrow Supposons que $\Gamma \models A$. Soit (I, e) une assignation modèle de Γ . Puisque x n'est pas libre dans Γ , pour tout état f identique à e sauf pour la valeur de x , (I, f) et (I, e) donnent la même valeur aux formules de Γ , donc (I, f) est modèle de Γ . Puisque $\Gamma \models A$, pour tout état f identique à e sauf pour la valeur de x , (I, f) est modèle de A , donc (I, e) est modèle de $\forall x A$.
- \Leftarrow Supposons que $\Gamma \models \forall x A$. Puisque la formule $\forall x A \Rightarrow A$ est valide (d'après le corollaire 4.3.38), nous avons $\Gamma \models A$.

□

Si nous observons d'un œil critique cette preuve, nous observons que c'est une paraphrase, dans un autre formalisme, de la loi $\forall I$. C'est l'équivalence ci-dessus, qui explique que la tactique « raisonner en arrière pour généraliser », est une tactique utilisable sans risque. En effet, en admettant la complétude du système, lorsque x n'est pas libre dans Γ , il y a une preuve de A dans l'environnement Γ si et seulement s'il y a une preuve de $\forall xA$ dans ce même environnement.

Propriété 6.3.2 Soient Γ un ensemble de formules, x une variable, A et B deux formules. Supposons que x ne soit libre ni dans Γ , ni dans B , alors nous avons : $\Gamma \models A \Rightarrow B$ si et seulement si $\Gamma \models (\exists xA) \Rightarrow B$

Preuve :

- \Rightarrow Supposons que $\Gamma \models A \Rightarrow B$. Soit (I, e) une assignation modèle de Γ . Puisque x n'est pas libre dans Γ , pour tout état f identique à e sauf pour la valeur de x , (I, f) et (I, e) donnent la même valeur aux formules de Γ , donc (I, f) est modèle de Γ . Puisque $\Gamma \models A \Rightarrow B$, pour tout état f identique à e sauf pour la valeur de x , (I, f) est modèle de $A \Rightarrow B$. Supposons que (I, e) est modèle de $\exists xA$, il existe g identique à e sauf pour la valeur de x tel que (I, g) est modèle de A . Puisque pour tout état f identique à e sauf pour la valeur de x , (I, f) est modèle de $A \Rightarrow B$, alors (I, g) est modèle de B . Puisque x n'est pas libre dans B , (I, e) est modèle de B .
- \Leftarrow Supposons que $\Gamma \models (\exists xA) \Rightarrow B$. Puisque la formule $A \Rightarrow (\exists xA)$ est valide (d'après le corollaire 4.3.38 page 94), nous avons $\Gamma \models A \Rightarrow B$.

□

Si nous observons d'un œil critique cette preuve, nous voyons que c'est une paraphrase, dans un autre formalisme, de la loi $\exists E$. C'est l'équivalence ci-dessus, qui explique que la tactique « raisonner en avant avec une hypothèse d'existence », est une tactique utilisable sans risque. En effet, en admettant la complétude du système, lorsque x n'est libre ni dans Γ , ni dans B , il y a une preuve de B dans l'environnement Γ, A si et seulement s'il y a une preuve de B dans l'environnement $\Gamma, \exists xA$.

Théorème 6.3.3 (Cohérence de la déduction) Si une formule est déduite d'un environnement de formules alors elle en est une conséquence.

Preuve : La preuve obéit au même plan que celle du même théorème dans le cas propositionnel (voir théorème 3.3.1 page 73) et nous en reprenons les notations en traitant uniquement le cas des nouvelles règles.

Soit Γ un ensemble de formules. Soit P une preuve de A dans cet environnement. Soient C_i la conclusion et H_i le contexte de la i -ème ligne de la preuve P . Rappelons que les lignes d'une preuve sont numérotées à partir de 1 et que H_0 est la liste vide.

Notons par Γ, H_i l'ensemble des formules de l'ensemble Γ et de la liste H_i .

Cas de base : Supposons que A est déduite de Γ par la preuve vide. Alors A est élément de Γ , donc $\Gamma \models A$. Puisque H_0 est la liste vide, nous pouvons conclure : $\Gamma, H_0 \models A$.

Induction : Supposons que pour toute ligne $i < k$, $\Gamma, H_i \models C_i$. Montrons que $\Gamma, H_k \models C_k$.

Nous examinons uniquement le cas des nouvelles règles et pour simplifier nous ne faisons pas de distinctions entre deux formules égales aux abréviations près de la négation et de l'équivalence.

- Supposons que $C_k = \forall xA$ et que cette ligne ait été déduite, par la règle $\forall I$, de la formule A avec $A = C_i$ et $0 \leq i < k$ ou $A \in \Gamma$. Si $A = C_i$ et $0 < i < k$, par hypothèse de récurrence nous avons, $\Gamma, H_i \models A$. Si $A \in \Gamma$ alors $\Gamma \models A$. Puisque H_0 est la liste vide, il existe i où $0 \leq i < k$ tel que $\Gamma, H_i \models A$. Vu les conditions d'application de la règle, x n'est pas libre dans Γ, H_i . Donc, d'après la propriété 6.3.1 page précédente, nous avons aussi $\Gamma, H_i \models \forall xA$. Puisque la ligne i est utilisable sur la ligne $k - 1$ et que H_0 est la liste vide, H_i est préfixe de H_{k-1} . Puisque le contexte n'est pas modifié par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models C_k$.
- Supposons que $C_k = A < x := t >$ et que cette ligne ait été déduite, par la règle $\forall E$, de la formule $\forall xA$ avec $\forall xA = C_i$ et $0 < i < k$ ou $\forall xA \in \Gamma$. Par hypothèse de récurrence ou parce que H_0 est la liste vide, il existe i où $0 \leq i < k$ tel que $\Gamma, H_i \models \forall xA$. D'après les conditions d'application de la règle, le terme t est libre pour la variable x dans la formule A . Donc, d'après le corollaire 4.3.38 page 94, la formule $\forall xA \Rightarrow A < x := t >$ est valide et par suite $\Gamma, H_i \models A < x := t >$. Puisque la ligne i est utilisable sur la ligne $k - 1$, et que H_0 est la liste vide, H_i est préfixe de H_{k-1} . Puisque le contexte n'est pas modifié par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models C_k$.
- Supposons que $C_k = \exists xA$ et que cette ligne ait été déduite, par la règle $\exists I$, de la formule $A < x := t >$ avec $A < x := t > = C_i$ et $0 < i < k$ ou $A < x := t > \in \Gamma$. Par hypothèse de récurrence ou parce que H_0 est la liste vide, il existe i où $0 \leq i < k$ tel que $\Gamma, H_i \models A < x := t >$. D'après les conditions d'applications de la règle, le terme t est libre pour la variable x dans la formule A . Donc, d'après le corollaire 4.3.38 page 94, la formule

$A \langle x := t \rangle \Rightarrow \exists x A$ est valide et par suite $\Gamma, H_i \models \exists x A$. Puisque la ligne i est utilisable sur la ligne $k - 1$, et que H_0 est la liste vide, H_i est pr  fixe de H_{k-1} . Puisque le contexte n'est pas modifi   par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models C_k$.

- Supposons que $C_k = B$ et que cette formule ait   t   d  duite, par la r  gle $\exists E$, de la formule $\exists x A$ avec $\exists x A = C_i$ et $0 < i < k$ ou $\exists x A \in \Gamma$ et de la formule $A \Rightarrow B$ avec $A \Rightarrow B = C_j$ et $0 < j < k$ ou $A \Rightarrow B \in \Gamma$. Par hypoth  se de r  currence ou parce H_0 est la liste vide, il existe i et j tels que $0 \leq i < k$, $0 \leq j < k$, $\Gamma, H_i \models \exists x A$ et $\Gamma, H_j \models A \Rightarrow B$. Vu les conditions d'application de la r  gle, x n'est libre ni dans Γ, H_j , ni dans B . Donc, d'apr  s la propri  t   6.3.2 page ci-contre, nous avons aussi $\Gamma, H_j \models (\exists x A) \Rightarrow B$. Puisque les lignes i et j sont utilisables sur la ligne $k - 1$, et que H_0 est la liste vide, H_i et H_j sont pr  fixes de H_{k-1} . Puisque le contexte n'est pas modifi   par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models \exists x A$ et $\Gamma, H_k \models (\exists x A) \Rightarrow B$. Par suite $\Gamma, H_k \models C_k$.
- Supposons que $C_k = A'$ et que cette formule ait   t   d  duite, par la r  gle de copie de la formule A avec $A = C_i$ et $0 < i < k$ ou $A \in \Gamma$. Par hypoth  se de r  currence ou parce H_0 est la liste vide, il existe i tel que $0 \leq i < k$, $\Gamma, H_i \models A$. Puisque, d'apr  s le th  or  me 4.4.6 page 100, les formules A et A' sont   quivalentes, nous avons $\Gamma, H_i \models A'$. Puisque la ligne i est utilisable sur la ligne $k - 1$, et que H_0 est la liste vide, H_i est pr  fixe de H_{k-1} . Puisque le contexte n'est pas modifi   par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models C_k$.
- Supposons que $C_k = (t = t)$. Puisque cette formule est valide (dans le sens attribu      l'  galit  ), $\Gamma, H_k \models C_k$.
- Supposons que $C_k = A \langle x := t \rangle$ et que cette ligne ait   t   d  duite, par la r  gle de congruence, de la formule $s = t$ avec $(s = t) = C_i$ et $0 < i < k$ ou $(s = t) \in \Gamma$ et de la formule $A \langle x := s \rangle$ avec $A \langle x := s \rangle = C_j$ et $0 < j < k$ ou $A \langle x := s \rangle \in \Gamma$. Par hypoth  se de r  currence ou parce H_0 est la liste vide, il existe i et j tels que $0 \leq i < k$, $0 \leq j < k$, $\Gamma, H_i \models (s = t)$ et $\Gamma, H_j \models A \langle x := s \rangle$. Puisque les lignes i et j sont utilisables sur la ligne $k - 1$, et que H_0 est la liste vide, H_i et H_j sont pr  fixes de H_{k-1} . Puisque le contexte n'est pas modifi   par la ligne k , nous avons $H_{k-1} = H_k$, donc $\Gamma, H_k \models (s = t)$ et $\Gamma, H_k \models A \langle x := s \rangle$. D'apr  s le th  or  me 4.3.36 page 94 et les conditions d'application de la r  gle, nous avons : $s = t, A \langle x := s \rangle \models A \langle x := t \rangle$. Par suite $\Gamma, H_k \models C_k$.

Puisque la dern  re ligne d'une preuve a un contexte vide, nous obtenons que sa conclusion est cons  quence de Γ . \square

6.4 Exercices

Exercice 96 (Dédutions naturelles) Prouver en déduction naturelle du premier ordre les formules suivantes :

1. le fameux syllogisme, « Tout homme est mortel, Socrate est un homme, donc est mortel », que nous formalisons par $\forall x(H(x) \Rightarrow M(x)) \wedge H(\text{socrate}) \Rightarrow M(\text{socrate})$.
2. $\forall xP(x) \Rightarrow \exists yP(y)$.
3. $\forall xP(x) \Rightarrow \exists xP(x)$.
4. $\forall x(P(x) \wedge Q(x)) \Rightarrow \forall xP(x) \wedge \forall xQ(x)$. Notez que l'exemple 6.1.2 page 137 donne la preuve de la réciproque.
5. $\exists x(P(x) \vee Q(x)) \Rightarrow \exists xP(x) \vee \exists xQ(x)$.
6. $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists xP(x) \Rightarrow \exists xQ(x)$.
7. $\exists xP(x) \vee \exists xQ(x) \Rightarrow \exists x(P(x) \vee Q(x))$. (**)
8. $\forall x(P(x) \Rightarrow Q(x)) \wedge \forall x(Q(x) \Rightarrow R(x)) \Rightarrow \forall x(P(x) \Rightarrow R(x))$.
9. $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x \neg Q(x) \Rightarrow \exists x \neg P(x)$.

Dans cet exercice, les formules $P(x)$ et $Q(x)$ peuvent être remplacées par des formules quelconques. □

Exercice 97 (Dédutions naturelles) Prouver les formules suivantes :

1. $\forall x \forall y P(x, y) \Rightarrow \forall y \forall x P(x, y)$.
2. $\exists x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y)$.
3. $\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$.
4. $\forall x(Q(x) \Rightarrow \forall y(R(y) \Rightarrow P(x, y))) \Rightarrow \forall y(R(y) \Rightarrow \forall x(Q(x) \Rightarrow P(x, y)))$. (*)

Dans cet exercice, la formule $P(x, y)$ peut être remplacée par une formule quelconque. Par contre $Q(x)$ peut être remplacée seulement par une formule n'ayant pas y comme variable libre, et $R(y)$ peut être remplacée par une formule n'ayant pas x comme variable libre : expliquez la raison de ces contraintes. □

Exercice 98 (Chercher la faute) Considérons la formule suivante :

$$\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x)).$$

Parmi les trois preuves suivantes une seule est correcte par déduction naturelle. Identifier la preuve correcte et justifier pourquoi les deux autres ne le sont pas.

1.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E1$ 1
1	3	$\forall x Q(x)$	$\wedge E2$ 1
1, 4	4	Supposons $P(x)$	
1, 4	5	$Q(x)$	$\forall E$ 3, x
1, 4	6	$P(x) \wedge Q(x)$	$\wedge I$ 4, 5
1, 4	7	$\exists x (P(x) \wedge Q(x))$	$\exists I$ 6, x
1	8	Donc $P(x) \Rightarrow \exists x (P(x) \wedge Q(x))$	$\Rightarrow I$ 4, 7
1	9	$\exists x (P(x) \wedge Q(x))$	$\exists E$ 2, 8
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x))$	$\Rightarrow I$ 1, 9

2.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E 1$
1	3	$\forall x Q(x)$	$\wedge E 2$
1,4	4	Supposons $P(x)$	
1,4	5	$Q(x)$	$\forall E 3, x$
1,4	6	$P(x) \wedge Q(x)$	$\wedge I 4,5$
1	7	Donc $P(x) \Rightarrow P(x) \wedge Q(x)$	$\Rightarrow I 4,6$
1	8	$P(x) \wedge Q(x)$	$\exists E 2,7$
1	9	$\exists x(P(x) \wedge Q(x))$	$\exists I 8, x$
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 1,9$

3.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E 1$
1	3	$\forall x Q(x)$	$\wedge E 2$
1,4	4	Supposons $P(x)$	
1	5	Donc $P(x) \Rightarrow P(x)$	$\Rightarrow I 4,4$
1	6	$P(x)$	$\exists E 2,5$
1	7	$Q(x)$	$\forall E 3, x$
1	8	$P(x) \wedge Q(x)$	$\wedge I 6,7$
1	9	$\exists x(P(x) \wedge Q(x))$	$\exists I 8, x$
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 1,9$

□

Exercice 99 (Copie) Prouver par déduction naturelle que $\forall x \forall y P(x,y) \Rightarrow \forall x \forall y P(y,x)$ en utilisant la règle de copie un minimum de fois. □

Exercice 100 (Dédution naturelle) Prouver les formules suivantes grâce à la déduction naturelle (notez que Q est une variable propositionnelle) :

1. $\forall x(Q \wedge P(x)) \Rightarrow Q \wedge \forall x P(x)$.
2. $Q \wedge \forall x P(x) \Rightarrow \forall x(Q \wedge P(x))$.
3. $\forall x(Q \vee P(x)) \Rightarrow Q \vee \forall x P(x)$. (**)
4. $Q \vee \forall x P(x) \Rightarrow \forall x(Q \vee P(x))$.
5. $\exists x(Q \wedge P(x)) \Rightarrow Q \wedge \exists x P(x)$.
6. $Q \wedge \exists x P(x) \Rightarrow \exists x(Q \wedge P(x))$.
7. $\exists x(Q \vee P(x)) \Rightarrow Q \vee \exists x P(x)$.
8. $Q \vee \exists x P(x) \Rightarrow \exists x(Q \vee P(x))$. (*)

Dans cet exercice, la formule $P(x)$ peut être remplacée par une formule quelconque. Par contre, Q peut être remplacée seulement par une formule n'ayant pas x comme variable libre. □

Exercice 101 (Preuve) Prouver la formule $\neg \exists x P(x) \Rightarrow \forall x \neg P(x)$. Vérifier que $P(x)$ peut être remplacée par une formule quelconque. Remarque : cette formule est la réciproque de la formule prouvée à l'exemple 6.1.6 page 138. □

Exercice 102 (Égalité) Prouver les formules suivantes :

1. $R(a,c) \wedge (a=b) \Rightarrow R(b,c)$.
2. $x=y \Rightarrow f(x,z) = f(y,z)$.
3. $\forall x \exists y (x=y)$.

4. $\exists x \forall y x = y \Rightarrow \forall x \forall y x = y$. (*)

□

Exercice 103 (Égalité,)** Prouver que la deuxième définition de « il existe un et un seul x » (voir sous-section 6.2.3 page 140) implique la première, autrement dit prouver la formule : $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y) \Rightarrow \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$.

□

Exercice 104 (Induction et déduction naturelle,*)** Nous pouvons définir l'addition grâce aux deux formules :

(a) $\forall n (n + 0 = n)$.

(b) $\forall n \forall p (n + s(p) = s(n + p))$.

Ces deux formules permettent de faire des additions : nous pouvons grâce à elles prouver que $s(0) + s(0) = s(s(0))$. Mais elles ne permettent pas de prouver les propriétés générales de l'addition, qui nécessitent d'avoir le principe de récurrence.

1. Montrer que des hypothèses (a) et (b), nous ne pouvons pas déduire $\forall n (0 + n = n)$.
2. Nous donnons un nom à la propriété ci-dessus et nous posons le principe de récurrence sur cette propriété :
 (c) $\forall n (P(n) \Leftrightarrow 0 + n = n)$.
 (d) $P(0) \wedge \forall n (P(n) \Rightarrow P(s(n))) \Rightarrow \forall n P(n)$.
 Prouver par la déduction naturelle que $(a) \wedge (b) \wedge (c) \wedge (d) \Rightarrow \forall n P(n)$.

□

Exercice 105 (Examen 2009) Les preuves demandées ci-dessous devront être justifiées.

1. Donner une preuve en déduction naturelle avec les justifications de la validité de la formule :
 $(\exists x p(x) \Rightarrow \forall x q(x)) \Rightarrow \forall x (p(x) \Rightarrow q(x))$.
2. Donner une preuve en déduction naturelle avec les justifications de la validité de la formule :
 $\exists x p(x) \wedge \forall x (p(x) \Rightarrow p(f(x))) \Rightarrow \exists x p(f(f(x)))$.
3. Notons $f^n(x)$ le terme obtenu en appliquant n fois f à x .
 Par exemple $f^0(x) = x, f^1(x) = f(x), f^2(x) = f(f(x))$.
 Soient Γ, Δ deux ensembles de formules et A, B deux formules. On rappelle que $\Gamma \vdash A$ est vrai s'il y a une preuve de A dans l'environnement Γ .
 On donne ci-dessous des propriétés triviales de la relation \vdash .
 Monotonie : si $\Gamma \vdash A$ et $\Gamma \subset \Delta$ alors $\Delta \vdash A$.
 Composition : si $\Gamma \vdash A$ et $\Gamma \vdash A \Rightarrow B$ alors $\Gamma \vdash B$.
 (a) Donnez une preuve en déduction naturelle avec les justifications de ce que la propriété suivante est vraie :
 $\forall x (p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x)) \Rightarrow \exists x p(f^{n+1}(x))$
 (b) Déduire de la propriété ci-dessus, de la monotonie et de la composition que pour tout entier naturel n :
 $\exists x p(x), \forall x (p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x))$.

□

Exercice 106 (Examen 2012) Prouver les formules suivantes par déduction naturelle au premier ordre.

1. $\neg \forall x P(x) \vee \neg \exists y Q(y) \Rightarrow \neg (\forall x P(x) \wedge \exists y Q(y))$.
2. $\forall x \forall y (P(y) \Rightarrow R(x)) \Rightarrow \exists y P(y) \Rightarrow \forall x R(x)$.
3. $\neg \forall x \neg P(x) \Rightarrow \exists x P(x)$.

□

Exercice 107 (Examen 2013) Prouver les formules suivantes par déduction naturelle au premier ordre.

1. $\exists x (Q(x) \Rightarrow P(x)) \wedge \forall x Q(x) \Rightarrow \exists x P(x)$.
2. $\forall x \forall y (R(x, y) \Rightarrow \neg R(y, x)) \Rightarrow \forall x \neg R(x, x)$.

□

Exercice 108 (Quelques questions posées en examen) Démontrer les formules suivantes par déduction naturelle.

1. $\exists x (P(x) \vee Q(x)) \wedge \forall x \neg Q(x) \Rightarrow \exists x P(x)$.
2. $\forall x (P(x) \Rightarrow Q(x)) \wedge \exists x (P(x) \wedge R(x)) \Rightarrow \exists x (Q(x) \wedge R(x))$.
3. $\exists x \neg (P(x) \vee \neg P(x)) \Rightarrow \forall x P(x)$.

□

Troisième partie

Annexes

Corrigés

Exercice 1, page 37

1. x est une formule.
2. $(x$ n'est pas une formule.
3. (x) n'est pas une formule, mais est une formule à priorité.
4. $(x \Rightarrow (y \wedge z))$ n'est pas une formule.
5. $(x \Rightarrow (y \wedge z))$ est une formule.
6. $\neg(x \vee y)$ est une formule.
7. $(\neg(x \vee y))$ n'est pas une formule, mais est une formule à priorité.
8. $\neg(x \Rightarrow y) \vee \neg(y \wedge z)$ n'est pas une formule, mais est une formule à priorité.
9. $\neg(x)$ n'est pas une formule, mais est une formule à priorité.
10. $(\neg((x \vee y) \wedge z) \Leftrightarrow (u \Rightarrow v)))$ n'est pas une formule, car la dernière parenthèse est de trop.

Exercice 2, page 37

Nous prouvons le résultat par récurrence sur la taille des formules.

Cas de base : Soit A une formule telle que $|A| = 0$. La formule A est soit une variable soit une constante par construction d'une formule. D'après la définition 1.1.14 page 12, A est une formule à priorité.

Induction : Supposons que toute formule A de taille inférieure ou égale à $n \geq 0$ soit une formule à priorité.

Pas de récurrence : Montrons que la propriété est encore vraie pour $n + 1$. Considérons une formule A de taille $n + 1$.

Comme $n \geq 0$ nous avons $n + 1 > 1$, donc A ne peut pas être une constante ni une variable. Ainsi, deux cas sont possibles :

— Soit A est de la forme $\neg A'$ où A' est une formule de taille n .

— Soit A est de la forme $(B' \circ C')$ où B' et C' sont des formules de taille inférieure ou égale à n .

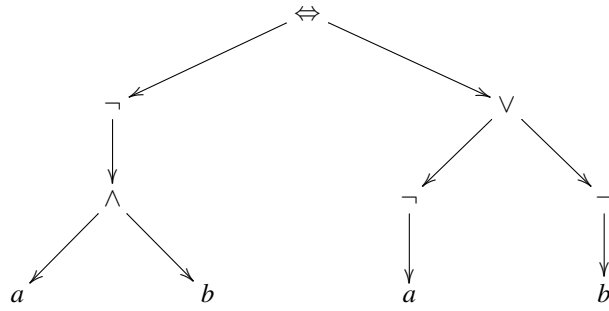
Par hypothèse de récurrence A' , B' et C' sont des formules à priorité. D'après les trois derniers points de la définition 1.1.14 page 12, $\neg A'$ et $(B' \circ C')$ sont des formules à priorité. D'où, A est une formule à priorité.

Exercice 3, page 37

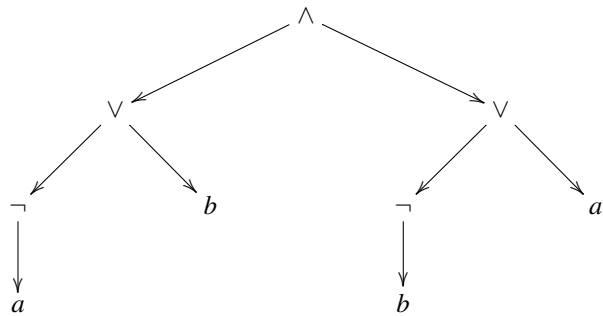
formule	formule à priorité	notations booléennes
1. $(\neg(a \wedge b) \Leftrightarrow (\neg a \vee \neg b))$	$\neg(a \wedge b) \Leftrightarrow \neg a \vee \neg b$	$\overline{ab} = \bar{a} + \bar{b}$
2. $((\neg a \vee b) \wedge (\neg b \vee a))$	$(\neg a \vee b) \wedge (\neg b \vee a)$	$(\bar{a} + b)(\bar{b} + a)$
3. $((((a \wedge b) \wedge c) \vee ((\neg a \wedge \neg b) \wedge \neg c))$	$a \wedge b \wedge c \vee \neg a \wedge \neg b \wedge \neg c$	$abc + \bar{a}\bar{b}\bar{c}$
4. $(p \Rightarrow (q \Rightarrow r))$	$p \Rightarrow q \Rightarrow r$	$p \Rightarrow q \Rightarrow r$
5. $((p \Rightarrow q) \Rightarrow r)$	$(p \Rightarrow q) \Rightarrow r$	$(p \Rightarrow q) \Rightarrow r$

Ci-après, nous donnons les structures sous forme d'arbre des formules :

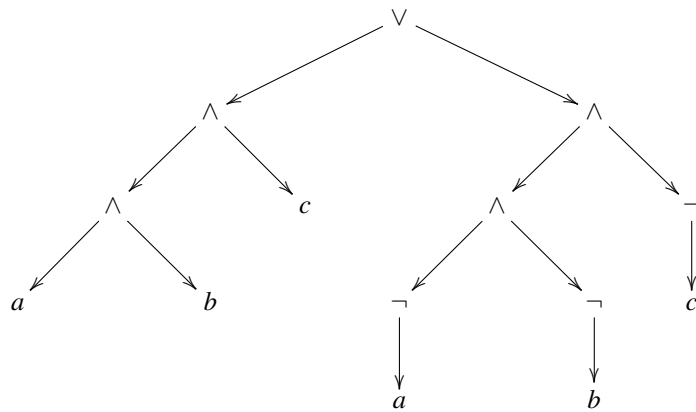
1. $(\neg(a \wedge b) \Leftrightarrow (\neg a \vee \neg b)).$



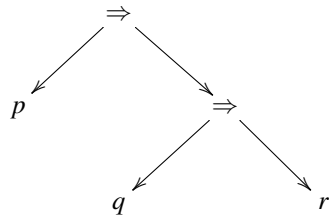
2. $((\neg a \vee b) \wedge (\neg b \vee a)).$



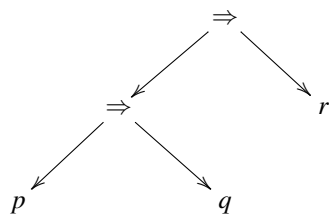
3. $((a \wedge b) \wedge c) \vee ((\neg a \wedge \neg b) \wedge \neg c).$



4. $(p \Rightarrow (q \Rightarrow r)).$



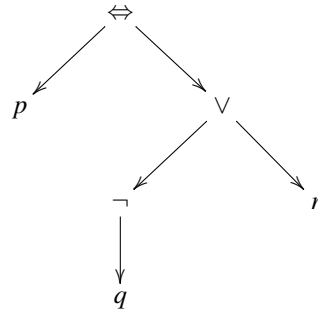
5. $((p \Rightarrow q) \Rightarrow r).$



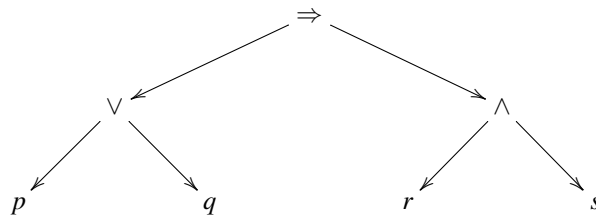
Exercice 4, page 37

Avant de dessiner les représentations sous forme d'arbre, nous donnons les formules sous la forme stricte.

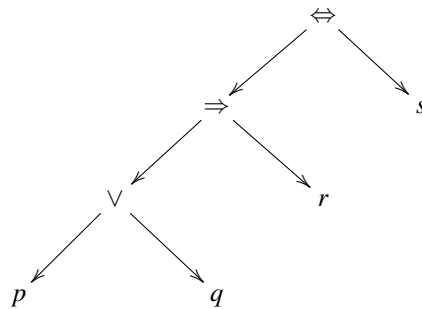
1. $p \Leftrightarrow \neg q \vee r \equiv (p \Leftrightarrow (\neg q \vee r)).$



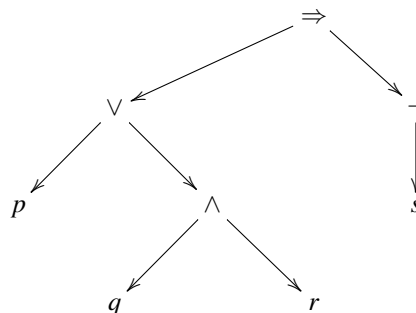
2. $p \vee q \Rightarrow r \wedge s \equiv ((p \vee q) \Rightarrow (r \wedge s)).$



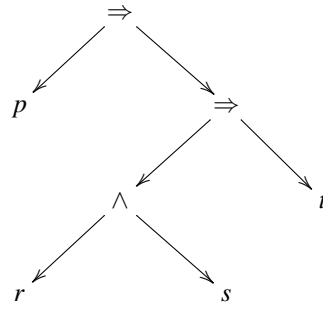
3. $p \vee q \Rightarrow r \Leftrightarrow s \equiv (((p \vee q) \Rightarrow r) \Leftrightarrow s).$



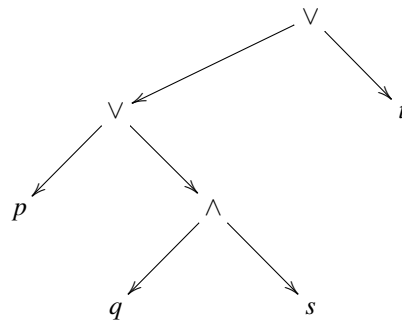
4. $p \vee q \wedge r \Rightarrow \neg s \equiv ((p \vee (q \wedge r)) \Rightarrow \neg s).$



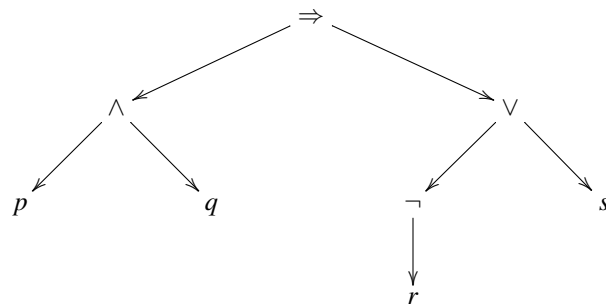
5. $p \Rightarrow r \wedge s \Rightarrow t \equiv (p \Rightarrow ((r \wedge s) \Rightarrow t)).$



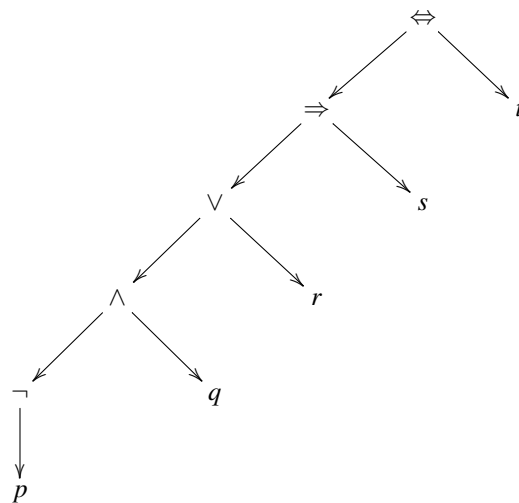
6. $p \vee q \wedge s \vee t \equiv ((p \vee (q \wedge s)) \vee t).$



7. $p \wedge q \Leftrightarrow \neg r \vee s \equiv ((p \wedge q) \Leftrightarrow (\neg r \vee s)).$



8. $\neg p \wedge q \vee r \Rightarrow s \Leftrightarrow t \equiv (((\neg p \wedge q) \vee r) \Rightarrow s) \Leftrightarrow t.$



Exercice 5, page 38

1. Nous montrons tout d'abord que cette définition est ambiguë, c'est-à-dire nous exhibons une formule qui peut avoir deux hauteurs différentes. Soit la formule $F = a \vee b \vee c \vee d$, nous avons deux calculs possibles :

$$\begin{aligned}
 h(F) &= 1 + \max(h(a), h(b \vee c \vee d)) \\
 &= 1 + \max(0, 1 + \max(h(b), h(c \vee d))) \\
 &= 1 + 1 + \max(0, 1 + \max(h(c), h(d))) \\
 &= 2 + 1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 h(F) &= 1 + \max(h(a \vee b), h(c \vee d)) \\
 &= 1 + \max(1 + \max(h(a), h(b)), 1 + \max(h(c), h(d))) \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

2. Pour résoudre les problèmes d'ambiguïté, il suffit de définir la hauteur sur les formules strictes. Nous définissons la hauteur h d'une formule stricte par :
- $h(\top) = 0$ et $h(\perp) = 0$.
 - Si A est une variable $h(A) = 0$.
 - $h(\neg A) = 1 + h(A)$.
 - $h((A \circ B)) = \max(h(A), h(B)) + 1$, si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$.

Exercice 6, page 38

Nous regroupons les réponses dans deux tables, une pour la formule à deux variables et une pour les formules à trois variables.

p	q	$p \Rightarrow (q \Rightarrow p)$
0	0	1
0	1	1
1	0	1
1	1	1

La formule $p \Rightarrow (q \Rightarrow p)$ est donc valide.

p	q	r	$p \Rightarrow (q \Rightarrow r)$	$(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$	$(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	1	1	1

La dernière formule est valide (remarquons que les deux formules valides sont par définition équivalentes). Les deux premières formules sont équivalentes.

Exercice 7, page 38

Ci-dessous les tables de vérité des formules : $a \Rightarrow b$, $b \Rightarrow c$, $c \Rightarrow a$, $a \Leftrightarrow b$, $b \Leftrightarrow c$, et $c \Leftrightarrow a$.

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$c \Rightarrow a$	$a \Leftrightarrow b$	$b \Leftrightarrow c$	$c \Leftrightarrow a$
0	0	0	1	1	1	1	1	1
0	0	1	1	1	0	1	0	0
0	1	0	1	0	1	0	0	1
0	1	1	1	1	0	0	1	0
1	0	0	0	1	1	0	1	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	1	1	0	0
1	1	1	1	1	1	1	1	1

Nous concluons grâce à la table de vérité que

$$(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge (c \Rightarrow a) \equiv (a \Leftrightarrow b) \wedge (b \Leftrightarrow c) \wedge (c \Leftrightarrow a)$$

car à chaque fois que les formules $(a \Rightarrow b)$, $(b \Rightarrow c)$ et $(c \Rightarrow a)$ sont vérifiées, nous avons à la fois les formules $(a \Leftrightarrow b)$, $(b \Leftrightarrow c)$, $(c \Leftrightarrow a)$ qui sont vérifiées, et réciproquement.

Exercice 8, page 38

Nous écrivons les tables de vérité de toutes les formules.

p	q	$p \wedge (p \vee q)$	$\neg p \wedge \neg q$	$\neg(p \wedge q)$	$\neg(p \vee q)$	$p \vee (p \wedge q)$	$\neg p \vee \neg q$
0	0	0	1	1	1	0	1
0	1	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	1	0	0	0	1	0

D'après la table de vérité, nous en déduisons que :

- Les formules p , $p \wedge (p \vee q)$, $p \vee (p \wedge q)$ sont équivalentes entre elles.
- Les formules $\neg p \wedge \neg q$, $\neg(p \vee q)$ sont équivalentes entre elles.
- Les formules $\neg(p \wedge q)$, $\neg p \vee \neg q$ sont équivalentes entre elles.

Exercice 9, page 38

Nous écrivons les tables de vérité de toutes les formules.

p	q	$(p \Rightarrow q) \wedge (q \Rightarrow p)$	$p \Rightarrow q$	$p \Leftrightarrow q$	$\neg q \Rightarrow \neg p$	$(p \wedge q) \vee (\neg p \wedge \neg q)$
0	0	1	1	1	1	1
0	1	0	1	0	1	0
1	0	0	0	0	0	0
1	1	1	1	1	1	1

D'après la table de vérité, nous en déduisons que :

- Les formules $(p \Rightarrow q) \wedge (q \Rightarrow p)$, $p \Leftrightarrow q$, $(p \wedge q) \vee (\neg p \wedge \neg q)$ sont équivalentes entre elles.
- Les formules $p \Rightarrow q$, $\neg q \Rightarrow \neg p$ sont équivalentes entre elles.

Exercice 10, page 38

Nous écrivons les tables de vérité de toutes les formules.

p	q	r	$p \Rightarrow q \vee r$	$q \wedge \neg r \Rightarrow p$	$p \wedge \neg r \Rightarrow q$	$\neg q \wedge \neg r \Rightarrow \neg p$	$q \vee \neg p \vee r$
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

D'après la table de vérité, nous en déduisons que les formules $p \Rightarrow q \vee r$, $p \wedge \neg r \Rightarrow q$, $\neg q \wedge \neg r \Rightarrow \neg p$, $q \vee \neg p \vee r$ sont équivalentes entre elles.

Exercice 11, page 38

Nous montrons qu'une assignation est un modèle d'un ensemble de formules, si et seulement si elle est un modèle de la conjonction des formules de l'ensemble avec ces deux étapes :

- \Rightarrow Soit v une assignation d'un modèle d'un ensemble de formules A_1, \dots, A_n . Par définition, v est un modèle de chaque A_i , donc $[A_i]_v = 1$. Nous en déduisons immédiatement que $[A_1 \wedge \dots \wedge A_n]_v = \min\{[A_1]_v, \dots, [A_n]_v\} = \min\{1, \dots, 1\} = 1$ ce qui signifie que v est un modèle de la conjonction des formules de l'ensemble A_1, \dots, A_n .
- \Leftarrow Soit v est un modèle de la conjonction des formules de l'ensemble A_1, \dots, A_n , c'est à dire que $[A_1 \wedge \dots \wedge A_n]_v = 1$. Or, $[A_1 \wedge \dots \wedge A_n]_v = \min\{[A_1]_v, \dots, [A_n]_v\}$. Donc, $\min\{[A_1]_v, \dots, [A_n]_v\} = 1$. Or, $[A_i]_v$ est soit 0, soit 1. Donc, nécessairement $\min\{[A_1]_v, \dots, [A_n]_v\} = \min\{1, \dots, 1\}$. Ainsi $[A_i]_v = 1$ pour tout i entre 1 et n , c'est-à-dire v est un modèle de l'ensemble des formules A_1, \dots, A_n .

Exercice 12, page 39

Nous montrons que les lois de simplification sont la conséquence des équivalences remarquables.

- $x \vee (x \wedge y) \equiv (x \wedge 1) \vee (x \wedge y) \equiv x \wedge (1 \vee y) \equiv x \wedge 1 \equiv x$
- $x \wedge (x \vee y) \equiv (x \wedge x) \vee (x \wedge y) \equiv x \vee (x \wedge y) \equiv x$
- $x \vee (\neg x \wedge y) \equiv (x \vee \neg x) \wedge (x \vee y) \equiv 1 \wedge (x \vee y) \equiv x \vee y$
- $x \wedge (\neg x \vee y) \equiv (x \wedge \neg x) \vee (x \wedge y) \equiv 0 \vee (x \wedge y) \equiv x \wedge y$

Exercice 13, page 39

Nous simplifions en utilisant l'identité $x \vee \neg x \wedge y = x \vee y$

$$\begin{aligned}
 (a \vee b) \wedge (\neg b \vee c) \Rightarrow a \vee c &= \neg((a \vee b) \wedge (\neg b \vee c)) \vee a \vee c \\
 &= \neg(a \vee b) \vee \neg(\neg b \vee c) \vee a \vee c \\
 &= (\neg a \wedge \neg b) \vee (\neg \neg b \wedge \neg c) \vee a \vee c \\
 &= a \vee (\neg a \wedge \neg b) \vee c \vee (\neg c \wedge b) \\
 &= a \vee \neg b \vee c \vee b \\
 &= a \vee c \vee \neg b \vee b \\
 &= a \vee c \vee 1 \\
 &= 1
 \end{aligned}$$

Exercice 14, page 39

Soit A la formule suivante :

$$(((a \Rightarrow \neg b) \Leftrightarrow \neg c) \wedge (c \vee d)) \wedge (a \Leftrightarrow d).$$

1. Une tautologie est une formule qui a la valeur 1 pour toute assignation, donc A n'est pas une tautologie car nous avons le contre-modèle suivant : $c = d = 0$.
2. Une contradiction est une formule qui a la valeur 0 pour toute assignation, donc A n'est pas une contradiction car nous avons un modèle : $a = b = c = d = 1$.
3. Après simplification, nous obtenons la forme normale conjonctive suivante :

$$(\neg a \vee \neg b \vee c) \wedge (a \vee \neg c) \wedge (b \vee \neg c) \wedge (c \vee d) \wedge (a \vee \neg d) \wedge (\neg a \vee d).$$

4. Après simplification, nous obtenons la forme normale disjonctive suivante :

$$a \wedge b \wedge c \wedge d \vee a \wedge \neg b \wedge \neg c \wedge d.$$

Exercice 15, page 39

Nous effectuons une preuve par récurrence sur la taille des formules. Considérons une formule F à une seule variable p sans négation.

Cas de base : $|F| = 0$. La récurrence est trivialement vérifiée dans ce cas.

Induction : $|F| = n + 1$ et supposons que toutes les formules à une seule variable construites sans négation de taille inférieure ou égale à n sont équivalentes à une formule de taille 0. Nous avons deux cas d'aux, où $|A| \leq n$ et $|B| \leq n$, donc par hypothèse de récurrence A et B sont donc équivalentes à une formule de taille 0, c'est-à-dire 0, 1, ou p :

- $F = (A \vee B)$. Nous avons donc 9 cas à examiner :
 - $A = 0$
 - $B = 0$, donc $F = 0$.
 - $B = 1$, donc $F = 1$.
 - $B = p$, donc $F = p$.
 - $A = 1$
 - $B = 0$, donc $F = 1$.
 - $B = 1$, donc $F = 1$.
 - $B = p$, donc $F = 1$.
 - $A = p$
 - $B = 0$, donc $F = p$.
 - $B = 1$, donc $F = 1$.
 - $B = p$, donc $F = p$.
- $F = (A \wedge B)$. Nous avons donc 9 cas à examiner :
 - $A = 0$
 - $B = 0$, donc $F = 0$.
 - $B = 1$, donc $F = 0$.
 - $B = p$, donc $F = 0$.
 - $A = 1$
 - $B = 0$, donc $F = 0$.
 - $B = 1$, donc $F = 1$.
 - $B = p$, donc $F = p$.
 - $A = p$
 - $B = 0$, donc $F = 0$.
 - $B = 1$, donc $F = p$.
 - $B = p$, donc $F = p$.

Exercice 16, page 39

Nous rappelons qu'un opérateur $*$ est :

- Commutatif si pour tout x, y , $x * y = y * x$.
- Associatif si pour tout x, y, z , $(x * y) * z = x * (y * z)$.
- Idempotent si pour tout x , $x * x = x$.

— Transitif si pour tout x, y, z , $x * y$ et $y * z$ implique $x * z$.

x	$x \Rightarrow x$	$x \Leftrightarrow x$
0	1	1
1	1	1

D'après la table de vérité, \Rightarrow et \Leftrightarrow ne sont pas idempotents.

x	y	$x \Rightarrow y$	$y \Rightarrow x$	$x \Leftrightarrow y$	$y \Leftrightarrow x$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	1	1	1	1

D'après la table de vérité, \Rightarrow n'est pas commutatif mais \Leftrightarrow l'est.

x	y	z	$x \Rightarrow (y \Rightarrow z)$	$(x \Rightarrow y) \Rightarrow z$	$x \Leftrightarrow (y \Leftrightarrow z)$	$(x \Leftrightarrow y) \Leftrightarrow z$
0	0	0	1	0	0	0
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	1	1	0	0
1	0	0	1	1	1	1
1	0	1	1	1	0	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1

D'après la table de vérité, \Rightarrow n'est pas associatif mais \Leftrightarrow l'est.

x	y	z	$x \Rightarrow y$	$y \Rightarrow z$	$x \Rightarrow z$	$x \Leftrightarrow y$	$y \Leftrightarrow z$	$x \Leftrightarrow z$
0	0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	0	0
0	1	0	1	0	1	0	0	1
0	1	1	1	1	1	0	1	0
1	0	0	0	1	0	0	1	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	1	1	1	1

D'après la table de vérité, \Rightarrow et \Leftrightarrow sont transitifs. À l'aide de toutes ces tables de vérité, nous concluons :

opération	associative	commutative	idempotente	transitive
\Rightarrow	non	non	non	oui
\Leftrightarrow	oui	oui	non	oui

Exercice 17, page 39

Considérons la table de vérité suivante :

p	0	p	$\neg p$	1
0	0	0	1	1
1	0	1	0	1

La table de vérité de toute formule à une variable p , est une des colonnes du tableau ci-dessus, donc toute formule à une variable p est équivalente à l'une des quatre formules $0, p, \neg p, 1$.

Exercice 18, page 39

Les 16 formules possibles peuvent être : $0, p \wedge q, \neg(p \Rightarrow q), p, \neg(q \Rightarrow p), q, \neg(p \Leftrightarrow q), p \vee q, \neg(p \vee q), p \Leftrightarrow q, \neg q, q \Rightarrow p, \neg p, p \Rightarrow q, \neg(p \wedge q), 1$. Considérons la table de vérité suivante :

p	q	0	$p \wedge q$	$\overline{p \Rightarrow q}$	p	$\overline{q \Rightarrow p}$	q	$\overline{p \Leftrightarrow q}$	$p \vee q$	$\overline{p \vee q}$	$p \Leftrightarrow q$	\overline{q}	$q \Rightarrow p$	\overline{p}	$p \Rightarrow q$	$\overline{p \wedge q}$	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

La table de vérité de toute formule à deux variables p et q est une des colonnes du tableau ci-dessus, donc toute formule à deux variables p et q est équivalente à l'une des 16 formules ci-dessus.

Exercice 19, page 39

Le domaine d'une telle fonction comporte 2^k éléments, sa portée comporte 2 éléments. Donc il y a 2^{2^k} fonctions booléennes à k arguments.

Exercice 20, page 39

Au cours de l'enquête dont il est chargé, l'adjudant Tifrice effectue le raisonnement suivant :

- Si le meurtre a eu lieu le jour, le meurtrier est l'ami de la victime
- Or le meurtre a eu lieu la nuit

Donc le meurtrier n'est pas l'ami de la victime. Le raisonnement de l'adjudant Tifrice est-il correct ?

1. Formalisation : les *faits* sont remplacés par des variables. La lettre p dénote que le meurtre a eu lieu le jour et la lettre q dénote que le meurtrier est un ami de la victime.
2. Avec ces abréviations, Tifrice affirme que $\neg q$ est conséquence de $p \Rightarrow q, \neg p$, c'est-à-dire $(p \Rightarrow q) \wedge \neg p \Rightarrow \neg q$.
3. Ce raisonnement est incorrect car l'assignation $p = 0, q = 1$ est modèle des hypothèses et contre-modèle de la conclusion.

Exercice 21, page 39

Les faits, Pinocchio chante, Quasimodo chante, Roméo chante, sont dénotés respectivement par les variables p, q, r . Les hypothèses sont formalisées par les formules :

1. $\neg p \Rightarrow q$
2. $q \Rightarrow p \wedge r$
3. $r \Rightarrow \neg q \vee \neg p$

Et la conclusion par p .

En utilisant la propriété 1.2.27 page 17, nous montrons que de $(\neg p \Rightarrow q) \wedge (q \Rightarrow p \wedge r) \wedge (r \Rightarrow \neg q \vee \neg p) \wedge \neg p$, nous pouvons déduire une contradiction. En enlevant les implications nous avons :

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

En utilisant $(x \vee \neg y) \wedge y = x \wedge y$, entre le premier et le dernier terme nous obtenons :

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

En utilisant $(x \vee \neg y) \wedge y = x \wedge y$, entre le premier et le second terme nous obtenons :

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

Nous obtenons une contradiction entre p et $\neg p$. Ainsi le raisonnement est correct.

Exercice 22, page 40

Les faits, Pierre est rentré chez lui, Jean est allé au cinéma, Marie est à la bibliothèque sont dénotés respectivement par les variables p, j, m . Les hypothèses sont formalisées par les formules :

1. $p \Rightarrow j$.
2. $m \vee p$.
3. $j \Rightarrow m \vee p$.
4. $\neg m \wedge j$.

Et la conclusion par p .

En utilisant la propriété 1.2.27 page 17, nous prouvons que $(p \Rightarrow j) \wedge (m \vee p) \wedge (j \Rightarrow m \vee p) \wedge (\neg m \wedge j) \wedge \neg p$ est contradictoire. Car en enlevant les implications nous avons :

$$(\neg p \vee j) \wedge (m \vee p) \wedge (\neg j \vee m \vee p) \wedge \neg m \wedge j \wedge \neg p.$$

En utilisant $(x \vee \neg y) \wedge y = x \wedge y$, entre la seconde et la quatrième clause nous obtenons :

$$(\neg p \vee j) \wedge p \wedge (\neg j \vee m \vee p) \wedge \neg m \wedge j \wedge \neg p.$$

Nous obtenons une contradiction entre p et $\neg p$. Et donc le raisonnement est correct.

Exercice 23, page 40

Ci-dessous les étapes de la transformation de la formule $\neg(a \Leftrightarrow b) \vee (b \wedge c) \Rightarrow c$ en une forme normale disjonctive.

$$\begin{aligned} \neg(a \Leftrightarrow b) \vee (b \wedge c) \Rightarrow c &\equiv \neg((a \Rightarrow b) \wedge (b \Rightarrow a)) \vee (b \wedge c) \Rightarrow c \\ &\equiv \neg((\neg a \vee b) \wedge (\neg b \vee a)) \vee (b \wedge c) \Rightarrow c \\ &\equiv ((\neg a \vee b) \wedge (\neg b \vee a) \wedge (\neg b \vee \neg c)) \vee c \\ &\equiv (((\neg a \wedge \neg b) \vee (b \wedge a)) \wedge (\neg b \vee \neg c)) \vee c \\ &\equiv (((\neg a \wedge \neg b) \vee (b \wedge a)) \vee c) \wedge (\neg b \vee \neg c \vee c) \\ &\equiv \neg a \wedge \neg b \vee a \wedge b \vee c \end{aligned}$$

Par suite tout modèle de la formule vérifie l'une des 3 conditions suivantes :

- $a = 1$ et $b = 1$.
- $a = 0$ et $b = 0$.
- $c = 1$.

Ci-dessous les étapes de la transformation de la formule $(a \Rightarrow b) \wedge (b \Rightarrow \neg a) \wedge (\neg a \Rightarrow b) \wedge (b \Rightarrow a)$ en une somme de monômes.

$$\begin{aligned} (a \Rightarrow b) \wedge (b \Rightarrow \neg a) \wedge (\neg a \Rightarrow b) \wedge (b \Rightarrow a) &\equiv (\neg a \vee b) \wedge (\neg b \vee \neg a) \wedge (a \vee b) \wedge (\neg b \vee a) \\ &\equiv \neg a \wedge a \\ &\equiv 0 \end{aligned}$$

La transformation donne 0 donc la formule n'est pas satisfaisable. Nous remarquons que cette formule indique que les variables a et $\neg a$ sont équivalentes à b .

Exercice 24, page 40

$$\begin{aligned} p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r) &\equiv (p \vee (q \wedge r) \Rightarrow (p \vee q) \wedge (p \vee r)) \wedge ((p \vee q) \wedge (p \vee r) \Rightarrow p \vee (q \wedge r)) \\ &\equiv ((\neg p \wedge (\neg q \vee \neg r)) \vee ((p \vee q) \wedge (p \vee r))) \wedge (((\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)) \vee p \vee (q \wedge r)) \\ &\equiv (((\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)) \vee ((p \vee q) \wedge (p \vee r))) \wedge (\neg q \vee \neg r \vee p \vee (q \wedge r)) \\ &\equiv 1 \wedge (\neg q \vee \neg r \vee p \vee r) \\ &\equiv 1 \end{aligned}$$

La transformation donne 1 donc A est valide. Notez que la formule A exprime la distributivité de la somme \vee sur le produit \wedge , donc la preuve suivante par remplacement est beaucoup plus courte.

$$\begin{aligned} p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r) &\equiv p \vee (q \wedge r) \Leftrightarrow p \vee (q \wedge r) \\ &\equiv a \Leftrightarrow a \\ &\equiv 1 \end{aligned}$$

Exercice 25, page 40

1. Montrons que l'application de l'élimination des équivalences, des implications et de déplacement des négations sur une formule donne une formule en forme normale.

Tout d'abord, à chaque fois que nous appliquons les identités d'élimination d'équivalences ou d'implications sur une formule donnée, le nombre d'équivalences et d'implications décroît strictement. Donc, après un nombre fini d'applications de ces règles, nous obtenons une formule sans équivalence ni implication.

Considérons maintenant des formules construites uniquement avec les opérateurs \neg , \wedge et \vee . Soit m la mesure suivante :

- $m(\perp) = 1$
- $m(\top) = 1$
- pour toute variable x , $m(x) = 1$
- $m(\neg A) = 2m(A)$
- $m(A \circ B) = m(A) + m(B) + 1$ où \circ est la conjonction ou la disjonction

Soient A et B , deux formules. Nous avons :

- $m(\neg \neg A) > m(A)$
- $m(\neg(A \wedge B)) = 2(m(A) + m(B) + 1)$ et $m(\neg A \vee \neg B) = 2(m(A) + m(B)) + 1$. Donc $m(\neg(A \wedge B)) > m(\neg A \vee \neg B)$
- De même, $m(\neg(A \vee B)) > m(\neg A \wedge \neg B)$

Ainsi, appliquer une des règles de déplacement de négations à une formule permet d'obtenir une formule de mesure strictement plus petite.

D'où, nous ne pouvons appliquer ces règles qu'un nombre fini de fois. Par définition, lorsque plus aucune des règles n'est applicable à une formule, cette formule est en forme normale.

2. Montrons que quel que soit l'ordre d'application de l'élimination des équivalences, des implications et de déplacement des négations sur une formule, l'algorithme de transformation en forme normale termine.

La preuve précédente est indépendante de l'ordre d'application des règles d'élimination des équivalences, des implications et de déplacement des négations, donc elle est vérifiée quel que soit l'algorithme de transformation en forme normale utilisant ces trois transformations.

Exercice 26, page 40

Formalisons par les variables p, q, r, s les faits respectifs suivants, Jean a rencontré Pierre l'autre nuit, Pierre est le meurtrier, Jean est un menteur, le crime a eu lieu après minuit. La conjonction des hypothèses et de la négation de la conclusion s'exprime par le produit de clauses suivant :

$$(p + q + r)(q + \bar{p})(q + s)(\bar{s} + q + r)\bar{q}.$$

Effectuons le produit de gauche à droite en utilisant la distributivité et les identités $x\bar{x} = 0$ et $x + 0 = x$, $x + xy = x$:

$$\begin{aligned} &(p + q + r)(q + \bar{p})(q + s)(\bar{s} + q + r)\bar{q} \\ = &(pq + q + rq + r\bar{p})(q + s)(\bar{s} + q + r)\bar{q} \\ = &(q + r\bar{p})(q + s)(\bar{s} + q + r)\bar{q} \\ = &(q + r\bar{p}s)(\bar{s} + q + r)\bar{q} \\ = &(q + r\bar{p}s)\bar{q} \\ = &r\bar{p}s\bar{q} \end{aligned}$$

Par suite l'assignation $r = s = 1$, $p = q = 0$ est modèle de la conjonction des hypothèses et de la négation de la conclusion : le raisonnement n'est pas correct.

Exercice 27, page 40

Cet exercice a pour but de faire réfléchir à la formalisation. Dans cette île si un habitant x déclare P , nous avons $x \Leftrightarrow P$. De plus notons par a « Aha est un Pur » et par $\neg a$ « Aha est un Pire », la lettre b a un sens analogue.

- (a) Supposons que Aha soit un Pire, il dirait la vérité, donc serait un Pur, ce qui est une contradiction. Donc Aha est un Pur, et par suite Bébé est un Pire.
Plus formellement (et plus facilement) vu la déclaration d'Aha et les propriétés des Purs et des Pires, nous avons : $a \Leftrightarrow (\neg a \vee \neg b)$.
Cette formule est équivalente (par réduction à la forme normale) à $a \wedge \neg b$, d'où Aha est un Pur, et Bébé est un Pire.
- (b) Vu que Aha déclare « L'un de nous deux au plus est un Pire », nous avons : $a \Leftrightarrow (a \vee b)$. Cette formule est équivalente (par réduction à la forme normale disjonctive) à $a \vee \neg b$, donc nous ne pouvons rien conclure.
- (c) Supposons que Aha soit un Pire, puisqu'il a menti, Bébé est un Pur. Supposons que Aha soit un Pur, puisqu'il dit la vérité, Bébé est (aussi) un Pur. Puisque l'un de ces deux cas est vrai, Bébé est un Pur.
Plus formellement, vu la déclaration d'Aha, nous avons : $a \Leftrightarrow ((a \wedge b) \vee (\neg a \wedge \neg b))$. Cette formule est équivalente (par réduction à la forme normale disjonctive) à b , donc Bébé est un Pur, mais nous ne pouvons pas déterminer qui est Aha.

Exercice 28, page 41

- L'ensemble $\{-, +\}$ est complet. En effet, les connecteurs manquants $0, 1, \bullet$ sont définissables avec les deux premiers car :
 - $1 = x + \bar{x}$, $0 = \bar{1} = \overline{x + \bar{x}}$ où x est une variable quelconque
 - $x \bullet y = \bar{x} + \bar{y}$
- L'ensemble $\{0, \Rightarrow\}$ est complet. En effet, les connecteurs $1, -, +, \bullet$ sont définissables de la façon suivante :
 - $1 = 0 \Rightarrow 0$
 - $\bar{x} = x \Rightarrow 0$
 - $x + y = \bar{x} \Rightarrow y = (x \Rightarrow 0) \Rightarrow y$
 - $x \bullet y = \bar{x} + \bar{y} = \bar{x} \Rightarrow \bar{y} = x \Rightarrow (y \Rightarrow 0) = (x \Rightarrow (y \Rightarrow 0)) \Rightarrow 0$
- Montrons que l'ensemble $|$ est complet, où $|$ est l'opérateur défini par : $x | y$ est vrai si et seulement si ni l'un, ni l'autre ne sont vrais, autrement dit $x | y = 1$ si et seulement si $x = 0$ et $y = 0$.
Les connecteurs $\{-, +\}$ sont définissables à l'aide de $|$:
 - $\bar{x} = x | x$
 - $x + y = \bar{x} | y = (x | y) | (x | y)$
 Or, comme nous l'avons vu ci-dessus, l'ensemble $\{-, +\}$ est complet.
- Montrons que l'ensemble $\{0, 1, +, \bullet\}$ est incomplet. Soit $f(x_1, \dots, x_n) = A$ une fonction booléenne où A est une formule sur les variables x_1, \dots, x_n et les opérations $\{0, 1, +, \bullet\}$. Par récurrence sur la taille de A , nous montrons que f est monotone.
Supposons que pour toute formule de taille inférieure à n , la fonction équivalente est monotone.
Soit A est une formule de taille 0 : A est une variable x_i ou une constante. Dans ces deux cas, f est monotone.
Soit A est de taille n où $n > 0$ et $A = (B + C)$ ou $A = (B \bullet C)$. Examinons le premier cas, le deuxième étant analogue. Posons $g(x_1, \dots, x_n) = B$ et $h(x_1, \dots, x_n) = C$. Par récurrence g et h sont monotones, donc si $a_1 \leq b_1, \dots, a_n \leq b_n$, alors $g(a_1, \dots, a_n) \leq g(b_1, \dots, b_n)$ et $h(a_1, \dots, a_n) \leq h(b_1, \dots, b_n)$. Puisque $+$ est monotone, nous avons :

$$f(a_1, \dots, a_n) = g(a_1, \dots, a_n) + h(a_1, \dots, a_n) \leq f(b_1, \dots, b_n) = g(b_1, \dots, b_n) + h(b_1, \dots, b_n).$$

Donc f est monotone.

Soit $k(x) = \neg x$. Cette fonction n'est pas monotone, donc ne peut pas être écrite avec les seules opérations $\{0, 1, +, \bullet\}$. Par suite cet ensemble est incomplet.

Nous proposons une autre alternative pour cette preuve. En utilisant l'exercice 15 page 37, qui dit que toute formule à une seule variable construite sans négation et avec les connecteurs \vee et \wedge est équivalente à une formule de taille 0. Ensuite nous remarquons que $\neg p$ est de taille 1 et donc n'est équivalente à aucune formule de taille 0.

5. Montrons que l'ensemble $\{1, \Rightarrow\}$ est incomplet.

Appelons stable pour 1, une fonction booléenne qui prend la valeur 1 quand tous ses arguments ont la valeur 1 : par exemple $+$ est stable pour 1, et \neg n'est pas stable pour 1.

Comme dans l'exemple précédent, nous montrons que toute fonction booléenne définie avec les seules opérations $\{1, \Rightarrow\}$ est stable pour 1, car $1 \Rightarrow 1 \equiv 1$. Par suite cet ensemble est incomplet.

6. Montrons que l'ensemble $\{0, \Leftrightarrow\}$ est incomplet.

Considérons les 8 fonctions booléennes suivantes : $f_1(x, y) = 0$, $f_2(x, y) = 1$, $f_3(x, y) = x$, $f_4(x, y) = \bar{x}$, $f_5(x, y) = y$, $f_6(x, y) = \bar{y}$, $f_7(x, y) = (x \Leftrightarrow y)$, $f_8(x, y) = \neg(x \Leftrightarrow y)$.

Nous montrons que toute fonction booléenne à deux arguments définie avec les seules opérations $\{0, \Leftrightarrow\}$ est l'une de ces 8 fonctions.

Supposons que la fonction f définie par une formule sur $\{0, \Leftrightarrow\}$ soit l'une de ces 8 fonctions quand le nombre d'opérations \Leftrightarrow de la formule est inférieur à n . Soit f définie par une formule sur $\{0, \Leftrightarrow\}$ avec n opérations \Leftrightarrow .

- Si $n = 0$, alors $f(x, y) = x$ ou $f(x, y) = y$ ou $f(x, y) = 0$, donc f est l'une des 8 fonctions f_i .
- Si $n > 0$, alors $f(x, y) = (A \Leftrightarrow B)$. Posons $g(x, y) = A$ et $h(x, y) = B$. Par récurrence, g (respectivement h) est l'une des 8 fonctions f_i . Pour montrer que f est aussi l'une des 8 fonctions, il suffit de montrer que $(f_i(x, y) \Leftrightarrow f_j(x, y))$ est équivalente à l'une des expressions $0, 1, x, \bar{x}, y, \bar{y}, x \Leftrightarrow y, \neg(x \Leftrightarrow y)$. Nous présentons ce calcul en donnant dans la case i, j , l'expression équivalente à $(f_i(x, y) \Leftrightarrow f_j(x, y))$ pour $i \leq j$. Comme l'équivalence est commutative, l'expression de la case i, j où $i > j$ est la même que celle de la case j, i , aussi nous n'avons pas complété ces cases.

$f(x, y)$	1	2	3	4	5	6	7	8
1	1	0	\bar{x}	x	\bar{y}	y	$\neg(x \Leftrightarrow y)$	$x \Leftrightarrow y$
2		1	x	\bar{x}	y	\bar{y}	$x \Leftrightarrow y$	$\neg(x \Leftrightarrow y)$
3			1	0	$x \Leftrightarrow y$	$\neg(x \Leftrightarrow y)$	y	\bar{y}
4				1	$\neg(x \Leftrightarrow y)$	$x \Leftrightarrow y$	\bar{y}	y
5					1	0	x	\bar{x}
6						1	\bar{x}	x
7							1	0
8								1

Ce tableau montre que f est bien l'une des 8 fonctions f_i . Donc toute fonction à deux arguments définie par une formule sur $\{0, \Leftrightarrow\}$ est l'une de ces fonctions. Soit $g(x, y) = x.y$. Puisque g n'est aucune de ces 8 fonctions, g ne peut pas être définie par une formule sur $\{0, \Leftrightarrow\}$ et par suite l'ensemble $\{0, \Leftrightarrow\}$ est incomplet.

Exercice 29, page 41

Dans la correction, nous utilisons la propriété suivante : soient A une formule et v une assignation. Par définition, $[A]_v$ vaut 0 ou 1. Or, $\bar{0} = 1$ et $\bar{1} = 0$. Donc :

$$\overline{[A]_v} = 1 - [A]_v$$

1. Si deux formules sont équivalentes, il en est de même de leurs duales.

- (a) Pour toute formule A : $[A^*]_v = \overline{[A]_{\bar{v}}}$. Admettons cette propriété pour toute sous-formule stricte de la formule A et montrons la pour A .

— Supposons que A soit une variable, nous avons $[A^*]_v = [A]_v = v(A)$ ce qui implique :

$$\overline{[A]_{\bar{v}}} = \overline{v(A)} = \overline{v(A)} = v(A).$$

Donc la propriété est vérifiée.

— Supposons que $A = \top$, nous avons $[A^*]_v = [\perp]_v = 0$, nous obtenons

$$\overline{[A]_{\bar{v}}} = \overline{[\top]_{\bar{v}}} = 0.$$

Donc la propriété est vérifiée. Nous omettons le cas analogue où $A = \perp$.

— Supposons que $A = \neg B$.

$$[A^*]_v = [(\neg B)^*]_v = [(\neg B^*)]_v = 1 - [B^*]_v$$

Or, par hypothèse de récurrence :

$$[B^*]_v = \overline{[B]_{\bar{v}}}$$

Ainsi,

$$[A^*]_v = 1 - \overline{[B]_{\bar{v}}} = 1 - (1 - [B]_{\bar{v}}) = [B]_{\bar{v}}$$

Or,

$$\overline{[A]_{\bar{v}}} = 1 - [A]_{\bar{v}} = 1 - [\neg B]_{\bar{v}} = 1 - (1 - [B]_{\bar{v}}) = [B]_{\bar{v}}$$

Nous pouvons donc conclure que $[A^*]_v = \overline{[A]_{\bar{v}}}$.

— Supposons que $A = (B \wedge C)$.

$$\begin{aligned} [A^*]_v &= [(B \wedge C)^*]_v \\ &= [(B^* \vee C^*)]_v \\ &= \max([B^*]_v, [C^*]_v) \\ &= \max(\overline{[B]_{\bar{v}}}, \overline{[C]_{\bar{v}}}) \\ &= \max(1 - [B]_{\bar{v}}, 1 - [C]_{\bar{v}}) \end{aligned}$$

(L'avant-dernière ligne est obtenue en utilisant l'hypothèse de récurrence.)

$$\begin{aligned} \overline{[A]_{\bar{v}}} &= 1 - [A]_{\bar{v}} \\ &= 1 - [(B \wedge C)]_{\bar{v}} \\ &= 1 - [\neg(\neg B \vee \neg C)]_{\bar{v}} \\ &= 1 - (1 - [(\neg B \vee \neg C)]_{\bar{v}}) \\ &= [(\neg B \vee \neg C)]_{\bar{v}} \\ &= \max([\neg B]_{\bar{v}}, [\neg C]_{\bar{v}}) \\ &= \max(1 - [B]_{\bar{v}}, 1 - [C]_{\bar{v}}) \end{aligned}$$

(Notez l'utilisation de la loi de De Morgan à la ligne 3.)

Ainsi, nous pouvons conclure que $[A^*]_v = \overline{[A]_{\bar{v}}}$. Nous omettons le cas analogue de la disjonction.

- (b) Si 2 formules sont équivalentes, il en est de même de leurs duales. Soit v une assignation quelconque et supposons que les formules A, B soient équivalentes. D'après la propriété précédente :

$$[A^*]_v = \overline{[A]_{\bar{v}}}, [B^*]_v = \overline{[B]_{\bar{v}}}.$$

Puisque les formules A et B sont équivalentes, nous avons $[A]_{\bar{v}} = [B]_{\bar{v}}$, donc $[A^*]_v = [B^*]_v$. Par suite les duales de A et B sont équivalentes.

2. Si une formule est valide, sa duale est contradictoire. Soit A valide, elle est équivalente à \top . D'après le résultat la duale de A est équivalente à la duale de \top , c'est-à-dire à \perp . Donc cette duale est contradictoire.

Exercice 30, page 41

- Évaluons le nombre de classes de formules. Nous associons à la classe de la formule A , la fonction booléenne $f(x_1, \dots, x_n) = A$. Nous montrons que cette association est une bijection entre les classes de formules et les fonctions booléennes à n arguments. Donc, d'après l'exercice 19 page 37, il y a 2^{2^n} classes de formules.
 - Cette association est une fonction. En effet soient A et B deux formules équivalentes. Soient les deux fonctions $f(x_1, \dots, x_n) = A$ et $g(x_1, \dots, x_n) = B$. Pour toute valeur $a_1, \dots, a_n \in \mathbb{B}$, $f(a_1, \dots, a_n) = [A]_v$ et $g(a_1, \dots, a_n) = [B]_v$ où v est l'assignation telle que $v(x_1) = a_1, \dots, v(x_n) = a_n$. Puisque A et B sont équivalentes, nous avons : $f(a_1, \dots, a_n) = g(a_1, \dots, a_n)$. Donc $f = g$.
 - Cette fonction est surjective. En effet soit f une fonction booléenne à n arguments, d'après le théorème 1.6.3 page 32, il y a une somme de monômes A telle que $f(x_1, \dots, x_n) = A$.

- Cette fonction est injective. Supposons que les deux formules A et B vérifient $f(x_1, \dots, x_n) = A = B$. D'après cette propriété, A et B sont équivalentes, donc les classes de A et B sont identiques.
- 2. Nous avons les équivalences suivantes :
 - $\bar{x} = 1 \oplus x$,
 - $x + y = x \oplus xy \oplus y$,
 - $x \Rightarrow y = 1 \oplus x \oplus xy$,
 - $x \Leftrightarrow y = 1 \oplus x \oplus y$.
- 3. Toute formule booléenne est équivalente à une formule comportant les seuls connecteurs \neg et \vee d'après l'exercice 28 page 38. Il suffit de remplacer ces connecteurs par leur définition en terme de \oplus puis d'appliquer la distributivité de \wedge sur \oplus .
- 4. Si C et D sont deux produits de variables ayant même ensemble de variables alors $C \oplus D = 0$. En appliquant cette identité à une somme ou-disjonctive de produits de variables et en enlevant les 0 (par l'identité $0 + x = x + 0 = x$) nous obtenons la forme désirée.
- 5. Par l'idempotence, la commutativité et l'associativité de \wedge , tout produit de variables peut être remplacé par un produit équivalent sans répétition de variables, les variables étant ordonnées de gauche à droite dans un ordre croissant. De même par la commutativité et l'associativité de \oplus , toute somme de produits de variables peut être réordonnée pour que les produits soient placés de gauche à droite dans l'ordre décroissant. La correspondance qui, à une formule canonique associe l'ensemble des ensembles des variables de ses produits est évidemment bijective. Puisque l'ensemble des ensembles de variables a 2^{2^n} éléments, il y a donc 2^{2^n} formes canoniques donc une forme canonique par classe de formules équivalentes : par suite deux formules sont équivalentes si elles ont même forme canonique équivalente, en bref la forme canonique est unique.

Exercice 31, page 64

L'assignation $a = b = 0$ est modèle de $b + \bar{b}$ mais pas de $a + b$ donc pas de $(a + b) \cdot (\bar{a} + \bar{b})$. Donc le résolvant C de 2 clauses A et B est conséquence de $\{A, B\}$ mais n'est pas en général équivalent à cet ensemble.

Exercice 32, page 64

- Les clauses $p + q + \bar{r} + q + p + s + q + \bar{r}$ et $s + q + \bar{r} + p$ sont égales, car elles ont le même ensemble de littéraux $\{p, q, \bar{r}, s\}$.
- Les clauses $p + q + r + p$ et $q + r + q + r + q + r$ ne sont pas égales. La clause $q + r + q + r + q + r$ est incluse dans la clause $p + q + r + p$, donc $q + r + q + r + q + r$ a pour conséquence $p + q + r + p$. La réciproque est fautive, car pour l'assignation $p = 1, q = 0, r = 0$, la clause $p + q + r + p$ vaut 1 tandis que la clause $q + r + q + r + q + r$ vaut 0.
- Ces deux clauses ont deux résolvants $a + c + \bar{c}$ et $a + b + \bar{b}$ qui sont valides.

Exercice 33, page 64

- Nous transformons en clauses chaque formule de l'ensemble de clauses $\{a, a \Rightarrow b, \bar{b}\}$. Ce qui donne immédiatement l'ensemble de clauses $\{a, \bar{a} + b, \bar{b}\}$. La preuve par résolution est :

1	a	Hyp
2	$\bar{a} + b$	Hyp
3	b	Résolvant de 1, 2
4	\bar{b}	Hyp
5	\perp	Résolvant de 3, 4
- Soit L'ensemble de clauses : $\{a + b, \bar{a} + c, \bar{a} + \bar{d}, d + \bar{c}, \bar{b} + a\}$. Nous suggérons une tactique pour obtenir une preuve : construire en priorité les résolvants ayant le moins de littéraux possibles.

1	$a + b$	Hyp
2	$\bar{b} + a$	Hyp
3	a	Résolvant de 1, 2
4	$\bar{a} + c$	Hyp
5	c	Résolvant de 3, 4
6	$\bar{a} + \bar{d}$	Hyp
7	\bar{d}	Résolvant de 3, 6
8	$d + \bar{c}$	Hyp
9	d	Résolvant de 5, 8
10	\perp	Résolvant de 7, 9

— Soit l'ensemble de clauses : $\{a + b + c, \bar{a} + b, \bar{b} + c, \bar{c} + a, \bar{a} + \bar{b} + \bar{c}\}$. Nous utilisons la même tactique pour obtenir une preuve.

1	$a + b + c$	Hyp
2	$\bar{a} + b$	Hyp
3	$b + c$	Résolvant de 1, 2
4	$\bar{c} + a$	Hyp
5	$b + \bar{c}$	Résolvant de 2, 4
6	b	Résolvant de 3, 5
7	$\bar{b} + c$	Hyp
8	c	Résolvant de 6, 7
9	a	Résolvant de 4, 8
10	$\bar{a} + \bar{b} + \bar{c}$	Hyp
11	$\bar{b} + \bar{c}$	9, 10
12	\bar{c}	Résolvant de 6, 11
13	\perp	Résolvant de 8, 12

Exercice 34, page 64

1. En utilisant les équivalences remarquables nous obtenons $\bar{x} \Leftrightarrow \bar{y} \equiv (\bar{x} \vee \bar{y}) \wedge (x \vee y)$.

2. Formalisation des hypothèses et de la conclusion :

(i) $\bar{c} \Leftrightarrow (o \wedge \neg r)$ donne

— Hyp (1) $c + \bar{o} + r$.

— Hyp (2) $\bar{c} + o$.

— Hyp (3) $\bar{c} + \bar{r}$.

(ii) $e \Rightarrow (r \Leftrightarrow c)$ donne $\bar{e} + c + \bar{r}, \bar{e} + r + \bar{c}$

— Hyp (4) $\bar{e} + c + \bar{r}$.

— Hyp (5) $\bar{e} + r + \bar{c}$.

(iii) $c \wedge \bar{r}$ donne c, \bar{r}

— Hyp (6) c .

— Hyp (7) \bar{r} .

(iv) $o \wedge \bar{e}$ donne pour la négation de la conclusion nous avons $\bar{o} + e$.

— Hyp (8) $\bar{o} + e$

3. Nous appliquons la résolution afin de déduire la véracité du raisonnement.

1	$\bar{c} + o$	Hyp (2)
2	c	Hyp (6)
3	o	Résolvant de 1,2
4	$\bar{e} + r + \bar{c}$	Hyp (5)
5	\bar{r}	Hyp (7)
6	$\bar{e} + \bar{c}$	Résolvant de 4,5
7	\bar{e}	Résolvant de 2, 6
8	$\bar{o} + e$	Hyp (8)
9	\bar{o}	Résolvant de 7,8
10	\perp	Résolvant de 3,9

Exercice 35, page 64

Nous transformons en un ensemble de clauses la conjonction des hypothèses et de la négation de la conclusion, ce qui donne : $\{\bar{r} + t, \bar{q} + t, \bar{i} + \bar{q} + r, q, t + r, \bar{i} + \bar{r}\}$. Nous construisons une preuve par résolution de la correction du raisonnement.

1	$\bar{q} + t$	Hyp
2	q	Hyp
3	t	Résolvant de 1, 2
4	$\bar{i} + \bar{r}$	Hyp
5	\bar{r}	Résolvant de 3, 4
6	$\bar{i} + \bar{q} + r$	Hyp
7	$\bar{q} + r$	Résolvant de 3, 6
8	r	Résolvant de 2, 7
9	\perp	Résolvant de 5,8

Exercice 36, page 64

En français « à moins que » signifie sauf si, sauf quand. Le fait que cela soit des phénomènes météorologiques induit aussi une confusion. Que pensez-vous de la phrase « je suis content sauf quand quelqu'un m'embête », elle a la même signification que la phrase suivante « je suis content à moins que quelqu'un ne m'embête ». Pour se convaincre il convient d'observer la table de vérité correspondante à la négation de l'équivalence.

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

Prenons les conventions suivantes :

- b = il fait beau,
- p = il pleut,
- n = il neige.

Le fait que ce raisonnement est contradictoire se formalise en :

$$\overline{b \Leftrightarrow n}, \overline{p \Leftrightarrow n}, \overline{b \Leftrightarrow p} \models \perp$$

Pour faire une preuve par résolution, nous devons transformer $\overline{b \Leftrightarrow n}, \overline{p \Leftrightarrow n}, \overline{b \Leftrightarrow p}$ en un ensemble de clauses équivalentes. En utilisant l'identité $\overline{x \Leftrightarrow y} = (x + y).(\bar{x} + \bar{y})$, nous obtenons :

$$\{b + n, \bar{b} + \bar{n}, p + n, \bar{p} + \bar{n}, b + p, \bar{b} + \bar{p}\}.$$

Ensuite, il reste à montrer, par résolution, que cet ensemble est contradictoire :

1	$\bar{b} + \bar{p}$	Hyp
2	$p + n$	Hyp
3	$\bar{b} + n$	Résolvant de 1, 2
4	$b + n$	Hyp
5	n	Résolvant de 3, 4
6	$\bar{b} + \bar{n}$	Hyp
7	\bar{b}	Résolvant de 5, 6
8	$\bar{p} + \bar{n}$	Hyp
9	\bar{p}	Résolvant de 5, 8
10	$b + p$	Hyp
11	p	Résolvant de 7, 10
12	\perp	Résolvant de 9, 11

Exercice 37, page 64

Une clause est définie inductivement par :

- \perp est une clause
- une clause non vide est une clause
- un littéral est une clause non vide
- si C est une clause non vide et si L est un littéral alors $C + L$ est une clause non vide

La dernière règle peut aussi être remplacée par l'une des deux règles suivantes :

- si C est une clause non vide et si L est un littéral alors $L + C$ est une clause non vide
- si C et D sont des clauses non vides alors $C + D$ est une clause non vide

Nous définissons alors l'ensemble s par récurrence sur l'ensemble des clauses, en déduisant des règles de construction que toute clause est soit \perp , soit un littéral, soit de la forme $C + L$ où C est une clause non vide et L un littéral :

- $s(\perp) = \emptyset$
- Soit L un littéral, $s(L) = \{L\}$
- Soient L un littéral et C une clause non vide, $s(C + L) = s(C) \cup \{L\}$

Exercice 38, page 64

Soient Γ, Δ deux ensembles de clauses et A, B deux clauses.

1. Monotonie de la déduction. Si $\Gamma \vdash A$ et si Γ est inclus dans Δ alors $\Delta \vdash A$.
2. Composition des déductions. Si $\Gamma \vdash A$, $\Gamma \vdash B$ et si C est un résolvant de A et B alors $\Gamma \vdash C$.

Monotonie : Soit une preuve de C à partir de Γ . Par définition des preuves, toute clause de cette preuve est élément de Γ ou résolvant de deux clauses placées avant dans la preuve. Puisque $\Gamma \subseteq \Delta$, toute clause de cette preuve est également élément de Δ ou résolvant de deux clauses placées avant dans la preuve. Donc c'est aussi une preuve de C à partir de Δ .

Composition : Soient P_A une preuve de A à partir de Γ et P_B une preuve de B à partir de Γ . Toutes les clauses de P_A, P_B, C sont éléments de Γ ou résolvants de deux clauses placées avant dans la preuve. Par suite P_A, P_B, C est une preuve de C à partir de Γ .

Exercice 39, page 65

Montrons que si l'un des parents d'un résolvant est valide, le résolvant est valide ou contient l'autre parent. Soit C un résolvant des clauses D et E . Il existe un littéral L tel que $C = (D - \{L\}) \cup (E - \{L^c\})$. Supposons que D soit valide, donc il existe un littéral M tel que $M, M^c \in D$.

1. $L \neq M$. Alors $M, M^c \in C$, donc C est valide.
2. $L = M$. alors $L^c \in (D - \{L\})$ donc C contient L^c et $(E - \{L^c\})$ donc E .

Ainsi C est valide ou contient E .

Exercice 40, page 65

Par définition du résolvant, les littéraux d'un résolvant de deux clauses sont éléments des deux clauses parentes. Par suite, en *répétant* des résolutions à partir de Γ , nous obtenons des clauses dont tous les littéraux sont éléments de Γ . Formalisons ce raisonnement : soit C une clause déduite de Γ , il y a une suite de clauses B_1, \dots, B_n , preuve de C à partir de Γ . Montrons par induction que pour tout i compris entre 1 et n , B_i (donc $C = B_n$) ne comporte que des littéraux de Γ . Supposons que pour tout j où $1 \leq j \leq n$ et $j < i$, les littéraux de B_j sont des littéraux de Γ et montrons qu'il en est de même de B_i lorsque $1 \leq i \leq n$. Soit i où $1 \leq i \leq n$, nous avons deux cas à considérer :

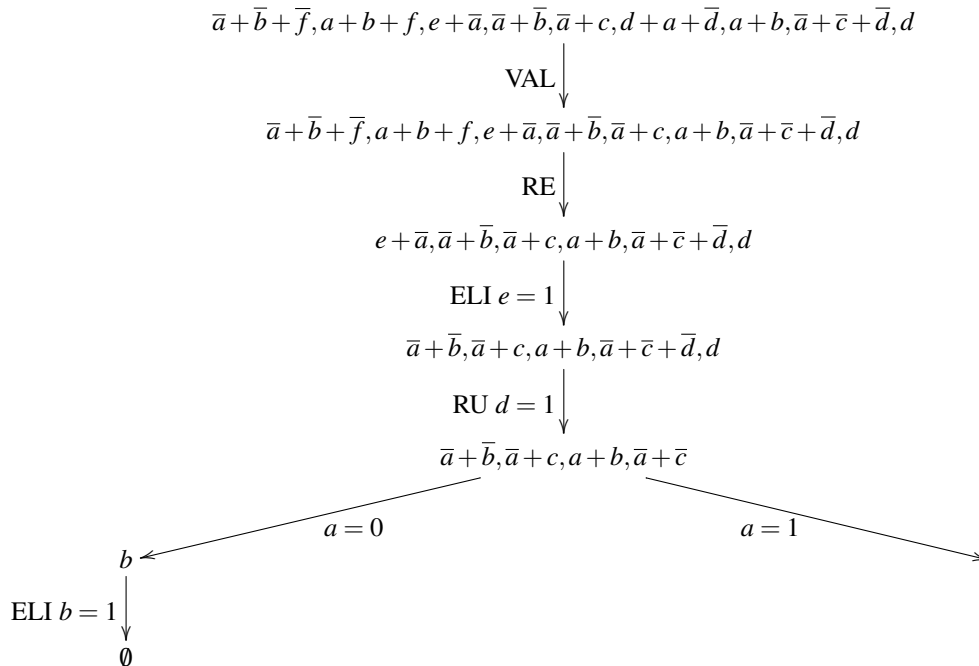
- $B_i \in \Gamma$ et évidemment tous les littéraux de B_i sont des littéraux de Γ .
- B_i est résolvant des clauses B_k et B_l avec $1 \leq k < i$ et $1 \leq l < i$. Par hypothèse de récurrence, les littéraux de B_k et B_l sont des littéraux de Γ . Puisque les littéraux d'un résolvant de deux clauses sont éléments des deux clauses parentes, les littéraux de B_i sont des littéraux de Γ .

Par suite pour tout i où $1 \leq i \leq n$, tous les littéraux de B_i sont des littéraux de Γ .

Exercice 41, page 65

En réduisant l'ensemble, nous obtenons l'ensemble équivalent $\{p + q, p + \bar{r}, \bar{r} + q\}$. Puisqu'aucune résolution n'est possible, l'ensemble est satisfaisable, par complétude de la résolution.

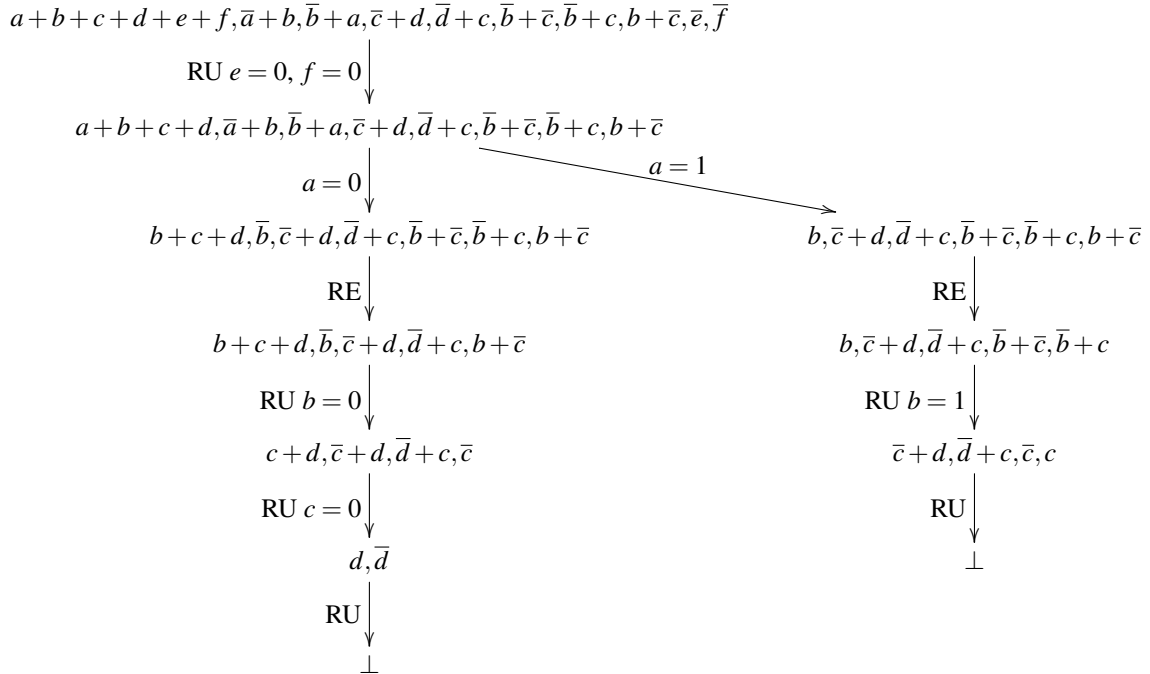
Exercice 42, page 65



Un modèle est $a = 0, b = 1, d = 1, e = 1$.

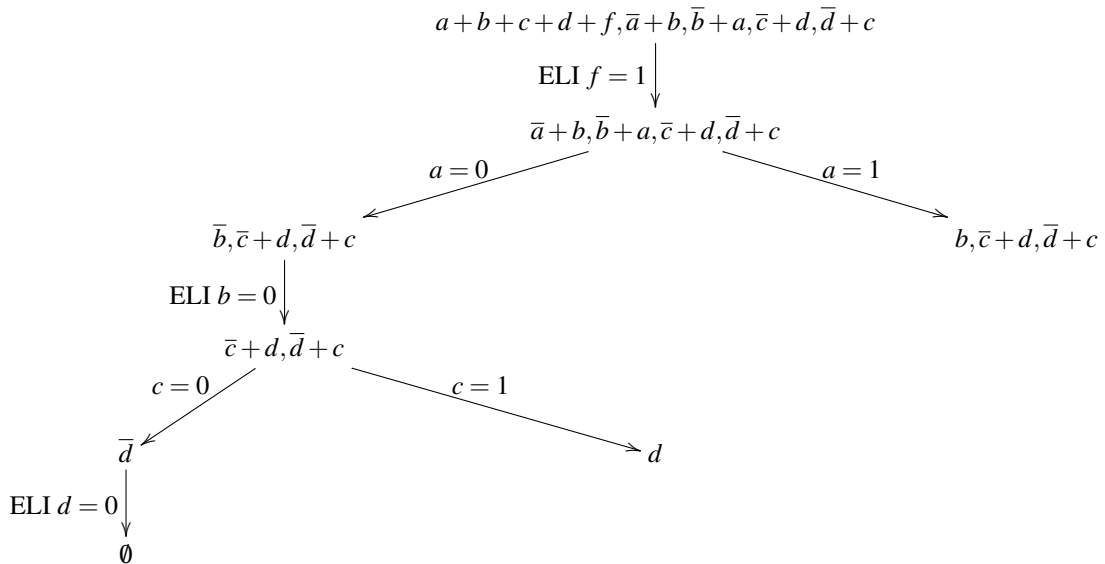
Exercice 43, page 65

- Nous choisissons la variable a comme pivot, nous aurions pu choisir une autre variable comme b ou c mais la trace serait différente de celle-ci.



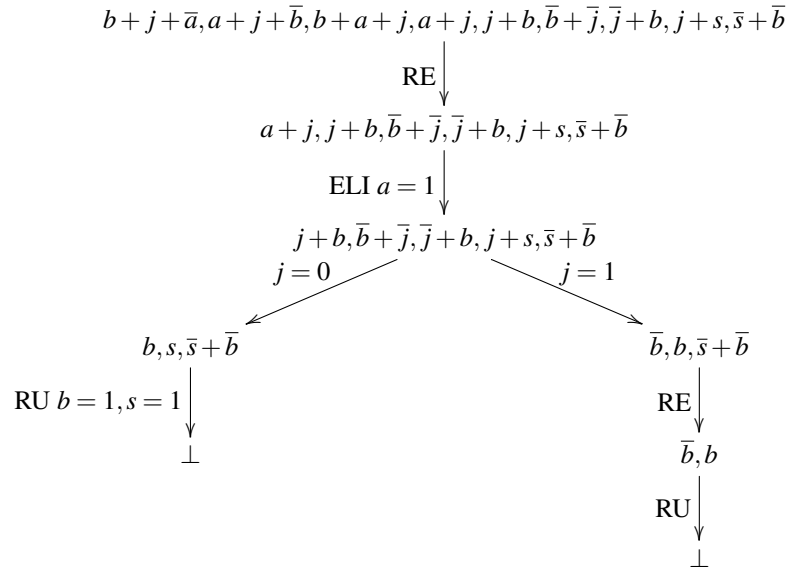
D'après la trace ci-dessus, l'ensemble est insatisfaisable.

- Nous choisissons d'abord la variable a puis c .

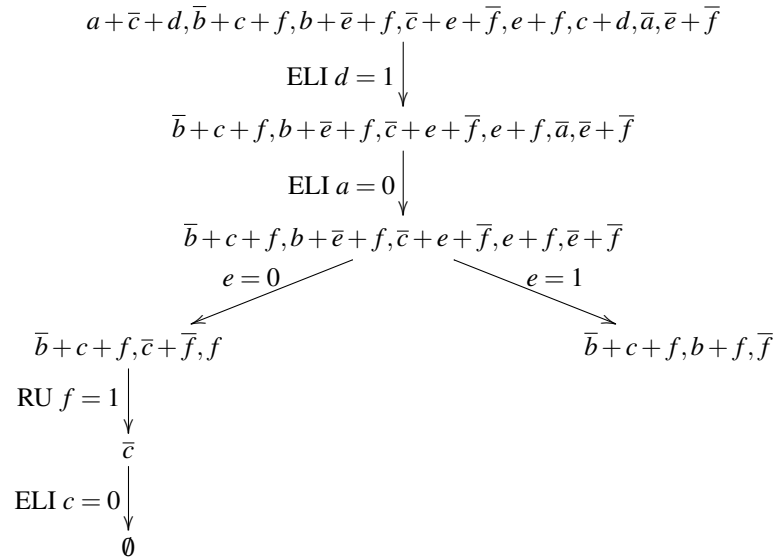


D'après la trace ci-dessus, l'ensemble est satisfaisable. Nous observerons que, d'après le sens de « ou alors », l'algorithme se termine sur la première feuille, où la valeur vrai est retournée. Un modèle est donc $d = 0, c = 1, b = 0, a = 0$, et $f = 1$.

- Nous choisissons la variable j .



— Nous choisissons la variable e .



On trouve un modèle : $d = 1, a = 0, e = 0, f = 1, c = 0$. L'autre branche donne un autre modèle.

Exercice 44, page 65

L'ensemble Δ_0 comprend les clauses :

- 1 $b + \bar{c}$
- 2 $b + c$
- 3 $\bar{a} + \bar{b} + c$
- 4 $a + \bar{b} + c$

Rappelons que Θ_0 est vide. Nous calculons les résolvents :

b	Résolvant de 1 et 2
$\bar{c} + \bar{a} + c$	Résolvant de 1 et 3
$b + \bar{a} + \bar{b}$	Résolvant de 1 et 3
$\bar{c} + a + c$	Résolvant de 1 et 4
$b + a + \bar{b}$	Résolvant de 1 et 4
$\bar{a} + c$	Résolvant de 2 et 3
$a + c$	Résolvant de 2 et 4
$\bar{b} + c$	Résolvant de 3 et 4

Après réduction, l'ensemble Δ_1 comprend les clauses :

5	b	Résolvant de 1 et 2
6	$\bar{a} + c$	Résolvant de 2 et 3
7	$a + c$	Résolvant de 2 et 4
8	$\bar{b} + c$	Résolvant de 3 et 4

L'ensemble Θ_1 est vide. Les nouveaux résolvants calculés sont :

c	Résolvant de 5 et 8
c	Résolvant de 6 et 7

L'ensemble Δ_2 comprend la clause :

9	c	Résolvant de 5 et 8
---	-----	---------------------

L'ensemble Θ_2 comprend la clause :

5	b	Résolvant de 1 et 2
---	-----	---------------------

L'ensemble Δ_3 est vide. L'ensemble Θ_3 comporte les clauses :

9	c	Résolvant de 5 et 8
5	b	Résolvant de 1 et 2

Nous avons obtenu la preuve suivante de c :

1	$b + \bar{c}$	
2	$b + c$	
5	b	Résolvant de 1 et 2
3	$\bar{a} + \bar{b} + c$	
4	$a + \bar{b} + c$	
8	$\bar{b} + c$	Résolvant de 3 et 4
9	c	Résolvant de 5 et 8

Nous avons obtenu la preuve suivante de c :

1	$b + \bar{c}$	
2	$b + c$	
5	b	Résolvant de 1 et 2

L'ensemble de clauses est donc équivalent à la formule $(b.c)$. Donc il a pour modèle $b = 1$, et $c = 1$. Nous résumons l'ensemble de ces calculs dans le tableau suivant :

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Rés de Δ_i & $\Delta_i \cup \Theta_i$
0	$b + \bar{c}, b + c, \bar{a} + \bar{b} + c, a + \bar{b} + c$	\emptyset	$b + \bar{c}, b + c, \bar{a} + \bar{b} + c, a + \bar{b} + c$	$b, \bar{c} + \bar{a} + c, b + \bar{a} + \bar{b}, \bar{c} + a + c, b + a + \bar{b}, \bar{a} + c, a + c, \bar{b} + c$
1	$b, \bar{a} + c, a + c, \bar{b} + c$	\emptyset	$b, \bar{a} + c, a + c, \bar{b} + c$	c
2	c	b	c, b	\emptyset
3	\emptyset	c, b		

Exercice 45, page 65

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Rés de Δ_i & $\Delta_i \cup \Theta_i$
0	$p + q, \bar{p} + s, \bar{s} + t, \bar{t}, \bar{q} + r, \bar{r}$	\emptyset	$p + q, \bar{p} + s, \bar{s} + t, \bar{t}, \bar{q} + r, \bar{r}$	$q + s, p + r, \bar{p} + t, \bar{s}, \bar{q}$
1	$q + s, p + r, \bar{p} + t, \bar{s}, \bar{q}$	$p + q, \bar{p} + s, \bar{t}, \bar{r}$	$q + s, p + r, \bar{p} + t, \bar{s}, \bar{q}, p + q, \bar{p} + s, \bar{t}, \bar{r}$	$s, q, r + t, r + s, p, \bar{p}, t + q$
2	$s, q, r + t, p, \bar{p}$	$\bar{s}, \bar{q}, \bar{t}, \bar{r}$	$s, q, r + t, p, \bar{p}, \bar{s}, \bar{q}, \bar{t}, \bar{r}$	\perp, t, r
3	\perp	\emptyset	\perp	\emptyset
4	\emptyset	\perp		

Nous concluons donc que l'ensemble de clauses est insatisfaisable.

Exercice 46, page 65

Nous donnons tout d'abord la fonction sous forme de table.

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

D'après le théorème 1.6.5 page 33, $f(x, y, z) = (x + y + z) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + y + \bar{z}) \cdot (\bar{x} + \bar{y} + z)$. Appliquons la stratégie complète à l'ensemble de clauses obtenu en remplaçant ci-dessus les points par des virgules. L'ensemble Δ_0 comprend les clauses :

- 1 $x + y + z$
- 2 $x + \bar{y} + \bar{z}$
- 3 $\bar{x} + y + \bar{z}$
- 4 $\bar{x} + \bar{y} + z$

Rappelons que Θ_0 est vide. Nous calculons les résolvants et nous constatons que les seuls résolvants sont des clauses valides. Par suite Δ_1 est vide et $\Theta_1 = \Delta_0$. L'algorithme s'arrête : les clauses initiales sont déjà minimales.

Exercice 47, page 65

Soient Γ un ensemble de clauses et Θ l'ensemble obtenu en enlevant de Γ , les clauses comportant un littéral isolé. Montrons que Γ a un modèle si et seulement si Θ en a un. Si Γ a un modèle, c'est aussi un modèle de Θ , car toute clause de Θ est une clause de Γ . Montrons la réciproque : supposons que Θ ait un modèle v et montrons que Γ a un modèle. Soit v' l'assignation identique à v , sauf que :

- $v'(x) = 1$ si la variable x est un littéral isolé de Γ .
- $v'(y) = 0$ si le littéral \bar{y} est un littéral isolé de Γ .

Notons tout d'abord que v' est bien une application, puisqu'il n'y a pas de variable x telle que x et \bar{x} soient tous deux des littéraux isolés de Γ . Soit C une clause de Γ . Montrons que v' est modèle de C .

- Si C comporte un littéral isolé alors v' est modèle de C , car v' donne la valeur 1 à ce littéral.
- Si C ne comporte pas de littéral isolé alors C ne comporte aucune variable dont la valeur a été redéfinie par v' donc v et v' donnent la même valeur à C . Puisque, dans ce cas, C est une clause de Θ , v est modèle de C donc v' est aussi modèle de C .

Exercice 48, page 65

Rappelons que $s(C)$ est l'ensemble des littéraux de la clause C .

⇒ Supposons que Γ ait un modèle v et montrons que v est modèle de Θ . Puisque Γ a un modèle, L ne contient pas de littéraux complémentaires donc $\Theta \neq \{\perp\}$. Soit C' une clause quelconque de Θ . Il existe une clause C élément de Γ telle que C' est obtenue en enlevant de C les littéraux complémentaires des éléments de L . Puisque v est modèle de Γ , v est modèle de C et des littéraux de L , v donne la valeur 0 aux littéraux enlevés, donc v est modèle d'un littéral non enlevé, donc de C' .

⇐ Supposons que v soit modèle de Θ . Nécessairement Θ ne contient pas la clause vide et L ne comporte pas de littéraux complémentaires. Soit v' l'application identique à v sauf que :

- $v'(x) = 1$ si la variable x est élément de L .
- $v'(x) = 0$ si le littéral \bar{x} est élément de L .

Notons que v' est bien une application, puisque L ne comporte pas de littéraux complémentaires. Soit C une clause quelconque de Γ . Montrons que v' est modèle de C .

- si C comporte un littéral de L , Puisque v' donne la valeur 1 aux éléments de L , v' est modèle de C
- si C ne comporte pas de littéral de L alors il existe une clause C' élément de Θ telle que C' est incluse dans C et ne comporte pas de littéraux complémentaires à ceux de L . Puisque C' ne comporte aucune variable dont la valeur a été redéfinie par v' , les assignations v et v' donnent la même valeur à C' . Puisque v est modèle de C' , v' est aussi modèle de C' . Puisque C' est incluse dans C , v' est modèle de C .

Exercice 49, page 66

1. Ci-dessous, nous montrons que la conjonction des clauses de C'_i est équivalente à c_i , lorsque c_i contient un seul littéral.

$$\begin{aligned}
 & (z_1 + y_1 + y_2) \cdot (z_1 + y_1 + \bar{y}_2) \cdot (z_1 + \bar{y}_1 + y_2) \cdot (z_1 + \bar{y}_1 + \bar{y}_2) \\
 = & (z_1 \cdot z_1 + z_1 \cdot y_1 + z_1 \cdot \bar{y}_2 + y_1 \cdot z_1 + y_1 \cdot y_1 + y_1 \cdot \bar{y}_2 + y_2 \cdot z_1 + y_2 \cdot y_1 + y_2 \cdot \bar{y}_2) \cdot (z_1 + \bar{y}_1 + y_2) \cdot (z_1 + \bar{y}_1 + \bar{y}_2) \\
 = & (z_1 + y_1) \cdot (z_1 + \bar{y}_1 + y_2) \cdot (z_1 + \bar{y}_1 + \bar{y}_2) \\
 = & (z_1 + y_1) \cdot (z_1 \cdot z_1 + z_1 \cdot \bar{y}_1 + z_1 \cdot \bar{y}_2 + \bar{y}_1 \cdot z_1 + \bar{y}_1 \cdot \bar{y}_1 + \bar{y}_1 \cdot \bar{y}_2 + y_2 \cdot z_1 + y_2 \cdot \bar{y}_1 + y_2 \cdot \bar{y}_2) \\
 = & (z_1 + y_1) \cdot (z_1 + \bar{y}_1) \\
 = & z_1 \cdot z_1 + z_1 \cdot \bar{y}_1 + y_1 \cdot z_1 + y_1 \cdot \bar{y}_1 \\
 = & z_1
 \end{aligned}$$

Dans la preuve, nous avons utilisé la commutativité et les identités remarquables : $x \cdot x = x$, $x \cdot \bar{x} = 0$ et $x + (x \cdot y) = x$.

2. Ci-dessous, nous montrons que la conjonction des clauses de C'_i est équivalente à c_i , lorsque c_i contient exactement deux littéraux.

$$\begin{aligned}
 (z_1 + z_2 + y_1) \cdot (z_1 + z_2 + \bar{y}_1) &= z_1 \cdot z_1 + z_1 \cdot z_2 + z_1 \cdot \bar{y}_1 + z_2 \cdot z_1 + z_2 \cdot z_2 + z_2 \cdot \bar{y}_1 + y_1 \cdot z_1 + y_1 \cdot z_2 + y_1 \cdot \bar{y}_1 \\
 &= z_1 + z_2
 \end{aligned}$$

Dans la preuve, nous avons utilisé la commutativité et les identités remarquables : $x \cdot x = x$, $x \cdot \bar{x} = 0$ et $x + (x \cdot y) = x$.

3. Soit v un modèle de c_i . Il existe i , $1 \leq i \leq k$ tel que $[z_i]_v = 1$. Considérons les trois cas suivants :
 - Cas $i \leq 2$: Soit v' une assignation telle que $[z_i]_{v'} = 1$ et pour tout $j \in [1..k-3]$, $[y_j]_{v'} = 0$. v' est modèle de Γ' .
 - Cas $i \geq k-1$: Soit v' une assignation telle que $[z_i]_{v'} = 1$ et pour tout $j \in [1..k-3]$, $[y_j]_{v'} = 1$. v' est modèle de Γ' .
 - Cas $2 < i < k-1$: Soit v' une assignation telle que $[z_i]_{v'} = 1$, pour tout $j \in [1..i-2]$, $[y_j]_{v'} = 1$ et pour tout $j \in [i-1..k-3]$, $[y_j]_{v'} = 0$. v' est modèle de Γ' .
4. Soit v un modèle de C'_i . Considérons les trois cas suivants :
 - Cas $[y_1]_v = 0$: Par définition, $[z_1 + z_2 + y_1]_v = 1$. Donc, $[z_1]_v = 1$ ou $[z_2]_v = 1$. Par suite, $[c_i]_v = 1$.
 - Cas $[y_{k-3}]_v = 1$: Par définition, $[\bar{y}_{k-3} + z_{k-1} + z_k]_v = 1$. Donc, $[z_{k-1}]_v = 1$ ou $[z_k]_v = 1$. Par suite, $[c_i]_v = 1$.
 - Cas $[y_1]_v = 1$ et $[y_{k-3}]_v = 0$: Par construction, il existe une clause de la forme $\bar{y}_{j-2} + z_j + y_{j-1}$ avec $2 < j < k-1$, telle que $[y_{j-2}]_v = 1$ et $[y_{j-1}]_v = 0$. Donc, $[z_j]_v = 1$. Par suite, $[c_i]_v = 1$.
5. Soit $\Gamma = \{z_1 + z_2 + z_3 + z_4\}$. D'après l'algorithme de réduction, $\Gamma' = \{z_1 + z_2 + y_1, \bar{y}_1 + z_3 + z_4\}$. Soit v l'assignation telle que $[z_1]_v = 1$, $[z_2]_v = 0$, $[z_3]_v = 0$, $[z_4]_v = 0$ et $[y_1]_v = 1$. v est un modèle de Γ et un contre-modèle de Γ' .

Exercice 50, page 79

La suite de lignes ci-dessous est un brouillon de preuve jusqu'à la ligne 10 incluse. À la ligne 11, il y a un « Donc » de trop. Nous avons mis les contextes dans la colonne « contexte », correspondants aux numéros des hypothèses.

contexte	numéro	preuve
1	1	Supposons a
1,2	2	Supposons b
1,2	3	c
1	4	Donc d
1,5	5	Supposons e
1,5	6	f
1	7	Donc g
1	8	h
1	9	i
	10	Donc j
	11	Donc k
	12	l

Exercice 51, page 79

1. $a \Rightarrow \neg\neg a$.

contexte	numéro	preuve	règle
1	1	Supposons a	
1,2	2	Supposons $(a \Rightarrow \perp)$	
1,2	3	\perp	$\Rightarrow E$ 1,2
1	4	Donc $\neg\neg a = ((a \Rightarrow \perp) \Rightarrow \perp)$	$\Rightarrow I$ 2,3
	5	Donc $a \Rightarrow \neg\neg a$	$\Rightarrow I$ 1,4

2. $\neg\neg a \Rightarrow a$ c'est la règle RAA.

contexte	numéro	preuve	règle
1	1	Supposons $\neg\neg a$	
1	2	a	RAA 1
	3	Donc $\neg\neg a \Rightarrow a$	$\Rightarrow I$ 1,2

3. $a \Leftrightarrow \neg\neg a$ est la combinaison des deux points précédents suivie de la recomposition des deux implications en l'équivalence.

4. $a \vee \neg a$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg(a \vee \neg a)$	
1,2	2	Supposons a	
1,2	3	$a \vee \neg a$	$\vee I$ 2
1,2	4	\perp	$\Rightarrow E$ 1,3
1	5	Donc $\neg a$	$\Rightarrow I$ 2,4
1	6	$a \vee \neg a$	$\vee I$ 5
1	7	\perp	$\Rightarrow E$ 1,6
	8	Donc $\neg\neg(a \vee \neg a)$	$\Rightarrow I$ 1,7
	9	$a \vee \neg a$	RAA 8

Exercice 52, page 79

1. La preuve de $a \Rightarrow c$ dans l'environnement $a \Rightarrow b, b \Rightarrow c$ est :

environnement			
référence		formule	
<i>i</i>		$a \Rightarrow b$	
<i>ii</i>		$b \Rightarrow c$	
contexte	numéro	preuve	règle
1	1	Supposons a	
1	2	b	$\Rightarrow E\ i,1$
1	3	c	$\Rightarrow E\ ii,2$
	4	Donc $a \Rightarrow c$	$\Rightarrow I\ 1,3$

2. La preuve de $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$ est :

contexte	numéro	preuve	règle
1	1	Supposons $(a \Rightarrow b) \wedge (b \Rightarrow c)$	
1	2	$(a \Rightarrow b)$	$\wedge E\ 1\ 1$
1	3	$(b \Rightarrow c)$	$\wedge E\ 1\ 1$
1,4	4	Supposons a	
1,4	5	b	$\Rightarrow E\ 2,4$
1,4	6	c	$\Rightarrow E\ 3,5$
1	7	Donc $(a \Rightarrow c)$	$\Rightarrow I\ 6,4$
	8	Donc $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$\Rightarrow I\ 1,7$

3. La preuve de $(a \Rightarrow b) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \Rightarrow c))$ est :

contexte	numéro	preuve	règle
1	1	Supposons $(a \Rightarrow b)$	
1,2	2	Supposons $(b \Rightarrow c)$	
1,2,3	3	Supposons a	
1,2,3	4	b	$\Rightarrow E\ 1,3$
1,2,3	5	c	$\Rightarrow E\ 2,4$
1,2	6	Donc $(a \Rightarrow c)$	$\Rightarrow I\ 5,3$
1	7	Donc $(b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$\Rightarrow I\ 6,2$
	8	Donc $(a \Rightarrow b) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \Rightarrow c))$	$\Rightarrow I\ 7,1$

Exercice 53, page 79

- Définition inductive de la fonction dpl :
 - $dpl(\perp) = \perp$.
 - $dpl(\top) = (\perp \Rightarrow \perp)$.
 - $dpl(x) = x$ si x est une variable.
 - $dpl(A \circ B) = (dpl(A) \circ dpl(B))$ si $\circ = \vee, \wedge, \Rightarrow$.
 - $dpl(\neg A) = (dpl(A) \Rightarrow \perp)$.
 - $dpl(A \Leftrightarrow B) = ((dpl(A) \Rightarrow dpl(B)) \wedge (dpl(B) \Rightarrow dpl(A)))$.
- Nous montrons par induction sur la taille d'une formule que les formules A et $dpl(A)$ sont équivalentes. Soit une formule A .

Cas de base : $|A| = 0$, alors A est vrai, faux ou une variable. Si $A = \top$, alors \top et $(\perp \Rightarrow \perp)$ sont équivalentes car $(\perp \Rightarrow \perp)$ est une tautologie. Dans les autres cas, $dpl(A) = A$ donc A et $dpl(A)$ sont équivalentes.

Induction : Supposons que pour toute formule A de taille inférieure ou égale à k , A et $dpl(A)$ soient équivalentes. Montrons que pour une formule F de taille $k + 1$ nous avons bien F et $dpl(F)$ équivalentes. Nous distinguons plusieurs cas :

- $F = \neg A$, par définition $dpl(F) = dpl(\neg A) = (dpl(A) \Rightarrow \perp)$ or par récurrence $dpl(A)$ est équivalente à A donc $(dpl(A) \Rightarrow \perp) \equiv (A \Rightarrow \perp) \equiv \neg A = F$.
- $F = (A \Leftrightarrow B)$, or par définition $dpl(F) = dpl((A \Leftrightarrow B)) = ((dpl(A) \Rightarrow dpl(B)) \wedge (dpl(B) \Rightarrow dpl(A)))$ par récurrence $dpl(A)$ est équivalente à A et $dpl(B)$ est équivalente à B donc $dpl(F) \equiv ((A \Rightarrow B) \wedge (B \Rightarrow A)) \equiv (A \Leftrightarrow B) = F$.
- $F = (A \circ B)$ avec $\circ = \vee, \wedge, \Rightarrow$, par définition $dpl(F) = dpl((A \circ B)) = (dpl(A) \circ dpl(B))$ or par hypothèse de récurrence $dpl(A)$ est équivalente à A et $dpl(B)$ est équivalente à B donc $dpl(F) \equiv (A \circ B) = F$.

Nous en déduisons que deux formules, égales aux abréviations près, sont équivalentes.

Exercice 54, page 79

1. $a \Rightarrow (b \Rightarrow a)$.

contexte	numéro	preuve	règle
1	1	Supposons a	
1,2	2	Supposons b	
1	3	Donc $b \Rightarrow a$	$\Rightarrow I$ 1,2
	4	Donc $a \Rightarrow (b \Rightarrow a)$	$\Rightarrow I$ 1,3

2. $a \wedge b \Rightarrow a$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \wedge b)$	
1	2	a	$\wedge E$ 1
	3	Donc $(a \wedge b) \Rightarrow a$	$\Rightarrow I$ 1,2

3. $\neg a \Rightarrow (a \Rightarrow b)$.

contexte	numéro	preuve	règle
1	1	Supposons $a \Rightarrow \perp$	
1,2	2	Supposons a	
1,2	3	\perp	$\Rightarrow E$ 1,2
1,2	4	b	Efq 3
1	5	Donc $a \Rightarrow b$	$\Rightarrow I$ 2,4
	6	Donc $\neg a \Rightarrow (a \Rightarrow b)$	$\Rightarrow I$ 1,5

4. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$.

contexte	numéro	preuve	règle
1	1	Supposons $a \Rightarrow (b \Rightarrow c)$	
1,2	2	Supposons $a \Rightarrow b$	
1,2,3	3	Supposons a	
1,2,3	4	b	$\Rightarrow E$ 2,3
1,2,3	5	$b \Rightarrow c$	$\Rightarrow E$ 1,3
1,2,3	6	c	$\Rightarrow E$ 4,5
1,2	7	Donc $(a \Rightarrow c)$	$\Rightarrow I$ 6,3
1	8	Donc $(a \Rightarrow b) \Rightarrow (a \Rightarrow c)$	$\Rightarrow I$ 7,2
	9	Donc $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$	$\Rightarrow I$ 8,1

5. $a \wedge b \Rightarrow b \wedge a$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \wedge b)$	
1	2	a	$\wedge E$ 1
1	3	b	$\wedge E$ 1
1	4	$b \wedge a$	$\wedge I$ 2,3
	5	Donc $(a \wedge b) \Rightarrow b \wedge a$	$\Rightarrow I$ 1,4

6. $a \vee b \Rightarrow b \vee a$.

contexte	numéro	preuve	règle
1	1	Supposons $a \vee b$	
1,2	2	Supposons a	
1,2	3	$b \vee a$	$\vee I 2$
1	4	Donc $a \Rightarrow b \vee a$	$\Rightarrow I 2,3$
1,5	5	Supposons b	
1,5	6	$b \vee a$	$\vee I 5$
1	7	Donc $b \Rightarrow b \vee a$	$\Rightarrow I 5,6$
1	8	$b \vee a$	$\vee E 1,4,7$
	9	Donc $a \vee b \Rightarrow b \vee a$	$\Rightarrow I 1,8$

7. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow (a \wedge b \Rightarrow c)$.

contexte	numéro	preuve	règle
1	1	Supposons $a \Rightarrow (b \Rightarrow c)$	
1,2	2	Supposons $a \wedge b$	
1,2	3	a	$\wedge E 1 2$
1,2	4	b	$\wedge E 2 2$
1,2	5	$b \Rightarrow c$	$\Rightarrow E 1,3$
1,2	6	c	$\Rightarrow E 4,5$
1	7	Donc $(a \wedge b \Rightarrow c)$	$\Rightarrow I 6,2$
	8	Donc $(a \Rightarrow (b \Rightarrow c)) \Rightarrow (a \wedge b \Rightarrow c)$	$\Rightarrow I 1,7$

8. $(a \wedge b \Rightarrow c) \Rightarrow (a \Rightarrow (b \Rightarrow c))$.

contexte	numéro	preuve	règle
1	1	Supposons $a \wedge b \Rightarrow c$	
1,2	2	Supposons a	
1,2,3	3	Supposons b	
1,2,3	4	$a \wedge b$	$\wedge I 2,3$
1,2,3	5	c	$\Rightarrow E 4,1$
1,2	6	Donc $b \Rightarrow c$	$\Rightarrow I 3,5$
1	7	Donc $a \Rightarrow (b \Rightarrow c)$	$\Rightarrow I 2,6$
	8	Donc $(a \wedge b \Rightarrow c) \Rightarrow (a \Rightarrow (b \Rightarrow c))$	$\Rightarrow I 1,7$

9. $(a \Rightarrow b) \wedge (c \Rightarrow d) \Rightarrow (a \wedge c \Rightarrow b \wedge d)$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \Rightarrow b) \wedge (c \Rightarrow d)$	
1	2	$a \Rightarrow b$	$\wedge E 1 1$
1	3	$c \Rightarrow d$	$\wedge E 2 1$
1,2	4	Supposons $a \wedge c$	
1,2	5	a	$\wedge E 1 4$
1,2	6	b	$\Rightarrow E 2,5$
1,2	7	c	$\wedge E 2 4$
1,2	8	d	$\Rightarrow E 3,7$
1,2	9	$b \wedge d$	$\wedge I 6,8$
1	10	Donc $a \wedge c \Rightarrow b \wedge d$	$\Rightarrow I 4,9$
	11	Donc $(a \Rightarrow b) \wedge (c \Rightarrow d) \Rightarrow (a \wedge c \Rightarrow b \wedge d)$	$\Rightarrow I 1,10$

Exercice 55, page 80

- 1.
- $a \Rightarrow b$
- dans l'environnement
- $\neg a \vee b$
- .

environnement			
référence		formule	
i		$\neg a \vee b$	
contexte	numéro	preuve	règle
1	1	Supposons a	
1,2	2	Supposons $\neg a$	
1,2	3	\perp	$\Rightarrow E$ 1,2
1,2	4	b	Efq 3
1	5	Donc $\neg a \Rightarrow b$	$\Rightarrow I$ 2,4
1,6	6	Supposons b	
1	7	Donc $b \Rightarrow b$	$\Rightarrow I$ 6,6
1	8	b	$\vee E$ i,5,7
	9	Donc $a \Rightarrow b$	$\Rightarrow I$ 1,8

- 2.
- $(\neg b \Rightarrow \neg a) \Rightarrow (a \Rightarrow b)$
- .

contexte	numéro	preuve	règle
1	1	Supposons $\neg b \Rightarrow \neg a$	
1,2	2	Supposons a	
1,2,3	3	Supposons $\neg b$	
1,2,3	4	$\neg a$	$\Rightarrow E$ 1,3
1,2,3	5	\perp	$\Rightarrow E$ 2,4
1,2	6	Donc $\neg \neg b$	$\Rightarrow I$ 3,5
1,2	7	b	RAA 6
1	8	$a \Rightarrow b$	$\Rightarrow I$ 2,7
	9	$(\neg b \Rightarrow \neg a) \Rightarrow (a \Rightarrow b)$	$\Rightarrow I$ 1,8

Exercice 56, page 80

- 1.
- $\neg(a \wedge \neg a)$
- .

contexte	numéro	preuve	règle
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E$ 1
1	3	$\neg a$	$\wedge E$ 1
1	4	\perp	$\Rightarrow E$ 2,3
	5	Donc $\neg(a \wedge \neg a)$	$\Rightarrow I$ 1,4

- 2.
- $a \vee a \Rightarrow a$
- .

contexte	numéro	preuve	règle
1	1	Supposons $a \vee a$	
1,2	2	Supposons a	
1	3	Donc $a \Rightarrow a$	$\Rightarrow I$ 2,2
1	4	a	$\vee E$ 1,3,3
	5	Donc $a \vee a \Rightarrow a$	$\Rightarrow I$ 1,4

- 3.
- $a \wedge a \Rightarrow a$
- .

contexte	numéro	preuve	règle
1	1	Supposons $a \wedge a$	
1	2	a	$\wedge E1\ 1$
	3	Donc $a \wedge a \Rightarrow a$	$\Rightarrow I\ 1,2$

4. $a \wedge (b \vee c) \Rightarrow a \wedge b \vee a \wedge c$.

contexte	numéro	preuve	règle
1	1	Supposons $a \wedge (b \vee c)$	
1	2	a	$\wedge E1\ 1$
1	3	$b \vee c$	$\wedge E2\ 1$
1,4	4	Supposons b	
1,4	5	$a \wedge b$	$\wedge I\ 2,4$
1,4	6	$a \wedge b \vee a \wedge c$	$\vee I1\ 5$
1	7	Donc $b \Rightarrow a \wedge b \vee a \wedge c$	$\Rightarrow I\ 4,6$
1,8	8	Supposons c	
1,8	9	$a \wedge c$	$\wedge I\ 2,8$
1,8	10	$a \wedge b \vee a \wedge c$	$\vee I2\ 9$
1	11	Donc $c \Rightarrow a \wedge b \vee a \wedge c$	$\Rightarrow I\ 8,10$
1	12	$a \wedge b \vee a \wedge c$	$\vee E\ 3,7,11$
	13	Donc $a \wedge (b \vee c) \Rightarrow a \wedge b \vee a \wedge c$	$\Rightarrow I\ 1,12$

5. $a \wedge b \vee a \wedge c \Rightarrow a \wedge (b \vee c)$.

contexte	numéro	preuve	règle
1	1	Supposons $a \wedge b \vee a \wedge c$	
1,2	2	Supposons $a \wedge b$	
1,2	3	a	$\wedge E1\ 2$
1,2	4	b	$\wedge E2\ 2$
1,2	5	$b \vee c$	$\vee I1\ 4$
1,2	6	$a \wedge (b \vee c)$	$\wedge I\ 3,5$
1	7	Donc $a \wedge b \Rightarrow a \wedge (b \vee c)$	$\Rightarrow I\ 2,6$
1,8	8	Supposons $a \wedge c$	
1,8	9	a	$\wedge E1\ 8$
1,8	10	c	$\wedge E2\ 8$
1,8	11	$b \vee c$	$\vee I2\ 10$
1,8	12	$a \wedge (b \vee c)$	$\wedge I\ 9,11$
1	13	Donc $a \wedge c \Rightarrow a \wedge (b \vee c)$	$\Rightarrow I\ 8,12$
1	14	$a \wedge (b \vee c)$	$\vee E\ 1,7,13$
	15	Donc $a \wedge b \vee a \wedge c \Rightarrow a \wedge (b \vee c)$	$\Rightarrow I\ 1,14$

6. $a \vee b \wedge c \Rightarrow (a \vee b) \wedge (a \vee c)$.

contexte	numéro	preuve	règle
1	1	Supposons $a \vee b \wedge c$	
1,2	2	Supposons a	
1,2	3	$a \vee b$	$\vee I1$ 2
1,2	4	$a \vee c$	$\vee I1$ 2
1,2	5	$(a \vee b) \wedge (a \vee c)$	$\wedge I$ 3,4
1	6	Donc $a \Rightarrow (a \vee b) \wedge (a \vee c)$	$\Rightarrow I$ 2,5
1,7	7	Supposons $b \wedge c$	
1,7	8	b	$\wedge E1$ 7
1,7	9	$a \vee b$	$\vee I2$ 8
1,7	10	c	$\wedge E2$ 7
1,7	11	$a \vee c$	$\vee I2$ 10
1,7	12	$(a \vee b) \wedge (a \vee c)$	$\wedge I$ 9,11
1	13	Donc $b \wedge c \Rightarrow (a \vee b) \wedge (a \vee c)$	$\Rightarrow I$ 7,12
1	14	$(a \vee b) \wedge (a \vee c)$	$\vee E$ 1,6,13
	15	Donc $a \vee b \wedge c \Rightarrow (a \vee b) \wedge (a \vee c)$	$\Rightarrow I$ 1,14

7. $(a \vee b) \wedge (a \vee c) \Rightarrow a \vee b \wedge c$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \vee b) \wedge (a \vee c)$	
1	2	$a \vee b$	$\wedge E1$ 1
1	3	$a \vee c$	$\wedge E2$ 1
1,4	4	Supposons a	
1,4	5	$a \vee b \wedge c$	$\vee I1$ 4
1	6	Donc $a \Rightarrow a \vee b \wedge c$	$\Rightarrow I$ 4,5
1,7	7	Supposons b	
1,7,8	8	Supposons c	
1,7,8	9	$b \wedge c$	$\wedge I$ 7,8
1,7,8	10	$a \vee b \wedge c$	$\vee I2$ 9
1,7	11	Donc $c \Rightarrow a \vee b \wedge c$	$\Rightarrow I$ 8,10
1,7	12	$a \vee b \wedge c$	$\vee E$ 3,6,11
1	13	Donc $b \Rightarrow a \vee b \wedge c$	$\Rightarrow I$ 7,12
1	14	$a \vee b \wedge c$	$\vee E$ 2,6,13
	15	Donc $(a \vee b) \wedge (a \vee c) \Rightarrow a \vee b \wedge c$	$\Rightarrow I$ 1,14

Exercice 57, page 80

1. $P1$ est une preuve de $B \vee C$ dans l'environnement $\neg B \Rightarrow C$:

environnement			
référence		formule	
i		$\neg B \Rightarrow C$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg(B \vee C)$	
1,2	2	Supposons $\neg B$	
1,2	3	C	$\Rightarrow E$ 2, i
1,2	4	$B \vee C$	$\vee I$ 2 3
1,2	5	\perp	$\Rightarrow E$ 1,4
1	6	Donc $\neg\neg B$	$\Rightarrow I$ 2,5
1	7	B	RAA 6
1	8	$B \vee C$	$\vee I$ 1 7
1	9	\perp	$\Rightarrow E$ 1,8
	10	Donc $\neg\neg(B \vee C)$	$\Rightarrow I$ 1,9
	11	$B \vee C$	RAA 10

Nous donnons une autre preuve de la même formule :

environnement			
référence		formule	
i		$\neg B \Rightarrow C$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg(B \vee C)$	
1,2	2	Supposons B	
1,2	3	$B \vee C$	$\vee I$ 1 2
1,2	4	\perp	$\Rightarrow E$ 1,3
1	5	Donc $\neg B$	$\Rightarrow I$ 2,4
1	6	C	$\Rightarrow E$ i ,5
1	7	$B \vee C$	$\vee I$ 2 6
1	8	\perp	$\Rightarrow E$ 1,7
	9	Donc $\neg\neg(B \vee C)$	$\Rightarrow I$ 1,8
	10	$B \vee C$	RAA 9

2. $P2$ est une preuve de $\neg B \vee C$ dans l'environnement $B \Rightarrow C$:

environnement			
référence		formule	
i		$B \Rightarrow C$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg(\neg B \vee C)$	
1,2	2	Supposons $\neg B$	
1,2	3	$\neg B \vee C$	$\vee I$ 1 2
1,2	4	\perp	$\Rightarrow E$ 3,1
1	5	Donc $\neg\neg B$	$\Rightarrow E$ 2,4
1	6	B	RAA 5
1	7	C	$\Rightarrow E$ i ,6
1	8	$\neg B \vee C$	$\vee I$ 2 7
1	9	\perp	$\Rightarrow E$ 1,8
	10	Donc $\neg\neg(\neg B \vee C)$	$\Rightarrow E$ 1,9
	11	$\neg B \vee C$	RAA 10

3. $P3$ est une preuve de $\neg B \vee \neg C$ dans l'environnement $\neg(B \wedge C)$:

environnement			
référence		formule	
i		$\neg(B \wedge C)$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg(\neg B \vee \neg C)$	
1,2	2	Supposons $\neg B$	
1,2	3	$\neg B \vee \neg C$	$\vee I1\ 2$
1,2	4	\perp	$\Rightarrow E\ 1,3$
1	5	Donc $\neg\neg B$	$\Rightarrow I\ 2,4$
1	6	B	$RAA\ 5$
1,7	7	Supposons $\neg C$	
1,7	8	$\neg B \vee \neg C$	$\vee I2\ 7$
1,7	9	\perp	$\Rightarrow E\ 8,1$
1	10	Donc $\neg\neg C$	$\Rightarrow I\ 7,9$
1	11	C	$RAA\ 10$
1	12	$B \wedge C$	$\wedge I\ 6,11$
1	13	\perp	$\Rightarrow E\ i,12$
	14	Donc $\neg\neg(\neg B \vee \neg C)$	$\Rightarrow I\ 1,13$
	15	$\neg B \vee \neg C$	$RAA\ 14$

4. $P4$ est une preuve de $\neg B$ dans l'environnement $\neg(B \vee C)$:

environnement			
référence		formule	
i		$\neg(B \vee C)$	
contexte	numéro	preuve	règle
1	1	Supposons B	
1	2	$B \vee C$	$\vee I1\ 1$
1	3	\perp	$\Rightarrow E\ i,2$
	4	Donc $\neg B$	$\Rightarrow I\ 1,3$

5. $P5$ est une preuve de $\neg C$ dans ce même environnement :

environnement			
référence		formule	
i		$\neg(B \vee C)$	
contexte	numéro	preuve	règle
1	1	Supposons C	
1	2	$B \vee C$	$\vee I2\ 1$
1	3	\perp	$\Rightarrow E\ i,2$
	4	Donc $\neg C$	$\Rightarrow I\ 1,3$

6. $P6$ est une preuve de B dans l'environnement $\neg(B \Rightarrow C)$:

environnement			
référence		formule	
i		$\neg(B \Rightarrow C)$	
contexte	numéro	preuve	règle
1	1	Supposons $\neg B$	
1,2	2	Supposons B	
1,2	3	\perp	$\Rightarrow E$ 2,1
1,2	4	C	Efq 3
1	5	Donc $B \Rightarrow C$	$\Rightarrow I$ 2,4
1	6	\perp	$\Rightarrow E$ i,5
	7	Donc $\neg\neg B$	$\Rightarrow I$ 1,6
	8	B	RAA 7

7. $P7$ est une preuve de $\neg C$ dans le même environnement :

environnement			
référence		formule	
i		$\neg(B \Rightarrow C)$	
contexte	numéro	preuve	règle
1	1	Supposons C	
1,2	2	Supposons B	
1	3	Donc $B \Rightarrow C$	$\Rightarrow I$ 1,2
1	4	\perp	$\Rightarrow E$ i,3
	5	Donc $\neg C$	$\Rightarrow I$ 1,4

Exercice 58, page 80

1. $\neg(a \vee b) \Rightarrow (\neg a \wedge \neg b)$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg(a \vee b)$	
1,2	2	Supposons a	
1,2	3	$a \vee b$	$\vee I$ 2
1,2	4	\perp	$\Rightarrow E$ 1,3
1	5	Donc $a \Rightarrow \perp$	$\Rightarrow I$ 2,4
1,6	6	Supposons b	
1,6	7	$a \vee b$	$\vee I$ 5
1,6	8	\perp	$\Rightarrow E$ 1,6
1	9	Donc $b \Rightarrow \perp$	$\Rightarrow I$ 7,8
1	10	$(\neg a \wedge \neg b)$	$\wedge I$ 5,9
	11	Donc $\neg(a \vee b) \Rightarrow (\neg a \wedge \neg b)$	$\Rightarrow I$ 1,10

2. $(\neg a \wedge \neg b) \Rightarrow \neg(a \vee b)$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg a \wedge \neg b$	
1,2	2	Supposons $a \vee b$	
1,2	3	$\neg a$	$\wedge E$ 1
1,2	4	$\neg b$	$\wedge E$ 1
1,2	5	\perp	$\vee E$ 2,3,4
1	6	Donc $\neg(a \vee b)$	$\Rightarrow I$ 2,5
	7	Donc $(\neg a \wedge \neg b) \Rightarrow \neg(a \vee b)$	$\Rightarrow I$ 1,6

3. $\neg(a \wedge b) \Rightarrow (\neg a \vee \neg b)$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg(a \wedge b)$	
1,2	2	Supposons $\neg(\neg a \vee \neg b)$	
1,2,3	3	Supposons a	
1,2,3,4	4	Supposons b	
1,2,3,4	5	$a \wedge b$	$\wedge I$ 3,4
1,2,3,4	6	\perp	$\Rightarrow E$ 1,5
1,2,3	7	Donc $b \Rightarrow \perp$	$\Rightarrow I$ 6,4
1,2,3	8	$\neg a \vee \neg b$	$\vee I2$ 7
1,2,3	9	\perp	$\Rightarrow E$ 2,8
1,2	10	Donc $a \Rightarrow \perp$	$\Rightarrow I$ 3,9
1,2	11	$(\neg a \vee \neg b)$	$\vee I1$ 10
1,2	12	\perp	$\Rightarrow E$ 2,11
1	13	Donc $\neg(\neg a \vee \neg b) \Rightarrow \perp$	$\Rightarrow I$ 2,12
1	14	$\neg a \vee \neg b$	RAA 13
	15	Donc $\neg(a \wedge b) \Rightarrow (\neg a \vee \neg b)$	$\Rightarrow I$ 1,14

4. $(\neg a \vee \neg b) \Rightarrow \neg(a \wedge b)$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg a \vee \neg b$	
1,2	2	Supposons $a \wedge b$	
1,2	3	a	$\wedge E1$ 2
1,2	4	b	$\wedge E2$ 2
1,2,5	5	Supposons $\neg a$	
1,2,5	6	\perp	$\Rightarrow E$ 3,5
1,2,5	7	$\neg b$	Efq 6
1,2	8	Donc $\neg a \Rightarrow \neg b$	$\Rightarrow I$ 7,5
1,2,9	9	Supposons $\neg b$	
1,2	10	Donc $\neg b \Rightarrow \neg b$	$\Rightarrow I$ 9,9
1,2	11	$\neg b$	$\vee E$ 1,8,10
1,2	12	\perp	$\Rightarrow E$ 11,4
1	13	Donc $\neg(a \wedge b)$	$\Rightarrow I$ 2,12
	14	Donc $(\neg a \vee \neg b) \Rightarrow \neg(a \wedge b)$	$\Rightarrow I$ 1,13

5. $(a \vee b) \wedge (\neg a \vee \neg b) \Rightarrow b$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \vee b) \wedge (\neg a \vee b)$	
1	2	$(a \vee b)$	$\wedge E 1$
1	3	$(\neg a \vee b)$	$\wedge E 1$
1,4	4	Supposons a	
1,4,5	5	Supposons $\neg a$	
1,4,5	6	\perp	$\Rightarrow E 4,5$
1,4,5	7	b	$Efq 6$
1,4	8	Donc $\neg a \Rightarrow b$	$\Rightarrow I 7,5$
1,4,9	9	Supposons b	
1,4	10	Donc $b \Rightarrow b$	$\Rightarrow I 9,9$
1,4	11	b	$\vee E 3,8,10$
1	12	Donc $a \Rightarrow b$	$\Rightarrow I 4,11$
1,13	13	Supposons b	
1	14	Donc $b \Rightarrow b$	$\Rightarrow I 9,9$
1	15	b	$\vee E 2,12,14$
	16	Donc $(a \vee b) \wedge (\neg a \vee b) \Rightarrow b$	$\Rightarrow I 1,15$

6. $(a \vee b) \wedge (\neg b \vee c) \Rightarrow a \vee c$.

contexte	numéro	preuve	règle
1	1	Supposons $(a \vee b) \wedge (\neg b \vee c)$	
1	2	$(a \vee b)$	$\wedge E 1$
1	3	$(\neg b \vee c)$	$\wedge E 1$
1,4	4	Supposons a	
1,4	5	$a \vee c$	$\vee I 4$
1	6	Donc $a \Rightarrow a \vee c$	$\Rightarrow I 4,5$
1,7	7	Supposons b	
1,7,8	8	Supposons $\neg b$	
1,7,8	9	\perp	$\Rightarrow E 7,8$
1,7,8	10	c	$Efq 9$
1,7	11	Donc $\neg b \Rightarrow c$	$\Rightarrow I 10,8$
1,7,12	12	Supposons c	
1,7	13	Donc $c \Rightarrow c$	$\Rightarrow I 12,12$
1,7	14	c	$\vee E 3,11,13$
1,7	15	$a \vee c$	$\vee I 2 14$
1	16	Donc $b \Rightarrow a \vee c$	$\Rightarrow I 7,15$
1	17	$a \vee c$	$\vee E 2,6,16$
	18	Donc $(a \vee b) \wedge (\neg b \vee c) \Rightarrow a \vee c$	$\Rightarrow I 1,17$

Exercice 59, page 80

1. $\neg p \wedge \neg(\neg p \wedge q) \Rightarrow \neg q$.

contexte	numéro	ligne	règle
1	1	Supposons $\neg p \wedge \neg(\neg p \wedge q)$	
1,2	2	Supposons q	
1,2	3	$\neg p$	$\wedge E1\ 1$
1,2	4	$\neg(\neg p \wedge q)$	$\wedge E2\ 1$
1,2	5	$\neg p \wedge q$	$\wedge I\ 3,2$
1,2	6	\perp	$\Rightarrow E\ 4,5$
1	7	Donc $\neg q$	$\Rightarrow I\ 2,6$
	8	Donc $\neg p \wedge \neg(\neg p \wedge q) \Rightarrow \neg q$	$\Rightarrow I\ 1,7$

2. $((p \wedge q) \wedge (p \wedge r)) \wedge \neg p \Rightarrow q \vee r$.

contexte	numéro	ligne	règle
1	1	Supposons $((p \wedge q) \wedge (p \wedge r)) \wedge \neg p$	
1	2	$((p \wedge q) \wedge (p \wedge r))$	$\wedge E1\ 1$
1	3	$\neg p$	$\wedge E2\ 1$
1	4	$p \wedge r$	$\wedge E2\ 2$
1	5	p	$\wedge E1\ 4$
1	6	\perp	$\Rightarrow E\ 3,5$
1	7	$q \vee r$	$Efq\ 6$
	8	Donc $((p \wedge q) \wedge (p \wedge r)) \wedge \neg p \Rightarrow q \vee r$	$\Rightarrow I\ 1,7$

3. (*) $\neg p \vee \neg(p \wedge q) \Rightarrow \neg q \vee \neg p$.

contexte	numéro	ligne	règle
1	1	Supposons $\neg p \vee \neg(p \wedge q)$	
1,2	2	Supposons $\neg p$	
1,2	3	$\neg q \vee \neg p$	$\vee I 2$
1	4	Donc $\neg p \Rightarrow \neg q \vee \neg p$	$\Rightarrow I 2,3$
1,5	5	Supposons $\neg(p \wedge q)$	
1,5,6	6	Supposons $\neg(\neg q \vee \neg p)$	
1,5,6,7	7	Supposons $\neg q$	
1,5,6,7	8	$\neg q \vee \neg p$	$\vee I 1 7$
1,5,6,7	9	\perp	$\Rightarrow E 6,8$
1,5,6	10	Donc $\neg \neg q$	$\Rightarrow I 7,9$
1,5,6,11	11	Supposons p	
1,5,6,7	12	q	$RAA 10$
1,5,6,7,13	13	Supposons $\neg(\neg p \vee \neg q)$	
1,5,6,7,13	14	$p \wedge q$	$\wedge I 11,12$
1,5,6,7,13	15	\perp	$\Rightarrow E 5,14$
1,5,6,7	16	Donc $\neg \neg(\neg p \vee \neg q)$	$\Rightarrow I 13,15$
1,5,6,7	17	$\neg p \vee \neg q$	$RAA 16$
1,5,6,7,18	18	Supposons $\neg p$	
1,5,6,7,18	19	\perp	$\Rightarrow E 18,11$
1,5,6,7	20	Donc $\neg p \Rightarrow \perp$	$\Rightarrow I 18,19$
1,5,6,7,21	21	Supposons $\neg q$	
1,5,6,7,21	22	\perp	$\Rightarrow E 21,12$
1,5,6,7	23	Donc $\neg q \Rightarrow \perp$	$\Rightarrow I 21,22$
1,5,6,7	24	\perp	$\vee E 17,20,23$
1,5,6	25	Donc $\neg p$	$\Rightarrow I 11,24$
1,5,6	26	$\neg q \vee \neg p$	$\Rightarrow E 4,25$
1,5,6	27	\perp	$\Rightarrow E 6,26$
1,5	28	Donc $\neg \neg(\neg q \vee \neg p)$	$\Rightarrow I 6,27$
1,5	29	$\neg q \vee \neg p$	$RAA 28$
1	30	Donc $\neg(p \wedge q) \Rightarrow \neg q \vee \neg p$	$\Rightarrow I 5,29$
1	31	$\neg q \vee \neg p$	$\vee E 1,4,30$
	32	Donc $\neg p \vee \neg(p \wedge q) \Rightarrow \neg q \vee \neg p$	$\Rightarrow I 1,31$

Exercice 60, page 80

1. $(p \vee q) \Rightarrow (\neg p \wedge \neg q) \Rightarrow r$.

contexte	numéro	preuve	règle
1	1	Supposons $p \vee q$	
1,2	2	Supposons $\neg p \wedge \neg q$	
1,2	3	$\neg p$	$\wedge E 1 2$
1,2	4	$\neg q$	$\wedge E 2 2$
1,2	5	\perp	$\vee E 1,3,4$
1,2	6	r	$Efq 5$
1	7	Donc $(\neg p \wedge \neg q) \Rightarrow r$	$\Rightarrow I 2,6$
	8	Donc $(p \vee q) \Rightarrow (\neg p \wedge \neg q) \Rightarrow r$	$\Rightarrow I 1,7$

2. $((p \Rightarrow q) \wedge (q \Rightarrow r)) \wedge \neg r \Rightarrow \neg p$.

contexte	numéro	preuve	règle
1	1	Supposons $((p \Rightarrow q) \wedge (q \Rightarrow r)) \wedge \neg r$	
1,2	2	Supposons p	
1,2	3	$(p \Rightarrow q) \wedge (q \Rightarrow r)$	$\wedge E$ 1
1,2	4	$\neg r$	$\wedge E$ 1
1,2	5	$p \Rightarrow q$	$\wedge E$ 3
1,2	6	$q \Rightarrow r$	$\wedge E$ 3
1,2	7	q	$\Rightarrow E$ 5,2
1,2	8	r	$\Rightarrow E$ 6,7
1,2	9	\perp	$\Rightarrow E$ 4,8
1	10	Donc $\neg p$	$\Rightarrow I$ 2,9
	11	Donc $((p \Rightarrow q) \wedge (q \Rightarrow r)) \wedge \neg r \Rightarrow \neg p$	$\Rightarrow I$ 1,10

3. $(p \Rightarrow q) \Rightarrow ((p \wedge q) \vee \neg p)$.

contexte	numéro	preuve	règle
1	1	Supposons $p \Rightarrow q$	
1,2	2	Supposons $\neg((p \wedge q) \vee \neg p)$	
1,2,3	3	Supposons p	
1,2,3	4	q	$\Rightarrow E$ 1,3
1,2,3	5	$p \wedge q$	$\wedge I$ 3,4
1,2,3	6	$(p \wedge q) \vee \neg p$	$\vee I$ 5
1,2,3	7	\perp	$\Rightarrow E$ 2,6
1,2	8	Donc $\neg p$	$\Rightarrow I$ 3,7
1,2	9	$(p \wedge q) \vee \neg p$	$\vee I$ 8
1,2	10	\perp	$\Rightarrow E$ 2,9
1	11	Donc $\neg \neg((p \wedge q) \vee \neg p)$	$\Rightarrow I$ 2,10
1	12	$(p \wedge q) \vee \neg p$	RAA 11
	13	Donc $(p \Rightarrow q) \Rightarrow ((p \wedge q) \vee \neg p)$	$\Rightarrow I$ 1,12

Exercice 61, page 81

1. $(p \vee q) \wedge (p \Rightarrow r) \Rightarrow q \vee r$

contexte	numéro	ligne	règle
1	1	Supposons $(p \vee q) \wedge (p \Rightarrow r)$	
1	2	$p \vee q$	$\wedge E$ 1
1	3	$p \Rightarrow r$	$\wedge E$ 1
1,4	4	Supposons p	
1,4	5	r	$\Rightarrow E$ 3,4
1,4	6	$q \vee r$	$\vee I$ 5
1	7	Donc $p \Rightarrow q \vee r$	$\Rightarrow I$ 4,6
1,8	8	Supposons q	
1,8	9	$q \vee r$	$\vee I$ 8
1	10	Donc $q \Rightarrow q \vee r$	$\Rightarrow I$ 8,9
1	11	$q \vee r$	$\vee E$ 7,10
	12	Donc $(p \vee q) \wedge (p \Rightarrow r) \Rightarrow q \vee r$	$\Rightarrow I$ 1,11

2. En utilisant :

- p : "Pierre est grand"
- j : "Jean est le fils de Pierre"

— m : "Marie est la soeur de Jean"

— (H1) : $p \Rightarrow \neg j$

— (H2) : $\neg p \Rightarrow j$

— (H3) : $j \Rightarrow m$

Nous démontrons que $H1 \wedge H2 \wedge H3 \Rightarrow C$:

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

contexte	numéro	preuve	règle
1	1	supposons $(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m)$	
1	2	$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j)$	$\wedge E1\ 1$
1	3	$(j \Rightarrow m)$	$\wedge E1\ 2$
1	4	$(\neg p \Rightarrow j)$	$\wedge E2\ 2$
1, 5	5	supposons $\neg(m \vee p)$	
1, 5, 6	6	supposons p	
1, 5, 6	7	$m \vee p$	$\vee I\ 6$
1, 5, 6	8	\perp	$\Rightarrow E\ 5, 7$
1, 5	9	donc $\neg p$	$\Rightarrow I\ 6, 8$
1, 5	10	j	$\Rightarrow E\ 4, 9$
1, 5	11	m	$\Rightarrow E\ 3, 10$
1, 5	12	$m \vee p$	$\vee I\ 11$
1, 5	13	\perp	$\Rightarrow E\ 5, 12$
1	14	donc $\neg\neg(m \vee p)$	$\Rightarrow I\ 5, 13$
1	15	$m \vee p$	$RAA\ 14$
	16	donc $(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$	$\Rightarrow I\ 1, 15$

Exercice 62, page 104

Nous donnons les structures grâce à la syntaxe des formules sans priorité :

1. La formule $\forall x(P(x) \Rightarrow \exists y Q(x, y))$ ne contient pas de variable libre.
2. La structure de la formule $\forall a \forall b (b \neq 0 \Rightarrow \exists q \exists r (a = b * q + r \wedge r < b))$ est $\forall a \forall b (\neq(b, 0) \Rightarrow \exists q \exists r (= (a, +(* (b, q), r)) \wedge <(r, b)))$. Cette formule ne contient pas de variable libre.
3. La structure de la formule $Pair(x) \Leftrightarrow \exists y (x = 2 * y)$ est $(Pair(x) \Leftrightarrow \exists y = (x, *(2, y)))$, et x est une variable libre.
4. La structure de la formule $x \text{ Divide } y \Leftrightarrow \exists z (y = z * x)$ est $(Divise(x, y) \Leftrightarrow \exists z = (y, *(z, x)))$, et x et y sont des variables libres.
5. La structure de la fonction $Premier(x) \Leftrightarrow \forall y (y \text{ Divide } x \Rightarrow y = 1 \vee y = x)$ est $(Premier(x) \Leftrightarrow \forall y (Divise(y, x) \Rightarrow (= (y, 1) \vee = (y, x))))$, et x est une variable libre.

Exercice 63, page 104

1. Tout frère du père de Robert est un oncle de Robert :

$$\forall x (f(x, p(r)) \Rightarrow o(x, r)).$$

2. Si les pères de deux enfants sont des frères alors ces deux enfants sont des cousins :

$$\forall x \forall y (f(p(x), p(y)) \Rightarrow c(x, y)).$$

3. Robert a un cousin plus jeune qu'un des frères de Robert :

$$\exists x (c(x, r) \wedge \exists y (f(y, r) \wedge j(x, y))).$$

1. $\nexists x j(p(x), x)$: personne n'est plus âgé que son père
2. $\forall x \forall y (p(p(x)) = p(p(y)) \Rightarrow c(x, y))$: deux personnes qui ont le même grand-père paternel sont cousines
3. $\forall x \exists y (f(x, y) \wedge j(x, y))$: tout le monde a un frère aîné (impossible mais ça n'empêche pas de l'exprimer...)
4. $\exists x \exists y (f(x, y) \wedge \neg(p(x) = p(y)))$: il y a deux frères qui n'ont pas le même père

Exercice 64, page 104

1. L'équipe de France a gagné un match et en a perdu un :

$$\exists x G(f, x) \wedge \exists x G(x, f).$$

2. L'équipe de France et l'équipe d'Allemagne ont fait match nul :

$$J(f, a) \wedge \neg G(f, a) \wedge \neg G(a, f).$$

3. Une équipe a gagné tous ses matchs :

$$\exists x (\forall y (J(x, y) \Rightarrow G(x, y))).$$

4. Aucune équipe n'a perdu tous ses matchs :

$$\neg (\exists x (\forall y (J(x, y) \Rightarrow G(y, x)))).$$

5. Les deux formules ci-dessous sont équivalentes et expriment la phrase :

Tous ceux qui ont joué contre une équipe qui a gagné tous ses matchs, ont gagné au moins un match.

$$\forall x (\exists y (J(x, y) \wedge \forall z (J(y, z) \Rightarrow G(y, z))) \Rightarrow \exists v G(x, v))$$

$$\forall x \forall y (J(x, y) \wedge \forall z (J(y, z) \Rightarrow G(y, z)) \Rightarrow \exists v G(x, v))$$

En réécrivant les formules selon : $\forall x (A \Rightarrow B) \equiv \exists x A \Rightarrow B$ si x n'apparaît pas dans B , la solution devient évidente.

Exercice 65, page 105

$$\Sigma = \{t^{f1}, s^{f1}, A^{r2}, C^{r1}, D^{r1}, E^{r1}, O^{r1}, P^{r2}\}.$$

1. Les chiens et les oiseaux sont des animaux domestiques : $\forall x (C(x) \vee O(x) \Rightarrow D(x))$.
Notez que le mot « et » est traduit par \vee . Une autre traduction équivalente est : $\forall x (C(x) \Rightarrow D(x)) \wedge \forall x (O(x) \Rightarrow D(x))$.
2. Toby est un chien qui aime les enfants : $C(t) \wedge \forall x (E(x) \Rightarrow A(t, x))$.
3. Les oiseaux n'aiment pas les chiens : $\forall x \forall y (O(x) \wedge C(y) \Rightarrow \neg A(x, y))$.
Une autre traduction équivalente est : $\neg \exists x \exists y (O(x) \wedge C(y) \wedge A(x, y))$.
4. Serge aime tous les animaux domestiques sauf les chiens : $\forall x (D(x) \Rightarrow (A(s, x) \Leftrightarrow \neg C(x)))$.
5. Tous les enfants n'ont pas peur des chiens : Il y a plusieurs lectures de la phrase, la négation porte sur l'ensemble de la phrase, il n'est pas vrai que tous les enfants ont peur des chiens. Mais l'article « des » a plusieurs traductions :
— des chiens = d'un chien
— des chiens = de tous les chiens
En rétablissant la forme négative soit il y a un enfant qui n'a peur d'aucun chien, soit il y a un enfant et un chien tels que l'enfant n'a pas peur du chien
(a) il y a un enfant qui n'ait peur d'aucun chien : $\exists x (E(x) \wedge \forall y (C(y) \Rightarrow \neg P(x, y)))$.
Une autre traduction équivalente est : $\neg \forall x (E(x) \Rightarrow \exists y (C(y) \wedge P(x, y)))$
(b) il y a un enfant et un chien tels que l'enfant n'a pas peur du chien : $\exists x \exists y (E(x) \wedge C(y) \wedge \neg P(x, y))$.
Une autre traduction équivalente est : $\neg \forall x \forall y (E(x) \wedge C(y) \Rightarrow P(x, y))$.
6. Certains chiens aiment les enfants : $\exists x (C(x) \wedge \forall y (E(y) \Rightarrow A(x, y)))$.
7. Certains chiens aiment les enfants et réciproquement : Le mot réciproquement est ambigu, dans toute phrase qui n'est pas une implication. Nous proposons plusieurs formulations en français et en formules

- (a) Certains chiens aiment les enfants et certains enfants aiment les chiens :

$$\exists x(C(x) \wedge \forall y(E(y) \Rightarrow A(x, y))) \wedge \exists x(E(x) \wedge \forall y(C(y) \Rightarrow A(x, y))).$$

- (b) Certains chiens aiment les enfants et les enfants aiment certains chiens :

$$\exists x(C(x) \wedge \forall y(E(y) \Rightarrow A(x, y))) \wedge \forall x(E(x) \Rightarrow \exists y(C(y) \wedge A(x, y))).$$

- (c) Certains chiens aiment les enfants et certains chiens sont aimés des enfants :

$$\exists x(C(x) \wedge \forall y(E(y) \Rightarrow A(x, y))) \wedge \exists x(C(x) \wedge \forall y(E(y) \Rightarrow A(x, y))).$$

- (d) Certains chiens aiment les enfants et ces chiens sont aimés des enfants :

$$\exists x(C(x) \wedge \forall y(E(y) \Rightarrow A(x, y) \wedge A(y, x))).$$

8. Les enfants aiment certains chiens : $\forall x(E(x) \Rightarrow \exists y(C(y) \wedge A(x, y)))$.

En conclusion, quand nous faisons des sciences, la formalisation est utile car elle permet de lever les ambiguïtés qui sont fréquentes dans toutes les langues naturelles.

Exercice 66, page 105

Soit I l'interprétation de domaine $\{0, 1\}$ telle que $P_I = \{0\}$, $Q_I = \{1\}$. Dans cette interprétation, nous avons :

1. $\forall x P(x) = 0$, $\forall x (P(x) \vee Q(x)) = 1$.
2. Les formules $\forall x P(x) \vee \forall x Q(x)$ et $\forall x (P(x) \vee Q(x))$ ne sont pas équivalentes, car nous avons dans I : $\forall x P(x) \vee \forall x Q(x) = 0$ et $\forall x (P(x) \vee Q(x)) = 1$.
3. $\exists x P(x) = 1$, $\exists x (P(x) \wedge Q(x)) = 0$.
4. Les formules $\exists x P(x) \wedge \exists x Q(x)$ et $\exists x (P(x) \wedge Q(x))$ ne sont pas équivalentes, car nous avons dans I : $\exists x P(x) \wedge \exists x Q(x) = 1$ et $\exists x (P(x) \wedge Q(x)) = 0$.
5. $\forall x (P(x) \Rightarrow Q(x)) = 0$ et $\forall x P(x) \Rightarrow \forall x Q(x) = 1$.
6. Donc les deux formules ne sont pas équivalentes.

Exercice 67, page 105

Dans la colonne d'une interprétation, nous avons indiqué les valeurs des 6 formules dans l'interprétation.

Formules	I1	I2	I3	I4
$\forall x \exists y (y = x + 1)$	1	1	1	1
$\exists y \forall x (y = x + 1)$	1	0	0	1
$\forall x \exists y (y = x + 1) \Rightarrow \exists y \forall x (y = x + 1)$	1	0	0	1
$\forall x \exists y (x = y + 1)$	0	0	1	0
$\exists x \forall y (y = x + y)$	1	1	1	1
$\exists x (x \neq 0 \wedge x + x = x)$	1	0	0	1

Exercice 68, page 105

Soit I l'interprétation de domaine $\{0, 1, 2\}$ telle que $P_I = \{0, 1\}$, $Q_I = \{1, 2\}$, $R_I = \{\}$. Dans cette interprétation, nous avons :

1. $\exists x R(x) = 0$.
2. $\forall x (P(x) \vee Q(x)) = 1$.
3. $\forall x (P(x) \Rightarrow Q(x)) = 0$.
4. $\forall x (R(x) \Rightarrow Q(x)) = 1$.
5. $\exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y)) = 0$.
6. $\exists x (P(x) \wedge Q(x) \wedge \forall y (P(y) \wedge Q(y) \Rightarrow x = y)) = 1$.

Exercice 69, page 106

formules	valeur dans I_1	valeur dans I_2
$f(a) = a$	1	0
$f(f(a)) = a$	0	0
$f(f(f(a))) = a$	1	1
$\exists x(f(x) = x)$	1	0
$\forall x(f(f(f(x)))) = x$	0	1
$\forall y \exists x(f(x) = y)$	0	1
$\forall x \forall y(f(x) = f(y) \Rightarrow x = y)$	0	1
$\neg \exists x \exists y(f(x) = f(y) \wedge x \neq y)$	0	1

Exercice 70, page 106

La signature est la suivante $\Sigma = \{Anatoli^{f0}, Boris^{f0}, Catarina^{f0}, Denka^{f0}, P^{r1}, Q^{r2}\}$. L'interprétation donnée dans l'énoncé se définit formellement comme l'interprétation I sur Σ de domaine $D = \{0, 1, 2, 3\}$ telle que :

- $Anatoli_I^{f0} = 0, Boris_I^{f0} = 1, Catarina_I^{f0} = 2, Denka_I^{f0} = 3,$
- $P_I^{r1} = \{1, 2\}$ et
- $Q_I^{r2} = \{(0, 2), (0, 3), (1, 2), (1, 3), (3, 1), (2, 3)\}.$

Formules	valeurs dans l'interprétation I
$\exists x(\neg P(x) \wedge \forall y \neg Q(y, x))$	1
$\forall x(P(x) \Rightarrow \exists y Q(y, x))$	1
$\neg \exists x \forall y(P(y) \Rightarrow Q(x, y))$	1
$\forall x(\exists y Q(x, y) \Rightarrow \exists z(Q(x, z) \wedge P(z)))$	0

Exercice 71, page 106

1. $\exists x P(x) \Rightarrow \exists x(P(x) \wedge Q(x))$, construisons la 1-expansion de cette formule :

$$P(0) \Rightarrow P(0).Q(0)$$

Cette expansion vaut 0 pour l'assignation $P(0) = 1, Q(0) = 0$, ce qui donne le contre-modèle I de domaine $\{0\}$ avec $P_I = \{0\}, Q_I = \{\}$.

2. $\forall x(P(x) \Rightarrow Q(x)) \Rightarrow \exists x Q(x)$, construisons la 1-expansion de cette formule :

$$(P(0) \Rightarrow Q(0)) \Rightarrow Q(0).$$

Cette expansion vaut 0 pour l'assignation $P(0) = 0, Q(0) = 0$, ce qui donne le contre-modèle I de domaine $\{0\}$ avec $P_I = \{\}, Q_I = \{\}$.

3. $\forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\exists x P(x) \Rightarrow \forall x Q(x))$, construisons la 2-expansion de cette formule :

$$(P(0) \Rightarrow Q(0)).(P(1) \Rightarrow Q(1)) \Rightarrow (P(0) + P(1) \Rightarrow Q(0).Q(1))$$

Cette expansion vaut 0 pour l'assignation $P(0) = 1, P(1) = 0, Q(0) = 1, Q(1) = 0$, ce qui donne le contre-modèle I de domaine $\{0, 1\}$ avec $P_I = \{0\}, Q_I = \{0\}$.

4. $(\exists x F(x) \Rightarrow \exists x G(x)) \Rightarrow \forall x(F(x) \Rightarrow G(x))$, construisons la 2-expansion de cette formule :

$$(F(0) + F(1) \Rightarrow G(0) + G(1)) \Rightarrow (F(0) \Rightarrow G(0)).(F(1) \Rightarrow G(1))$$

Cette expansion vaut 0 pour l'assignation $F(0) = 1, F(1) = 1, G(0) = 0, G(1) = 1$, ce qui donne le contre-modèle I de domaine $\{0, 1\}$ avec $F_I = \{0, 1\}, G_I = \{1\}$.

5. $\forall x \exists y R(x, y) \Rightarrow \exists x R(x, x)$, construisons la 2-expansion de cette formule :

$$(R(0, 0) + R(0, 1)) \cdot (R(1, 0) + R(1, 1)) \Rightarrow R(0, 0) + R(1, 1)$$

Cette expansion vaut 0 pour l'assignation $R(0, 0) = 0, R(0, 1) = 1, R(1, 0) = 1, R(1, 1) = 0$, ce qui donne le contre-modèle I de domaine $\{0, 1\}$ avec $R_I = \{(0, 1), (1, 0)\}$.

6. $\forall x \forall y (R(x, y) \Rightarrow R(y, x)) \Rightarrow \forall x R(x, x)$, construisons la 1-expansion de cette formule :

$$(R(0, 0) \Rightarrow R(0, 0)) \Rightarrow R(0, 0)$$

Cette expansion vaut 0 pour l'assignation $R(0, 0) = 0$, ce qui donne le contre-modèle I de domaine $\{0\}$ avec $R_I = \{\}$.

Exercice 72, page 106

Nous construisons une 2-expansion des hypothèses et de la conclusion.

— Hypothèses :

1. $P(0) + P(1)$.
2. $Q(0) + Q(1)$.
3. $(P(0) \cdot Q(0) \Rightarrow R(0)) \cdot (P(1) \cdot Q(1) \Rightarrow R(1))$.

— Conclusion : $R(0) + R(1)$.

L'assignation $P(0) = 1, P(1) = 0, Q(0) = 0, Q(1) = 1, R(0) = 0, R(1) = 0$ est modèle des expansions des hypothèses et n'est pas modèle de la conclusion. Revenons au premier ordre, l'interprétation I de domaine $\{0, 1\}$ avec $P_I = \{0\}, Q_I = \{1\}, R_I = \{\}$ est modèle des hypothèses et contre-modèle de la conclusion. Donc le raisonnement est incorrect.

Exercice 73, page 107

1. $\forall x \exists y (x = y) \Rightarrow \exists y \forall x (y = x)$, nous construisons la 2-expansion de cette formule :

$$((0 = 0) + (0 = 1)) \cdot ((1 = 0) + (1 = 1)) \Rightarrow (0 = 0) \cdot (0 = 1) + (1 = 0) \cdot (1 = 1)$$

En tenant compte du sens de l'égalité, l'expansion vaut 0, donc toute interprétation avec au moins 2 éléments est contre-modèle de la formule.

2. $F(a) \wedge (a \neq b) \Rightarrow \neg F(b)$. Remplaçons a et b par 0 et 1, nous obtenons la formule $F(0) \wedge (0 \neq 1) \Rightarrow \neg F(1)$. En tenant compte du sens de l'égalité, elle se simplifie en $F(0) \Rightarrow \neg F(1)$. L'assignation $F(0) = 1, F(1) = 1$ lui donne la valeur 0. Donc l'interprétation I de domaine $\{0, 1\}$ telle que $a_I = 0, b_I = 1, F_I = \{0, 1\}$ est contre-modèle de la formule donnée.
3. $\exists x \exists y (F(x) \wedge F(y) \wedge x \neq y) \Rightarrow \forall x F(x)$. La formule « raconte » que s'il a deux éléments (au moins) qui vérifient la propriété F , alors F est toujours vérifiée. Avec cette lecture de la formule, la solution saute aux yeux : Soit I de domaine $\{0, 1, 2\}$ avec $F_I = \{0, 1\}$. L'interprétation I est un contre-modèle de la formule donnée. Par la méthode des expansions, c'est plus long, puisqu'il faudrait construire une 3-expansion, nous laissons au lecteur le soin de calculer cette 3-expansion.
4. $\forall x \forall y (F(x, y) \Rightarrow x = y) \Rightarrow \exists x F(x, x)$, nous construisons la 1-expansion : $(F(0, 0) \Rightarrow 0 = 0) \Rightarrow F(0, 0)$. En tenant compte du sens de l'égalité, cette formule devient $(F(0, 0) \Rightarrow 1) \Rightarrow F(0, 0)$. Pour l'assignation $F(0, 0) = 0$, elle vaut 0. Donc l'interprétation I de domaine $\{0\}$ avec $F_I = \{\}$ est contre-modèle de la formule.

Exercice 74, page 107

1. $\forall y \exists x (f(x) = y)$, la 2-expansion est :

$$((f(0) = 0) + (f(1) = 0)) \cdot ((f(0) = 1) + (f(1) = 1))$$

Il suffit de rendre une des clauses fausses, on choisit arbitrairement le premier. Il n'y a alors pas le choix : $f(0) = f(1) = 1$.

2. $\forall x \forall y (f(x) = f(y) \Rightarrow x = y)$, la 2-expansion est :

$$(f(0) = f(0) \Rightarrow 0 = 0) \cdot (f(0) = f(1) \Rightarrow 0 = 1) \cdot (f(1) = f(0) \Rightarrow 1 = 0) \cdot (f(1) = f(1) \Rightarrow 1 = 1)$$

qui se simplifie en $\neg(f(0) = f(1)) \cdot \neg(f(1) = f(0))$. On choisit arbitrairement $f(0) = 0$ et alors $f(1) = 1$ nous donne le contre-modèle.

3. $\exists x \forall y (f(y) = x)$, la 2-expansion est :

$$(f(0) = 0) \cdot (f(1) = 0) + (f(0) = 1) \cdot (f(1) = 1)$$

On choisit arbitrairement $f(0) = 0$, ce qui rend le second monôme faux. Pour rendre le premier faux également, on doit alors prendre $f(1) = 1$.

4. $\forall x (P(x) \Rightarrow P(f(x)))$, la 2-expansion est :

$$(P(0) \Rightarrow P(f(0))) \cdot (P(1) \Rightarrow P(f(1)))$$

Il faut qu'une des deux implications soit fausse, on choisit pour cela $P(0) = 1$ et $P(f(0)) = 0$. Ceci impose que $f(0) \neq 0$ et donc $f(0) = 1$, d'où $P(1) = 0$. La valeur de $f(1)$ est laissée libre.

Exercice 75, page 107

Les preuves sont une suite de formules « élémentairement » équivalentes accompagnées des équivalences utilisées.

1. $\neg \forall x \exists y P(x, y) \equiv \exists y \forall x \neg P(x, y)$.

- (1) $\neg \forall x \exists y P(x, y)$
- (2) $\equiv \exists x \neg \exists y P(x, y)$ relation entre les quantificateurs
- (3) $\equiv \exists x \forall y \neg P(x, y)$ relation entre les quantificateurs
- (4) $\equiv \exists y \forall x \neg P(x, y)$ par changement de variables liées

2. $\exists x (P(x) \Rightarrow Q(x)) \equiv \forall x P(x) \Rightarrow \exists x Q(x)$.

- (1) $\exists x (P(x) \Rightarrow Q(x))$
- (2) $\equiv \exists x (\neg P(x) \vee Q(x))$ élimination de \Rightarrow
- (3) $\equiv \exists x \neg P(x) \vee \exists x Q(x)$ distributivité de \exists sur \vee
- (4) $\equiv \neg \forall x P(x) \vee \exists x Q(x)$ équivalence $\exists \neg$ et $\neg \forall$
- (5) $\equiv \forall x P(x) \Rightarrow \exists x Q(x)$ ajout de \Rightarrow

3. Nous transformons une formule en l'autre par équivalences successives.

- (1) $\neg(\exists x (M(x) \wedge (\exists y (A(x, y) \wedge C(y))))$
- (2) $\equiv \forall x \neg(M(x) \wedge (\exists y (A(x, y) \wedge C(y))))$ $\neg \exists = \forall \neg$
- (3) $\equiv \forall x (\neg M(x) \vee \neg(\exists y (A(x, y) \wedge C(y))))$ identité de De Morgan
- (4) $\equiv \forall x (\neg M(x) \vee \forall y (\neg(A(x, y) \wedge C(y))))$ $\neg \exists = \forall \neg$
- (5) $\equiv \forall x \forall y (\neg M(x) \vee (\neg(A(x, y) \wedge C(y))))$ déplacement \forall
- (6) $\equiv \forall x \forall y (\neg M(x) \vee (\neg A(x, y) \vee \neg C(y)))$ identité de De Morgan
- (7) $\equiv \forall x \forall y ((\neg M(x) \vee \neg A(x, y)) \vee \neg C(y))$ associativité de \vee
- (8) $\equiv \forall x \forall y (\neg(M(x) \wedge A(x, y)) \vee \neg C(y))$ identité de De Morgan
- (9) $\equiv \forall x \forall y ((M(x) \wedge A(x, y)) \Rightarrow \neg C(y))$ identité propositionnelle

Exercice 76, page 107

En utilisant les propriétés 1 page 98 et 2 page 98 du lemme 4.4.1 page 98

- $\neg \exists x A \equiv \forall x \neg A$
 $\neg \exists x A$
 $\equiv \neg \exists x \neg \neg A$ identité de la double négation
 $\equiv \forall x \neg A$ par l'identité 2 page 98

- $\exists x A \equiv \neg \forall x \neg A$
- $\exists x A$
- $\equiv \exists x \neg \neg A$ identité de la double négation
- $\equiv \neg \forall x \neg A$ par l'identité 1 page 98

Exercice 77, page 107

Soient I l'interprétation de domaine $\{0, 1\}$ avec $P_I = \{0\}$, $Q_I = \{0, 1\}$ et e un état des variables avec $e(x) = 0$. Il est clair que l'assignation (I, e) donne la valeur 0 à la première formule et 1 à la deuxième.

Exercice 78, page 133

Voici cinq éléments du domaine de Herbrand : $a, f(a), g(a, a), f(f(a)), g(a, f(a))$. Le domaine de Herbrand D est défini inductivement par les trois seules règles suivantes :

- $a \in D$.
- Si $d \in D$ alors $f(d) \in D$.
- Si $d, e \in D$ alors $g(d, e) \in D$.

Exercice 79, page 133

1. Les signatures des ensembles de formules sont :

- $\Sigma_{\Gamma_1} = \{a^{f^0}, b^{f^0}, c^{f^0}, P^{r1}, Q^{r1}, R^{r1}\}$.
- $\Sigma_{\Gamma_2} = \{a^{f^0}, P^{r1}, Q^{r1}, f^{f1}\}$.
- $\Sigma_{\Gamma_3} = \{a^{f^0}, P^{r1}, Q^{r1}, R^{r1}, f^{f1}\}$.

Les domaines de Herbrand des ensembles de formules sont :

- $D_{\Sigma_{\Gamma_1}} = \{a, b, c\}$.
- $D_{\Sigma_{\Gamma_2}} = \{f^n(a), n \in \mathbb{N}\}$.
- $D_{\Sigma_{\Gamma_3}} = \{f^n(a), n \in \mathbb{N}\}$.

Les bases de Herbrand des ensembles de formules sont :

- $B_{\Sigma_{\Gamma_1}} = \{P(a), P(b), P(c), Q(a), Q(b), Q(c), R(a), R(b), R(c)\}$.
- $B_{\Sigma_{\Gamma_2}} = \{P(f^n(a)), n \in \mathbb{N}\} \cup \{Q(f^n(a)), n \in \mathbb{N}\}$.
- $B_{\Sigma_{\Gamma_3}} = \{P(f^n(a)), n \in \mathbb{N}\} \cup \{Q(f^n(a)), n \in \mathbb{N}\} \cup \{R(f^n(a)), n \in \mathbb{N}\}$.

2. Nous regardons les fermetures universelles de ces ensembles de formules.

- $\forall(\Gamma_1)$ a un modèle : Soit $E = \{P(b), Q(c), R(a)\}$. $H_{\Sigma_{\Gamma_1}, E}$ est une interprétation de Herbrand qui est modèle de l'ensemble des instances fermées de Γ_1 c'est-à-dire modèle de :

$$\{P(a) \vee Q(a) \vee R(a), P(b) \vee Q(b) \vee R(b), P(c) \vee Q(c) \vee R(c), \neg P(a), \neg Q(b), \neg R(c)\}.$$

- $\forall(\Gamma_2)$ est contradictoire car l'ensemble fini d'instances fermées suivant est insatisfaisable :

$$\{P(f(a)), \neg Q(f(a)), \neg P(f(a)) \vee Q(f(a))\}.$$

- $\forall(\Gamma_3)$ est contradictoire car l'ensemble fini d'instances fermées suivant est insatisfaisable :

$$\{P(f(f(f(a)))) , Q(f(a)), \neg R(f(f(a))), \neg P(f(f(f(a)))) \vee \neg Q(f(a)) \vee R(f(f(a)))\}.$$

Exercice 80, page 133

Soit I l'interprétation ayant comme domaine D l'ensemble des réels de l'intervalle fermé $[0, 1]$ avec $a_I = 0$, $b_I = 1$ et $f_I(x, y) = \frac{x+y}{2}$. Cette interprétation est clairement un modèle de $\forall(\Delta)$. D'après le théorème 5.1.16 page 110, puisque $\forall(\Delta)$ a un modèle, il a aussi un modèle sur le domaine de Herbrand D engendré par a, b, f .

Exercice 81, page 133

$$1. \neg(\exists x P(x) \vee \exists x Q(x) \Rightarrow \exists x(P(x) \vee Q(x))).$$

Forme normale : $(\exists x P(x) \vee \exists x Q(x)) \wedge \forall x(\neg P(x) \wedge \neg Q(x))$

Forme propre : $(\exists x P(x) \vee \exists y Q(y)) \wedge \forall z(\neg P(z) \wedge \neg Q(z))$

Elimination de \exists : $(P(a) \vee Q(b)) \wedge \forall z(\neg P(z) \wedge \neg Q(z))$

Forme de Skolem : $(P(a) \vee Q(b)) \wedge (\neg P(z) \wedge \neg Q(z))$

Forme clause : $\{P(a) \vee Q(b), \neg P(z), \neg Q(z)\}$

$$2. \neg(\forall x \forall y \forall z (e(x, y) \wedge e(y, z) \Rightarrow \neg e(x, z)) \Rightarrow \neg \exists x \forall y e(x, y)).$$

Forme normale : $(\forall x \forall y \forall z (\neg e(x, y) \vee \neg e(y, z) \vee \neg e(x, z))) \wedge \exists x \forall y e(x, y)$

Forme propre : $(\forall x \forall y \forall z (\neg e(x, y) \vee \neg e(y, z) \vee \neg e(x, z))) \wedge \exists u \forall v e(u, v)$

Elimination de \exists : $(\forall x \forall y \forall z (\neg e(x, y) \vee \neg e(y, z) \vee \neg e(x, z))) \wedge \forall v e(a, v)$

Forme de Skolem : $(\neg e(x, y) \vee \neg e(y, z) \vee \neg e(x, z)) \wedge e(a, v)$

Forme clause : $\{\neg e(x, y) \vee \neg e(y, z) \vee \neg e(x, z), e(a, v)\}$

$$3. \neg(\neg \forall x P(x) \vee \neg \forall x Q(x) \Rightarrow \neg(\forall x P(x) \wedge \forall x Q(x))).$$

Forme normale : $(\exists x \neg P(x) \vee \exists x \neg Q(x)) \wedge \forall x P(x) \wedge \forall x Q(x)$

Forme propre : $(\exists x \neg P(x) \vee \exists y \neg Q(y)) \wedge \forall u P(u) \wedge \forall v Q(v)$

Elimination de \exists : $(\neg P(a) \vee \neg Q(b)) \wedge \forall u P(u) \wedge \forall v Q(v)$

Forme de Skolem : $(\neg P(a) \vee \neg Q(b)) \wedge P(u) \wedge Q(v)$

Forme clause : $\{\neg P(a) \vee \neg Q(b), P(u), Q(v)\}$

$$4. \forall x((\exists y P(x, y) \Rightarrow \exists x Q(x)) \wedge \exists y P(x, y) \wedge \neg \exists x Q(x)).$$

Forme normale : $\forall x((\forall y \neg P(x, y) \vee \exists x Q(x)) \wedge \exists y P(x, y) \wedge \forall x \neg Q(x))$

Forme propre : $\forall x((\forall y \neg P(x, y) \vee \exists z Q(z)) \wedge \exists r P(x, r) \wedge \forall s \neg Q(s))$

Elimination de \exists : $\forall x((\forall y \neg P(x, y) \vee Q(a)) \wedge P(x, f(x)) \wedge \forall s \neg Q(s))$

Forme de Skolem : $(\neg P(x, y) \vee Q(a)) \wedge P(x, f(x)) \wedge \neg Q(s)$

Forme clause : $\{\neg P(x, y) \vee Q(a), P(x, f(x)), \neg Q(s)\}$

$$5. \neg(\exists x \forall y \forall z ((P(y) \Rightarrow Q(z)) \Rightarrow (P(x) \Rightarrow Q(x))))$$

Forme normale : $\forall x \exists y \exists z ((\neg P(y) \vee Q(z)) \wedge P(x) \wedge \neg Q(x))$

Forme propre : idem

Elimination de \exists : $\forall x((\neg P(f(x)) \vee Q(g(x))) \wedge P(x) \wedge \neg Q(x))$

Forme de Skolem : $(\neg P(f(x)) \vee Q(g(x))) \wedge P(x) \wedge \neg Q(x)$

Forme clause : $\{\neg P(f(x)) \vee Q(g(x)), P(x), \neg Q(x)\}$

Exercice 82, page 133

Posons $h(g(x), f(a, y), z) = h(y, z, f(u, x))$.

— Par décomposition, nous avons : $g(x) = y, f(a, y) = z, z = f(u, x)$.

— Par orientation, $y = g(x), z = f(a, y), z = f(u, x)$.

— Par élimination de y via la première équation, nous obtenons : $y := g(x), z = f(a, g(x)), z = f(u, x)$.

— Par élimination de z via la deuxième équation, nous obtenons : $y := g(x), z := f(a, g(x)), f(a, g(x)) = f(u, x)$.

— Par décomposition, nous obtenons : $y := g(x), z := f(a, g(x)), a = u, g(x) = x$.

— Par orientation de la dernière équation nous obtenons : $y := g(x), z := f(a, g(x)), a = u, x = g(x)$.

— L'équation $x = g(x)$ conduit à un échec : les termes ne sont pas unifiables.

Notez que nous pouvons appliquer les règles de l'algorithme d'unification dans un ordre arbitraire, ainsi sur cet exemple, nous avons appliqué des règles à la dernière équation, en ignorant l'avant dernière équation. Néanmoins nous conseillons de traiter systématiquement les équations de gauche à droite.

Posons $h(g(x), f(a, y), z) = h(y, z, f(u, g(x)))$.

- Par décomposition, $g(x) = y, f(a, y) = z, z = f(u, g(x))$.
- Par orientation, nous avons : $y = g(x), z = f(a, y), z = f(u, g(x))$.
- Par élimination de y via la première équation, nous obtenons : $y := g(x), z = f(a, g(x)), z = f(u, g(x))$.
- Par élimination de z via la deuxième équation, nous obtenons : $y := g(x), z := f(a, g(x)), f(a, g(x)) = f(u, g(x))$.
- Par décomposition, nous obtenons : $y := g(x), z := f(a, g(x)), a = u, g(x) = g(x)$.
- Par orientation, nous obtenons : $y := g(x), z := f(a, g(x)), u = a, g(x) = g(x)$.
- Par élimination, nous obtenons : $y := g(x), z := f(a, g(x)), u := a, g(x) = g(x)$.
- La dernière équation peut être supprimée. Ainsi, l'unificateur le plus général est : $y := g(x), z := f(a, g(x)), u := a$.

Exercice 83, page 133

Nous donnons directement les unificateurs des termes demandés.

- $\text{pair}(a, \text{crypt}(z, k))$ et $\text{pair}(x, y) : \sigma = \{x := a, y := \text{crypt}(z, b)\}$
- $\text{pair}(\text{crypt}(x, b), \text{crypt}(y, b))$ et $\text{pair}(\text{crypt}(a, b), z) : \sigma = \{x := a, z := \text{crypt}(y, b)\}$
- $\text{crypt}(\text{pair}(z, a), x)$ et $\text{crypt}(\text{pair}(y, \text{crypt}(x, b)), b) : \text{pas de solution}$
- $\text{crypt}(\text{pair}(a, z), x)$ et $\text{crypt}(\text{pair}(y, \text{crypt}(x, b)), b) : \sigma = \{y := a, z := \text{crypt}(b, b), x := b\}$
- $f(x, y, g(a, a))$ et $f(g(y, y), z, z) : \sigma = \{x := g(g(a, a), g(a, a)), y := z := g(a, a)\}$
- $f(x, y, a)$ et $f(y, g(z, z), x) : \text{pas de solution}$

Exercice 84, page 133

- Par décomposition nous obtenons : $g(y) = u, y = z$.
- Par orientation de la première équation, nous avons $u = g(y), y = z$.
- Par élimination de u , nous avons $u := g(y), y = z$.
- Par élimination de y , nous obtenons la solution : $u := g(z), y := z$.

Mais l'unificateur $u := g(y), z := y$ est aussi une solution la plus générale, obtenue en « orientant » $y = z$ en $z = y$. Dès que le système d'équations comporte une égalité entre deux variables, l'orientation de cette égalité est arbitraire et il y a plusieurs solutions « les plus générales ». Cependant, notez que bien qu'elle soit valide, cette réorientation n'est pas permise par l'algorithme afin de garantir sa terminaison.

Exercice 85, page 133

La négation de la formule donne (à un renommage près des variables) les clauses :

- $\neg P(x) \vee Q(x)$.
- $P(y)$.
- $\neg Q(a)$.

En instanciant dans chaque clause x et y par la constante a , nous obtenons les clauses :

- $\neg P(a) \vee Q(a)$
- $P(a)$
- $\neg Q(a)$

Par résolution propositionnelle, nous en déduisons \perp :

- | | | |
|-----|-----------------------|--------------------------|
| (1) | $\neg P(a) \vee Q(a)$ | Hyp |
| (2) | $P(a)$ | Hyp |
| (3) | $\neg Q(a)$ | Hyp |
| (4) | $Q(a)$ | par résolution de 1 et 2 |
| (5) | \perp | par résolution de 3 et 4 |

Exercice 86, page 134

Transformation en clauses : Nous obtenons les clauses suivantes à partir des hypothèses et de la conclusion.

- H_1 est transformée en $\neg P(y) \vee P(x)$.
- H_2 est transformée en $P(x) \vee Q(x)$.
- $\neg C$ est transformée en les deux clauses $\neg Q(a)$ et $\neg P(b)$.

Instanciation : Nous obtenons les instances contradictoires suivantes :

- $\neg P(a) \vee P(b)$
- $P(a) \vee Q(a)$
- $\neg Q(a)$
- $\neg P(b)$

Par résolution propositionnelle (et sans répéter les hypothèses) nous avons la preuve :

(1)	$\neg P(a) \vee P(b)$	Hyp
(2)	$P(a) \vee Q(a)$	Hyp
(3)	$\neg Q(a)$	Hyp
(4)	$\neg P(b)$	Hyp
(5)	$\neg P(a)$	Résolvant 1,4
(6)	$Q(a)$	Résolvant 2,5
(7)	\perp	Résolvant 3,6

Résolution binaire :

numéro	conclusion	justification	unificateur
(1)	$\neg P(y) \vee P(x)$	Hyp	
(2)	$P(x) \vee Q(x)$	Hyp	
(3)	$\neg Q(a)$	Hyp	
(4)	$\neg P(b)$	Hyp	
(5)	$\neg P(y)$	RB 1(2),4	$x := b$
(6)	$Q(x)$	RB 2(1),5	$y := x$
(7)	\perp	RB 3,6	$x := a$

Notons qu'à l'étape (5), nous avons précisé par la notation 1(2), que c'est le deuxième littéral de la clause 1, qui est éliminé.

Exercice 87, page 134

numéro	conclusion	justification	unificateur
(1)	$\neg S(y, z) \vee \neg S(a, b)$	Hyp	
(2)	$\neg S(a, b)$	Fact 1	$y := a, z := b$
(3)	$Q(a, a) \vee R(x, x, b) \vee S(a, b)$	Hyp	
(4)	$Q(a, a) \vee R(x, x, b)$	RB 2,3	
(5)	$S(a, z) \vee \neg R(x, x, b)$	Hyp	
(6)	$\neg R(x, x, b)$	RB 2,5	$z := b$
(7)	$\neg R(y, y, b)$	Copie 6	$x := y$
(8)	$Q(a, a)$	RB 4,7	$y := x$
(9)	$P(f(x)) \vee \neg Q(y, a)$	Hyp	
(10)	$P(f(x))$	RB 8,9	$y := a$
(11)	$\neg P(f(c)) \vee R(x, a, b)$	Hyp	
(12)	$\neg P(f(c)) \vee R(z, a, b)$	Copie 11	$x := z$
(13)	$R(z, a, b)$	RB 10,12	$x := c$
(14)	\perp	RB 6,13	$x := a, z := a$

numéro	conclusion	justification	unificateur
(1)	$\neg P(y) \vee Q(y, x)$	Hyp	
(2)	$\neg P(y_0) \vee Q(y_0, x_0)$	Copie 1	$x := x_0, y := y_0$
(3)	$P(x)$	Hyp	
(4)	$Q(y_0, x_0)$	RB 2,3	$x := y_0$
(5)	$\neg Q(x, a) \vee \neg Q(b, y) \vee \neg Q(b, a) \vee \neg P(f(y))$	Hyp	
(6)	$\neg Q(b, a) \vee \neg P(f(a))$	Fact 5	$x := b, y := a$
(7)	$\neg P(f(a))$	RB 4,6	$y_0 := b, x_0 := a$
(8)	\perp	RB 3,7	$x := f(a)$

numéro	conclusion	justification	unificateur
(1)	$y = b$	Hyp	
(2)	$y_0 = b$	Copie 1	$y := y_0$
(3)	$\neg(a + b < x) \vee \neg(y = b)$	Hyp	
(4)	$\neg(a + b < x)$	RB 2,3	$y_0 := y$
(5)	$\neg(a + b < x_0)$	Copie 4	$x := x_0$
(6)	$\neg(b < x + b) \vee (a + b < a)$	Hyp	
(7)	$\neg(b < x + b)$	RB 5,6	$x_0 := a$
(8)	$b < z$	Hyp	
(9)	\perp	RB 7,8	$z := x + b$

Exercice 88, page 134

- Soit $\Gamma_1 = \{P(x, f(x, b), u), \neg P(g(a), z, h(z))\}$ l'ensemble des *deux* clauses unitaires. L'équation $P(x, f(x, b), u) = P(g(a), z, h(z))$ a une solution : $x = g(a), z = f(g(a), b), u = h(f(g(a), b))$. Donc par résolution binaire des deux clauses, nous obtenons la clause vide. Par suite, $\forall(\Gamma_1)$ est insatisfaisable.
- Soit $\Gamma_2 = \{P(x, f(x, b), u), \neg P(g(z), z, h(z))\}$ l'ensemble des *deux* clauses unitaires. L'équation $P(x, f(x, b), u) = P(g(z), z, h(z))$ n'a pas de solution. Car en appliquant l'algorithme d'unification, nous en déduisons l'équation sans solution $z = f(g(z), b)$. Donc, les deux clauses n'ont aucun résolvant : par suite d'après la complétude du calcul par résolution, $\forall(\Gamma)$ est satisfaisable.

Exercice 89, page 134

Forme de Skolem $(\neg \text{homme}(x) \vee \text{mortel}(x)) \wedge \text{homme}(\text{Socrate}) \wedge \neg \text{mortel}(\text{Socrate})$.

Forme clauseale

- $\neg \text{homme}(x) \vee \text{mortel}(x)$
- $\text{homme}(\text{socrate})$
- $\neg \text{mortel}(\text{socrate})$

Instanciation

Nous obtenons les clauses contradictoires suivantes :

- $\neg \text{homme}(\text{socrate}) \vee \text{mortel}(\text{socrate})$
- $\text{homme}(\text{socrate})$
- $\neg \text{mortel}(\text{socrate})$

Nous montrons cette contradiction par la preuve en résolution propositionnelle ci-dessous.

numéro	conclusion	justification
(1)	$\neg \text{homme}(\text{socrate}) \vee \text{mortel}(\text{socrate})$	Hyp
(2)	$\text{homme}(\text{socrate})$	Hyp
(3)	$\text{mortel}(\text{socrate})$	Résolvant 1,2
(4)	$\neg \text{mortel}(\text{socrate})$	Hyp
(5)	\perp	Résolvant 3,4

Preuve par résolution

numéro	conclusion	justification	unificateur
(1)	$\neg \text{homme}(x) \vee \text{mortel}(x)$	Hyp	
(2)	$\text{homme}(\text{socrate})$	Hyp	
(3)	$\text{mortel}(\text{socrate})$	RB 1,2	$x := \text{socrate}$
(4)	$\neg \text{mortel}(\text{socrate})$	Hyp	
(5)	\perp	RB 3,4	

Exercice 90, page 135

Transformation en clauses : Nous obtenons que

- A_1 donne les deux clauses $P(a)$ et $\neg R(v) \vee Q(a, v)$.
- A_2 donne la clause $\neg P(u) \vee \neg S(v) \vee \neg Q(u, v)$.
- A_3 donne les deux clauses $R(b)$ et $S(b)$.

Instanciation : Nous obtenons les instances contradictoires suivantes :

- $P(a)$.
- $\neg R(b) \vee Q(a, b)$.
- $\neg P(a) \vee \neg S(b) \vee \neg Q(a, b)$.
- $R(b)$.
- $S(b)$.

Nous montrons cette contradiction par résolution propositionnelle :

numéro	conclusion	justification
(1)	$P(a)$	Hyp
(2)	$\neg R(b) \vee Q(a, b)$	Hyp
(3)	$\neg P(a) \vee \neg S(b) \vee \neg Q(a, b)$	Hyp
(4)	$R(b)$	Hyp
(5)	$S(b)$	Hyp
(6)	$\neg S(b) \vee \neg Q(a, b)$	Résolvant 1,3
(7)	$\neg Q(a, b)$	Résolvant 5,6
(8)	$\neg R(b)$	Résolvant 2,7
(9)	\perp	Résolvant 4,8

Preuve par résolution :

numéro	conclusion	justification	unificateur
(1)	$P(a)$	Hyp	
(2)	$\neg R(v) \vee Q(a, v)$	Hyp	
(3)	$\neg P(u) \vee \neg S(v) \vee \neg Q(u, v)$	Hyp	
(4)	$R(b)$	Hyp	
(5)	$S(b)$	Hyp	
(6)	$\neg S(v) \vee \neg Q(a, v)$	RB 1, 3	$u := a$
(7)	$\neg Q(a, b)$	RB 5, 6	$v := b$
(8)	$\neg R(b)$	RB 2, 7	$v := b$
(9)	\perp	RB 4, 8	

Exercice 91, page 135

Soient Γ un ensemble de clauses et C une clause. Soit P une preuve de C par factorisation, copie et résolution binaire dans l'environnement Γ .

Cas de base : Supposons que P contienne une seule ligne (C). Par définition, $C \in \Gamma$, donc $\Gamma \models C$ et $\forall(\Gamma) \models \forall(C)$.

Hypothèse de récurrence : Supposons que pour toute preuve P' plus courte que P , on a : $\Gamma \vdash_{1fcb} P' : D \Rightarrow \forall(\Gamma) \models \forall(D)$.

Pas de récurrence : Supposons que $|P| = k$ avec $k > 1$. Quatre cas sont possibles :

- $C \in \Gamma$. Comme pour le cas de base, nous pouvons conclure que $\forall(\Gamma) \models \forall(C)$.
- C est obtenue en factorisant une clause C' qui précède C dans P . D'après la propriété 5.4.4 page 124, $\forall(C') \models \forall(C)$. De plus, par hypothèse de récurrence, $\forall(\Gamma) \models \forall(C')$. Donc, de $\forall(C') \models \forall(C)$ et $\forall(\Gamma) \models \forall(C')$, nous déduisons $\forall(\Gamma) \models \forall(C)$.
- C est obtenue par copie d'une clause C' qui précède C dans P . D'après la propriété 5.4.9 page 125, $\forall(C') \equiv \forall(C)$. De plus, C' précède C dans P et par hypothèse de récurrence, $\forall(\Gamma) \models \forall(C')$. Puisque $\forall(C') \equiv \forall(C)$, nous pouvons remplacer C' par C et nous obtenons $\forall(\Gamma) \models \forall(C)$.
- C est un résolvant binaire de deux clauses C' et C'' , telles que (1) C' précède C dans P et (2) C'' précède C dans P . Nous obtenons comme précédemment, $\forall(\Gamma) \models \forall(C')$ et $\forall(\Gamma) \models \forall(C'')$. D'après la propriété 5.4.12 page 125, on a $\forall(C'), \forall(C'') \models \forall(C)$. Par suite, $\forall(\Gamma) \models \forall(C)$.

Ainsi dans tous les cas, nous obtenons, $\forall(\Gamma) \models \forall(C)$, ce qui prouve la propriété.

Exercice 92, page 135

Transformation en clauses

E_1 donne la clause :

- (a) $x = x$

E_2 donne la clause :

- (b) $\neg(x = y) \vee (y = x)$

E_3 donne la clause :

- (c) $\neg(x = y) \vee \neg(z = t) \vee \neg(x \in z) \vee (y \in t)$

$\neg C$ donne les trois clauses :

- (d) $a = b$
 (e) $c \in a \vee c \in b$
 (f) $\neg(c \in a) \vee \neg(c \in b)$

Notez que la conjonction des trois dernières clauses indique que les deux ensembles a, b sont égaux, et que c est élément d'un des ensembles et pas de l'autre : il n'est pas étonnant que nous en déduisions une contradiction.

Instanciation : Nous obtenons les instances contradictoires suivantes et nous montrons la contradiction :

numéro	conclusion	justification
(1)	$c = c$	instance de (a)
(2)	$\neg(a = b) \vee (b = a)$	instance de (b)
(3)	$\neg(c = c) \vee \neg(a = b) \vee \neg(c \in a) \vee (c \in b)$	instance de (c)
(4)	$\neg(c = c) \vee \neg(b = a) \vee \neg(c \in b) \vee (c \in a)$	instance de (c)
(5)	$a = b$	clause (d)
(6)	$c \in a \vee c \in b$	clause (e)
(7)	$\neg(c \in a) \vee \neg(c \in b)$	clause (f)
(8)	$\neg(a = b) \vee \neg(c \in a) \vee (c \in b)$	Résolvant 1, 3
(9)	$\neg(b = a) \vee \neg(c \in b) \vee (c \in a)$	Résolvant 1, 4
(10)	$\neg(c \in a) \vee (c \in b)$	Résolvant 5, 8
(11)	$b = a$	Résolvant 2, 5
(12)	$\neg(c \in b) \vee (c \in a)$	Résolvant 9, 11
(13)	$c \in b$	Résolvant 6, 10
(14)	$\neg(c \in b)$	Résolvant 7, 12
(15)	\perp	Résolvant 13, 14

Preuve par résolution : La factorisation n'est pas utilisée :

numéro	conclusion	justification	unificateur
(1)	$x = x$	Hyp	
(2)	$\neg(x = y) \vee (y = x)$	Hyp	
(3)	$\neg(x = y) \vee \neg(z = t) \vee \neg(x \in z) \vee (y \in t)$	Hyp	
(4)	$a = b$	Hyp	
(5)	$c \in a \vee c \in b$	Hyp	
(6)	$\neg(c \in a) \vee \neg(c \in b)$	Hyp	
(7)	$x_1 = x_1$	Copie 1	$x := x_1$
(8)	$\neg(z = t) \vee \neg(x_1 \in z) \vee (x_1 \in t)$	RB 3, 7	$x := x_1, y := x_1$
(9)	$\neg(x_1 \in a) \vee (x_1 \in b)$	RB 4, 8	$z := a, t := b$
(10)	$b = a$	RB 2, 4	$x := a, y := b$
(11)	$\neg(x_1 \in b) \vee (x_1 \in a)$	RB 8, 10	$z := b, t := a$
(12)	$c \in b$	RB 5, 9	$x_1 := c$
(13)	$\neg(c \in b)$	RB 6, 11	$x_1 := c$
(14)	\perp	Résolvant 12, 13	

Exercice 93, page 135

Transformation en clauses

H_1 donne la clause :

— $\neg R(u, v) \vee R(u, f(u))$

H_2 donne la clause :

— $R(u, g(u))$

H_3 donne la clause :

— $R(f(f(a)), a)$

$\neg C$ donne la clause :

— $\neg R(u, v) \vee \neg R(v, w) \vee \neg R(w, u)$

Instanciation : Nous découvrons les instanciations en même temps que nous donnons une preuve de la contradiction.

numéro	conclusion	justification
(1)	$\neg R(a, g(a)) \vee R(a, f(a))$	instance de (1)
(2)	$R(a, g(a))$	instance de (2)
(3)	$R(a, f(a))$	Résolvant 1, 2
(4)	$\neg R(f(a), g(f(a))) \vee R(f(a), f(f(a)))$	instance de (1)
(5)	$R(f(a), g(f(a)))$	instance de (2)
(6)	$R(f(a), f(f(a)))$	Résolvant 4, 5
(7)	$\neg R(a, f(a)) \vee \neg R(f(a), f(f(a))) \vee \neg R(f(f(a)), a)$	instance de (4)
(8)	$\neg R(f(a), f(f(a))) \vee \neg R(f(f(a)), a)$	Résolvant 3, 7
(9)	$\neg R(f(f(a)), a)$	Résolvant 6, 8
(10)	$R(f(f(a)), a)$	clause (3)
(11)	\perp	Résolvant 9, 10

Preuve par résolution : La factorisation n'est pas utilisée.

numéro	conclusion	justification	unificateur
(1)	$\neg R(u, v) \vee R(u, f(u))$	Hyp	
(2)	$R(u, g(u))$	Hyp	
(3)	$R(f(f(a)), a)$	Hyp	
(4)	$\neg R(u, v) \vee \neg R(v, w) \vee \neg R(w, u)$	Hyp	
(5)	$R(u_1, g(u_1))$	Copie 2	$u := u_1$
(6)	$R(u_1, f(u_1))$	RB 1(1), 5	$u := u_1; v := g(u_1)$
(7)	$\neg R(f(u_1), w) \vee \neg R(w, u_1)$	RB 4(1), 6	$u := u_1, v := f(u_1)$
(8)	$\neg R(f(u_2), w) \vee \neg R(w, u_2)$	Copie 7	$u_1 := u_2$
(9)	$\neg R(f(f(u_2)), u_2)$	RB 6, 8(2)	$u_1 := f(u_2), w := f(f(u_2))$
(10)	\perp	RB 3, 9	$u_2 := a$

Exercice 94, page 136

Nous obtenons les clauses suivantes :

1. $P(x, f(x))$
2. $A(x, z) \vee \neg P(x, y) \vee \neg P(y, z)$
3. $\neg A(a, y)$

Cet ensemble d'instances est contradictoire :

- $P(a, f(a))$, instance de 1
- $P(f(a), f(f(a)))$, instance de 1
- $A(a, f(f(a))) \vee \neg P(a, f(a)) \vee \neg P(f(a), f(f(a)))$, instance de 2
- $\neg A(a, f(f(a)))$, instance de 3

La preuve par résolution propositionnelle est :

(1)	$\neg A(a, f(f(a)))$	Hyp
(2)	$A(a, f(f(a))) \vee \neg P(a, f(a)) \vee \neg P(f(a), f(f(a)))$	Hyp
(3)	$\neg P(a, f(a)) \vee \neg P(f(a), f(f(a)))$	Résolvant 1,2
(4)	$P(a, f(a))$	Hyp
(5)	$\neg P(f(a), f(f(a)))$	Résolvant 3,4
(6)	$P(f(a), f(f(a)))$	Hyp
(7)	\perp	Résolvant 5,6

La preuve par factorisation, copie, résolution binaire est :

numéro	conclusion	justification	unificateur
(1)	$\neg A(a, y)$	Hyp	
(2)	$A(x_1, z_1) \vee \neg P(x_1, y_1) \vee \neg P(y_1, z_1)$	Copie de (2)	$x := x_1, z := z_1, y := y_1$
(3)	$\neg P(a, y_1) \vee \neg P(y_1, y)$	RB 1, 2	$x_1 := a, z_1 := y$
(4)	$P(x, f(x))$	Hyp	
(5)	$\neg P(f(a), y)$	RB 3, 4	$x := a, y_1 := f(a)$
(6)	\perp	RB 5, 4	$x := f(a), y := f(f(a))$

Exercice 95, page 136

Soit A la formule fermée $\exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_p B$ où B est sans quantificateur. Par skolemisation, la formule A est transformée en une forme de Skolem C dont les seules symboles de fonction sont n constantes utilisées pour éliminer les n quantificateurs existentiels. A est satisfaisable si et seulement si $\forall(C)$ est satisfaisable, et $\forall(C)$ est satisfaisable si et seulement si l'ensemble des instances de C sur le domaine de Herbrand est satisfaisable. Puisque ce domaine est fini (et comprend n^p éléments), l'ensemble des instances de C est fini (et comprend n^p éléments) donc la satisfaisabilité de cet ensemble est décidable, donc aussi celle de A .

Exercice 96, page 147

1. Le syllogisme fameux, tout homme est mortel, Socrate est un homme, donc est mortel, que nous formalisons par $\forall x(H(x) \Rightarrow M(x)) \wedge H(socrate) \Rightarrow M(socrate)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(H(x) \Rightarrow M(x)) \wedge H(socrate)$	
1	2	$H(socrate)$	$\wedge E 2$
1	3	$\forall x(H(x) \Rightarrow M(x))$	$\wedge E 1$
1	4	$H(socrate) \Rightarrow M(socrate)$	$\forall E 3, socrate$
1	5	$M(socrate)$	$\Rightarrow E 2,4$
	6	Donc $\forall x(H(x) \Rightarrow M(x)) \wedge H(socrate) \Rightarrow M(socrate)$	$\Rightarrow I 1,5$

2. $\forall xP(x) \Rightarrow \exists yP(y)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall xP(x)$	
1	2	$P(y)$	$\forall E 1, y$
1	3	$\exists yP(y)$	$\exists I 2, y$
	4	Donc $\forall xP(x) \Rightarrow \exists yP(y)$	$\Rightarrow I 1,3$

3. $\forall xP(x) \Rightarrow \exists xP(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall xP(x)$	
1	2	$P(x)$	$\forall E 1, x$
1	3	$\exists xP(x)$	$\exists I 2, x$
	4	Donc $\forall xP(x) \Rightarrow \exists xP(x)$	$\Rightarrow I 1,4$

4. $\forall x(P(x) \wedge Q(x)) \Rightarrow \forall xP(x) \wedge \forall xQ(x)$. Notez que l'exemple 6.1.2 donne la preuve de la réciproque.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(P(x) \wedge Q(x))$	
1	2	$P(x) \wedge Q(x)$	$\forall E 1, x$
1	3	$P(x)$	$\wedge E 1 2$
1	4	$Q(x)$	$\wedge E 2 2$
1	5	$\forall xP(x)$	$\forall I 3$
1	6	$\forall xQ(x)$	$\forall I 4$
1	7	$\forall xP(x) \wedge \forall xQ(x)$	$\wedge I 5,6$
	8	Donc $\forall x(P(x) \wedge Q(x)) \Rightarrow \forall xP(x) \wedge \forall xQ(x)$	$\Rightarrow I 1,7$

5. $\exists x(P(x) \vee Q(x)) \Rightarrow \exists xP(x) \vee \exists xQ(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x(P(x) \vee Q(x))$	
1,2	2	Supposons $P(x) \vee Q(x)$	
1,2,3	3	Supposons $P(x)$	
1,2,3	4	$\exists xP(x)$	$\exists I$ 3, x
1,2,3	5	$\exists xP(x) \vee \exists xQ(x)$	$\vee I$ 4
1,2	6	Donc $P(x) \Rightarrow \exists xP(x) \vee \exists xQ(x)$	$\Rightarrow I$ 3,5
1,2,7	7	Supposons $Q(x)$	
1,2,7	8	$\exists xQ(x)$	$\exists I$ 7, x
1,2,7	9	$\exists xP(x) \vee \exists xQ(x)$	$\vee I$ 8
1,2	10	Donc $Q(x) \Rightarrow \exists xP(x) \vee \exists xQ(x)$	$\Rightarrow I$ 7,9
1,2	11	$\exists xP(x) \vee \exists xQ(x)$	$\vee E$ 2,6,10
1	12	Donc $(P(x) \vee Q(x)) \Rightarrow \exists xP(x) \vee \exists xQ(x)$	$\Rightarrow I$ 2,11
1	13	$\exists xP(x) \vee \exists xQ(x)$	$\exists E$ 1,12
	14	Donc $\exists x(P(x) \vee Q(x)) \Rightarrow \exists xP(x) \vee \exists xQ(x)$	$\Rightarrow I$ 1,13

6. $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists xP(x) \Rightarrow \exists xQ(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists xP(x)$	
1	2	$\forall x(P(x) \Rightarrow Q(x))$	$\wedge E$ 1
1	3	$\exists xP(x)$	$\wedge E$ 2
1,4	4	Supposons $P(x)$	
1,4	5	$P(x) \Rightarrow Q(x)$	$\forall E$ 2, x
1,4	6	$Q(x)$	$\Rightarrow E$ 4,5
1,4	7	$\exists xQ(x)$	$\exists I$ 6, x
1	8	Donc $P(x) \Rightarrow \exists xQ(x)$	$\Rightarrow I$ 4,7
1	9	$\exists xQ(x)$	$\exists E$ 3,8
	10	Donc $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists xP(x) \Rightarrow \exists xQ(x)$	$\Rightarrow I$ 1,9

7. $\exists xP(x) \vee \exists xQ(x) \Rightarrow \exists x(P(x) \vee Q(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists xP(x) \vee \exists xQ(x)$	
1,2	2	Supposons $\exists xP(x)$	
1,2,3	3	Supposons $P(x)$	
1,2,3	4	$P(x) \vee Q(x)$	$\vee I$ 3
1,2,3	5	$\exists x(P(x) \vee Q(x))$	$\exists I$ 4, x
1,2	6	Donc $P(x) \Rightarrow \exists x(P(x) \vee Q(x))$	$\Rightarrow I$ 3,5
1,2	7	$\exists x(P(x) \vee Q(x))$	$\exists E$ 2,6
1	8	Donc $\exists xP(x) \Rightarrow \exists x(P(x) \vee Q(x))$	$\Rightarrow I$ 2,7
1,9	9	Supposons $\exists xQ(x)$	
1,9,10	10	Supposons $Q(x)$	
1,9,10	11	$P(x) \vee Q(x)$	$\vee I$ 10
1,9,10	12	$\exists x(P(x) \vee Q(x))$	$\exists I$ 11, x
1,9	13	Donc $Q(x) \Rightarrow \exists x(P(x) \vee Q(x))$	$\Rightarrow I$ 10,12
1,9	14	$\exists x(P(x) \vee Q(x))$	$\exists E$ 9,13
1	15	Donc $\exists xQ(x) \Rightarrow \exists x(P(x) \vee Q(x))$	$\Rightarrow I$ 9,14
1	16	$\exists x(P(x) \vee Q(x))$	$\vee E$ 1,8,15
	17	Donc $\exists xP(x) \vee \exists xQ(x) \Rightarrow \exists x(P(x) \vee Q(x))$	$\Rightarrow I$ 1,16

8. $\forall x(P(x) \Rightarrow Q(x)) \wedge \forall x(Q(x) \Rightarrow R(x)) \Rightarrow \forall x(P(x) \Rightarrow R(x)).$

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(P(x) \Rightarrow Q(x)) \wedge \forall x(Q(x) \Rightarrow R(x))$	
1	2	$\forall x(P(x) \Rightarrow Q(x))$	$\wedge E 1, 1$
1	3	$\forall x(Q(x) \Rightarrow R(x))$	$\wedge E 2, 1$
1	4	$P(x) \Rightarrow Q(x)$	$\forall E 2, x$
1	5	$Q(x) \Rightarrow R(x)$	$\forall E 3, x$
1,6	6	Supposons $P(x)$	
1,6	7	$Q(x)$	$\Rightarrow E 6,4$
1,6	8	$R(x)$	$\Rightarrow E 7,5$
1	9	Donc $P(x) \Rightarrow R(x)$	$\Rightarrow I 6,8$
1	10	$\forall x(P(x) \Rightarrow R(x))$	$\forall I 9$
	11	Donc $\forall x(P(x) \Rightarrow Q(x)) \wedge \forall x(Q(x) \Rightarrow R(x)) \Rightarrow \forall x(P(x) \Rightarrow R(x))$	$\Rightarrow I 1,9$

9. $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x \neg Q(x) \Rightarrow \exists x \neg P(x).$

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x \neg Q(x)$	
1	2	$\forall x(P(x) \Rightarrow Q(x))$	$\wedge E 1, 1$
1	3	$\exists x \neg Q(x)$	$\wedge E 2, 1$
1,4	4	Supposons $\neg Q(x)$	
1,4	5	$P(x) \Rightarrow Q(x)$	$\forall E 2$
1,4,6	6	Supposons $P(x)$	
1,4,6	7	$Q(x)$	$\Rightarrow E 5,6$
1,4,6	8	\perp	$\Rightarrow E 4,7$
1,4	9	Donc $\neg P(x)$	$\Rightarrow I 6,8$
1,4	10	$\exists x \neg P(x)$	$\exists I 9$
1	11	Donc $\neg Q(x) \Rightarrow \exists x \neg P(x)$	$\Rightarrow I 4,10$
1	12	$\exists x \neg P(x)$	$\exists E 3,11$
	13	Donc $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x \neg Q(x) \Rightarrow \exists x \neg P(x)$	$\Rightarrow I 1,12$

Exercice 97, page 147

Tout d'abord, les contraintes sur les formules sont nécessaires, car si elles ne sont pas respectées, nous ne pouvons pas généraliser, c'est-à-dire, introduire le « \forall ».

1. $\forall x \forall y P(x, y) \Rightarrow \forall y \forall x P(x, y).$

contexte	numéro	preuve	règle
1	1	Supposons $\forall x \forall y P(x, y)$	
1	2	$\forall y P(x, y)$	$\forall E 1, x$
1	3	$P(x, y)$	$\forall E 2, y$
1	4	$\forall x P(x, y)$	$\forall I 3$
1	5	$\forall y \forall x P(x, y)$	$\forall I 4$
	6	Donc $\forall x \forall y P(x, y) \Rightarrow \forall y \forall x P(x, y)$	$\Rightarrow I 1,5$

2. $\exists x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y).$

contexte	numéro	preuve	règle
1	1	Supposons $\exists x \exists y P(x, y)$	
1,2	2	Supposons $\exists y P(x, y)$	
1,2,3	3	Supposons $P(x, y)$	
1,2,3	4	$\exists x P(x, y)$	$\exists I$ 3, x
1,2,3	5	$\exists y \exists x P(x, y)$	$\exists I$ 4, y
1,2	6	Donc $P(x, y) \Rightarrow \exists y \exists P(x, y)$	$\Rightarrow I$ 5,3
1,2	7	$\exists y \exists x P(x, y)$	$\exists E$ 6,2
1	8	Donc $\exists y P(x, y) \Rightarrow \exists y \exists P(x, y)$	$\Rightarrow I$ 7,2
1	9	$\exists y \exists x P(x, y)$	$\exists E$ 8,1
	10	Donc $\exists x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y)$	$\Rightarrow I$ 1,9

3. $\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x \forall y P(x, y)$	
1,2	2	Supposons $\forall y P(x, y)$	
1,2	3	$P(x, y)$	$\forall E$ 2, y
1,2	4	$\exists x P(x, y)$	$\exists I$ 3, x
1,2	5	$\forall y \exists x P(x, y)$	$\forall I$ 4
1	6	Donc $\forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$	$\Rightarrow I$ 5,2
1	7	$\forall y \exists x P(x, y)$	$\forall E$ 6,1
	8	Donc $\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$	$\Rightarrow I$ 1,7

4. $\forall x (Q(x) \Rightarrow \forall y (R(y) \Rightarrow P(x, y))) \Rightarrow \forall y (R(y) \Rightarrow \forall x (Q(x) \Rightarrow P(x, y)))$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x (Q(x) \Rightarrow \forall y (R(y) \Rightarrow P(x, y)))$	
1	2	$Q(x) \Rightarrow \forall y (R(y) \Rightarrow P(x, y))$	$\forall E$ 1, x
1,3	3	Supposons $R(y)$	
1,3,4	4	Supposons $Q(x)$	
1,3,4	5	$\forall y (R(y) \Rightarrow P(x, y))$	$\Rightarrow E$ 2,4
1,3,4	6	$R(y) \Rightarrow P(x, y)$	$\forall E$, y
1,3,4	7	$P(x, y)$	$\Rightarrow E$ 6,3
1,3	8	Donc $Q(x) \Rightarrow P(x, y)$	$\Rightarrow I$ 7,4
1,3	9	$\forall x Q(x) \Rightarrow P(x, y)$	$\forall I$ 8
1	10	Donc $R(y) \Rightarrow \forall x (Q(x) \Rightarrow P(x, y))$	$\Rightarrow I$ 9,3
1	11	$\forall y (R(y) \Rightarrow \forall x (Q(x) \Rightarrow P(x, y)))$	$\forall I$ 10
	12	Donc $\forall x (Q(x) \Rightarrow \forall y (R(y) \Rightarrow P(x, y))) \Rightarrow \forall y (R(y) \Rightarrow \forall x (Q(x) \Rightarrow P(x, y)))$	$\Rightarrow I$ 1,11

Exercice 98, page 148

Les démonstrations de 2 et 3 sont ERRONÉES.

1. $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E 1, 1$
1	3	$\forall x Q(x)$	$\wedge E 2, 1$
1,4	4	Supposons $P(x)$	
1,4	5	$Q(x)$	$\forall E 3, x$
1,4	6	$P(x) \wedge Q(x)$	$\wedge I 4,5$
1,4	7	$\exists x(P(x) \wedge Q(x))$	$\exists I 6, x$
1	8	Donc $P(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 4,7$
1	9	$\exists x(P(x) \wedge Q(x))$	$\exists E 2,8$
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 1,9$

2. $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E 1, 1$
1	3	$\forall x Q(x)$	$\wedge E 2, 1$
1,4	4	Supposons $P(x)$	
1,4	5	$Q(x)$	$\forall E 3, x$
1,4	6	$P(x) \wedge Q(x)$	$\wedge I 4,5$
1	7	Donc $P(x) \Rightarrow P(x) \wedge Q(x)$	$\Rightarrow I 4,6$
1	8	$P(x) \wedge Q(x)$	$\exists E 2,7$ ERREUR
1	9	$\exists x(P(x) \wedge Q(x))$	$\exists I 8, x$
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 1,9$

Il y a une erreur à la ligne 8 car x est libre dans $P(x) \wedge Q(x)$.

3. $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x P(x) \wedge \forall x Q(x)$	
1	2	$\exists x P(x)$	$\wedge E 1, 1$
1	3	$\forall x Q(x)$	$\wedge E 2, 1$
1,4	4	Supposons $P(x)$	
1	5	Donc $P(x) \Rightarrow P(x)$	$\Rightarrow I 4,4$
1	6	$P(x)$	$\exists E 2,5$ ERREUR
1	7	$Q(x)$	$\forall E 3, x$
1	8	$P(x) \wedge Q(x)$	$\wedge I 6,7$
1	9	$\exists x(P(x) \wedge Q(x))$	$\exists I 8, x$
	10	Donc $\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x(P(x) \wedge Q(x))$	$\Rightarrow I 1,9$

Il y a une erreur à la ligne 6, car x est libre dans $P(x)$.

Exercice 99, page 148

contexte	numéro	preuve	règle
1	1	Supposons $\forall x \forall y P(x, y)$	
1	2	$\forall y P(z, y)$	$\forall E$ 1, z
1	3	$P(z, x)$	$\forall E$ 2, x
1	4	$\forall z P(z, x)$	$\forall I$ 3
1	5	$\forall x \forall z P(z, x)$	$\forall I$ 4
1	6	$\forall x \forall y P(y, x)$	Copie 5
	7	Donc $\forall x \forall y P(x, y) \Rightarrow \forall x \forall y P(y, x)$	$\Rightarrow I$ 1,6

Exercice 100, page 148

1. $\forall x(Q \wedge P(x)) \Rightarrow Q \wedge \forall x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(Q \wedge P(x))$	
1	2	$Q \wedge P(x)$	$\forall E$ 1, x
1	3	Q	$\wedge E$ 1 2
1	4	$P(x)$	$\wedge E$ 2 2
1	5	$\forall x P(x)$	$\forall I$ 4
1	6	$Q \wedge \forall x P(x)$	$\wedge I$ 3,5
	7	Donc $\forall x(Q \wedge P(x)) \Rightarrow Q \wedge \forall x P(x)$	$\Rightarrow I$ 1,6

2. $Q \wedge \forall x P(x) \Rightarrow \forall x(Q \wedge P(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $Q \wedge \forall x P(x)$	
1	2	Q	$\wedge E$ 1 1
1	3	$\forall x P(x)$	$\wedge E$ 1 1
1	4	$P(x)$	$\forall E$ 3, x
1	5	$Q \wedge P(x)$	$\wedge I$ 2,4
1	6	$\forall x(Q \wedge P(x))$	$\forall I$ 5
	7	Donc $Q \wedge \forall x P(x) \Rightarrow \forall x(Q \wedge P(x))$	$\Rightarrow I$ 1,6

3. $\forall x(Q \vee P(x)) \Rightarrow Q \vee \forall x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(Q \vee P(x))$	
1,2	2	Supposons $\neg(Q \vee \forall xP(x))$	
1,2	3	$Q \vee P(x)$	$\forall E$ 1, x
1,2,4	4	Supposons $\neg P(x)$	
1,2,4,5	5	Supposons $\neg Q$	
1,2,4,5	6	\perp	$\forall E$ 3,4,5
1,2,4	7	Donc $\neg\neg Q$	$\Rightarrow I$ 5,6
1,2,4	8	Q	RAA 7
1,2,4	9	$Q \vee \forall xP(x)$	$\forall I$ 8
1,2,4	10	\perp	$\Rightarrow E$ 2,9
1,2	11	Donc $\neg\neg P(x)$	$\Rightarrow I$ 4,10
1,2	12	$P(x)$	RAA 11
1,2	13	$\forall xP(x)$	$\forall I$ 12
1,2	14	$Q \vee \forall xP(x)$	$\forall I$ 13
1,2	15	\perp	$\Rightarrow E$ 2,14
1	16	Donc $\neg\neg(Q \vee \forall xP(x))$	$\Rightarrow I$ 2,15
1	17	$Q \vee \forall xP(x)$	RAA 16
	18	Donc $\forall x(Q \vee P(x)) \Rightarrow Q \vee \forall xP(x)$	$\Rightarrow I$ 1,17

4. $Q \vee \forall xP(x) \Rightarrow \forall x(Q \vee P(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $Q \vee \forall xP(x)$	
1,2	2	Supposons Q	
1,2	3	$Q \vee P(x)$	$\forall I$ 2
1	4	Donc $Q \Rightarrow Q \vee P(x)$	$\Rightarrow I$ 2,3
1,5	5	Supposons $\forall xP(x)$	
1,5	6	$P(x)$	$\forall E$ 5, x
1,5	7	$Q \vee P(x)$	$\forall I$ 6
1	8	Donc $\forall xP(x) \Rightarrow Q \vee P(x)$	$\Rightarrow I$ 8,5
1	9	$Q \vee P(x)$	$\forall E$ 8,1,4
1	10	$\forall x(Q \vee P(x))$	$\forall I$ 9
	11	Donc $Q \vee \forall xP(x) \Rightarrow \forall x(Q \vee P(x))$	$\Rightarrow I$ 1,10

5. $\exists x(Q \wedge P(x)) \Rightarrow Q \wedge \exists xP(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x(Q \wedge P(x))$	
1,2	2	Supposons $Q \wedge P(x)$	
1,2	3	Q	$\wedge E$ 1 2
1,2	4	$P(x)$	$\wedge E$ 2 2
1,2	5	$\exists xP(x)$	$\exists I$ 4, x
1,2	6	$Q \wedge \exists xP(x)$	$\wedge I$ 3,5
1	7	Donc $Q \wedge P(x) \Rightarrow Q \wedge \exists xP(x)$	$\Rightarrow I$ 2,6
1	8	$Q \wedge \exists xP(x)$	$\exists E$ 7,1
	9	Donc $\exists x(Q \wedge P(x)) \Rightarrow Q \wedge \exists xP(x)$	$\Rightarrow I$ 1,8

6. $Q \wedge \exists xP(x) \Rightarrow \exists x(Q \wedge P(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $Q \wedge \exists xP(x)$	
1	2	Q	$\wedge E 1$
1	3	$\exists xP(x)$	$\wedge E 1$
1,4	4	Supposons $P(x)$	
1,4	5	$Q \wedge P(x)$	$\wedge I 2,4$
1,4	6	$\exists x(Q \wedge P(x))$	$\exists I 5, x$
1	7	Donc $P(x) \Rightarrow \exists x(Q \wedge P(x))$	$\Rightarrow I 4,6$
1	8	$\exists x(Q \wedge P(x))$	$\exists E 3,7$
	9	Donc $Q \wedge \exists xP(x) \Rightarrow \exists x(Q \wedge P(x))$	$\Rightarrow I 1,8$

7. $\exists x(Q \vee P(x)) \Rightarrow Q \vee \exists xP(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x(Q \vee P(x))$	
1,2	2	Supposons $Q \vee P(x)$	
1,2,3	3	Supposons Q	
1,2,3	4	$Q \vee \exists xP(x)$	$\vee I 1$
1,2	5	Donc $Q \Rightarrow Q \vee \exists xP(x)$	$\Rightarrow I 3,4$
1,2,6	6	Supposons $P(x)$	
1,2,6	7	$\exists xP(x)$	$\exists I 6, x$
1,2,6	8	$Q \vee \exists xP(x)$	$\vee I 2$
1,2	9	Donc $P(x) \Rightarrow Q \vee \exists xP(x)$	$\Rightarrow I 6,8$
1,2	10	$Q \vee \exists xP(x)$	$\vee E 2,5,9$
1	11	Donc $Q \vee P(x) \Rightarrow Q \vee \exists xP(x)$	$\Rightarrow I 2,10$
1	12	$Q \vee \exists xP(x)$	$\exists E 1,11$
	13	Donc $\exists x(Q \vee P(x)) \Rightarrow Q \vee \exists xP(x)$	$\Rightarrow I 1,12$

8. $Q \vee \exists xP(x) \Rightarrow \exists x(Q \vee P(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $Q \vee \exists xP(x)$	
1,2	2	Supposons Q	
1,2	3	$Q \vee P(x)$	$\vee I 2$
1,2	4	$\exists x(Q \vee P(x))$	$\exists I 3, x$
1	5	Donc $Q \Rightarrow \exists x(Q \vee P(x))$	$\Rightarrow I 2,4$
1,6	6	Supposons $\exists xP(x)$	
1,6,7	7	Supposons $P(x)$	
1,6,7	8	$Q \vee P(x)$	$\vee I 2$
1,6,7	9	$\exists x(Q \vee P(x))$	$\exists I 8, x$
1,6	10	Donc $P(x) \Rightarrow \exists x(Q \vee P(x))$	$\Rightarrow I 7,9$
1,6	11	$\exists x(Q \vee P(x))$	$\exists E 6,10$
1	12	Donc $\exists xP(x) \Rightarrow \exists x(Q \vee P(x))$	$\Rightarrow I 6,11$
1	13	$\exists x(Q \vee P(x))$	$\vee E 1,5,12$
	14	Donc $Q \vee \exists xP(x) \Rightarrow \exists x(Q \vee P(x))$	$\Rightarrow I 1,13$

Exercice 101, page 148

contexte	numéro	preuve	règle
1	1	Supposons $\neg \exists x A$	
1,2	2	Supposons A	
1,2	3	$\exists x A$	$\exists I$ 2, x
1,2	4	\perp	$\Rightarrow E$ 1,3
1	5	Donc $\neg A$	$\Rightarrow I$ 2,4
1	6	$\forall x \neg A$	$\forall I$ 5
	7	Donc $\neg \exists x A \Rightarrow \forall x \neg A$	$\Rightarrow I$ 1,6

Remarquez que dans cette preuve nous utilisons le fait que x est libre pour lui-même.

Exercice 102, page 148

1. $R(a, c) \wedge (a = b) \Rightarrow R(b, c)$.

contexte	numéro	preuve	règle
1	1	Supposons $R(a, c) \wedge (a = b)$	
1	2	$R(a, c)$	$\wedge E$ 1
1	3	$a = b$	$\wedge E$ 1
1	4	$R(\underline{b}, c)$	Congruence 2,3
	5	Donc $R(a, c) \wedge (a = b) \Rightarrow R(b, c)$	$\Rightarrow I$ 1,4

2. Preuve de $x = y \Rightarrow f(x, z) = f(y, z)$.

contexte	numéro	preuve	règle
1	1	Supposons $x = y$	
1	2	$f(x, z) = f(x, z)$	Réflexivité
1	3	$f(x, z) = f(\underline{y}, z)$	Congruence 1,2
	4	Donc $x = y \Rightarrow f(x, z) = f(y, z)$	$\Rightarrow I$ 1,3

L'égalité 1 permet de remplacer dans 2 l'occurrence qui est soulignée en 3

3. Preuve de $\forall x \exists y (x = y)$.

contexte	numéro	preuve	règle
	1	$x = x$	Réflexivité
	2	$\exists y (x = y)$	$\exists I$ 1, x
	3	$\forall x \exists y (x = y)$	$\forall I$ 2

Pour la deuxième ligne, remarquez que $x = x \equiv (x = y) < y := x >$.

4. $\exists x \forall y x = y \Rightarrow \forall x \forall y x = y$

contexte	numéro	ligne	règle
1	1	Supposons $\exists x \forall y x = y$	
1,2	2	Supposons $\forall y x = y$	
1,2	3	$x = u$	$\forall E$ 2, u
1,2	4	$x = y$	$\forall E$ 2, y
1,2	5	$\underline{u} = y$	Congruence 3,4
1,2	6	$\forall y u = y$	$\forall I$ 5
1,2	7	$\forall u \forall y u = y$	$\forall I$ 6
1,2	8	$\forall x \forall y x = y$	Copie 7
1	9	Donc $\forall y x = y \Rightarrow \forall x \forall y x = y$	$\Rightarrow I$ 2,8
1	10	$\forall x \forall y x = y$	$\exists E$ 8,9
	11	Donc $\exists x \forall y x = y \Rightarrow \forall x \forall y x = y$	$\Rightarrow I$ 1,11

Exercice 103, page 148

contexte	numéro	preuve	règle
1	1	Supposons $\exists xP(x) \wedge \forall x\forall y(P(x) \wedge P(y) \Rightarrow x = y)$	
1	2	$\exists xP(x)$	$\wedge E 1$
1	3	$\forall x\forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\wedge E 2$
1,4	4	Supposons $P(x)$	
1,4,5	5	Supposons $P(y)$	
1,4,5	6	$P(x) \wedge P(y)$	$\wedge I 4,5$
1,4,5	7	$\forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\forall E 3, x$
1,4,5	8	$P(x) \wedge P(y) \Rightarrow x = y$	$\forall E 7, y$
1,4,5	9	$x = y$	$\Rightarrow E 6,8$
1,4	10	Donc $P(y) \Rightarrow x = y$	$\Rightarrow I 5,9$
1,4	11	$\forall y(P(y) \Rightarrow x = y)$	$\forall I 10$
1,4	12	$P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	$\wedge I 4,11$
1,4	13	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	$\exists I 12, x$
1	14	Donc $P(x) \Rightarrow \exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	$\Rightarrow I 4,13$
1	15	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	$\exists E 2,14$
	16	Donc $\exists xP(x) \wedge \forall x\forall y(P(x) \wedge P(y) \Rightarrow x = y) \Rightarrow \exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	$\Rightarrow I 1,15$

Exercice 104, page 149

1. Des hypothèses (a) et (b), nous ne pouvons pas déduire $\forall n(0 + n = n)$. Considérons l'interprétation de domaine $\{0, 1\}$ avec :

— le sens de s défini par : $s(0) = 0, s(1) = 1$.

— le sens de $+$ défini par : $0 + 0 = 0, 0 + 1 = 0, 1 + 0 = 1, 1 + 1 = 0$.

Dans ce domaine la propriété (a) vaut : $(0 + 0 = 0) \cdot (1 + 0 = 1) = 1$. La propriété (b) est aussi vraie, car $n + s(p) = n + p = s(n + p)$, vu que s est l'identité sur le domaine. Par contre (c) vaut 0, car l'addition n'est pas commutative dans l'interprétation. Donc (c) n'est pas conséquence de (a) et (b).

2. Commençons par une preuve « informelle » de cette propriété. D'après (a) il est évident que $0 + 0 = 0$ donc que $P(0)$. Supposons $P(n)$ et prouvons $P(s(n))$: $P(n)$ s'écrit $n + 0 = n$. D'après (b) nous avons $0 + s(n) = s(n + 0)$. Par suite $0 + s(n) = s(n)$, c'est-à-dire $P(s(n))$.

La preuve formelle reprend ce plan de preuve, mais elle est beaucoup plus longue, car elle explicite la définition de P et ne saute aucune étape de la récurrence. Dans tous les exercices, où la preuve, en déduction naturelle, peut être longue, nous conseillons d'écrire une preuve « informelle » avant de la transcrire en déduction naturelle.

contexte	numéro	preuve	règle
1	1	Supposons $(a) \wedge (b) \wedge (c) \wedge (d)$	
1	2	$\forall n(n + 0 = n)$	$\wedge E 1$
1	3	$(b) \wedge (c) \wedge (d)$	$\wedge E 2$
1	4	$\forall n\forall p(n + s(p) = s(n + p))$	$\wedge E 1$
1	5	$(c) \wedge (d)$	$\wedge E 2$
1	6	$\forall n(P(n) \Leftrightarrow 0 + n = n)$	$\wedge E 1$
1	7	$P(0) \wedge \forall n(P(n) \Rightarrow P(s(n))) \Rightarrow \forall nP(n)$	$\wedge E 2$

Nous prouvons $P(0)$.

contexte	numéro	preuve	règle
1	8	$0 + 0 = 0$	$\forall E 2, 0$
1	9	$P(0) \Leftrightarrow 0 + 0 = 0$	$\forall E 6, 0$
1	10	$0 + 0 = 0 \Rightarrow P(0)$	$\wedge E 2$
1	11	$P(0)$	$\Rightarrow E 10,8$

Nous prouvons $\forall n(P(n) \Rightarrow P(s(n)))$.

contexte	numéro	preuve	règle
1,12	12	Supposons $P(n)$	
1,12	13	$P(n) \Leftrightarrow 0 + n = n$	$\forall E$ 6, n
1,12	14	$P(n) \Rightarrow 0 + n = n$	$\wedge E$ 13
1,12	15	$0 + n = n$	$\Rightarrow E$ 14,12
1,12	16	$\forall p(0 + s(p) = s(0 + p))$	$\forall E$ 4, 0
1,12	17	$0 + s(n) = s(0 + n)$	$\forall E$ 16, n
1,12	18	$0 + s(n) = s(\underline{n})$	Congruence 15,17
1,12	19	$P(s(n)) \Leftrightarrow 0 + s(n) = s(n)$	$\forall E$ 6, $s(n)$
1,12	20	$0 + s(n) = s(n) \Rightarrow P(s(n))$	$\wedge E$ 19
1,12	21	$P(s(n))$	$\Rightarrow E$ 20,18
1	22	Donc $P(n) \Rightarrow P(s(n))$	$\Rightarrow I$ 12,21
1	23	$\forall n(P(n) \Rightarrow P(s(n)))$	$\forall I$ 22

Nous appliquons le principe de récurrence.

contexte	numéro	preuve	règle
1	24	$P(0) \wedge \forall n(P(n) \Rightarrow P(s(n)))$	$\wedge I$ 11,23
1	25	$\forall n P(n)$	$\Rightarrow E$ 7,24
	26	Donc $(a) \wedge (b) \wedge (c) \wedge (d) \Rightarrow \forall n P(n)$	$\Rightarrow I$ 1,24

Exercice 105, page 149

1. Preuve de $(\exists x p(x) \Rightarrow \forall x q(x)) \Rightarrow \forall x(p(x) \Rightarrow q(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x p(x) \Rightarrow \forall x q(x)$	
1,2	2	Supposons $p(x)$	
1,2	3	$\exists x p(x)$	$\exists I$ 2
1,2	4	$\forall x q(x)$	$\Rightarrow E$ 1,3
1,2	5	$q(x)$	$\forall E$ 4, x
1	6	Donc $p(x) \Rightarrow q(x)$	$\Rightarrow I$ 2,5
1	7	$\forall x(p(x) \Rightarrow q(x))$	$\forall I$ 6
	8	Donc $(\exists x p(x) \Rightarrow \forall x q(x)) \Rightarrow \forall x(p(x) \Rightarrow q(x))$	$\Rightarrow I$ 1,7

2. Preuve de $\exists x p(x) \wedge \forall x(p(x) \Rightarrow p(f(x))) \Rightarrow \exists x p(f(f(x)))$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x p(x) \wedge \forall x(p(x) \Rightarrow p(f(x)))$	
1	2	$\exists x p(x)$	$\wedge E$ 1
1,3	3	Supposons $p(x)$	
1,3	4	$\forall x(p(x) \Rightarrow p(f(x)))$	$\wedge E$ 2
1,3	5	$p(x) \Rightarrow p(f(x))$	$\forall E$ 4, x
1,3	6	$p(f(x))$	$\Rightarrow E$ 3,5
1,3	7	$p(f(x)) \Rightarrow p(f(f(x)))$	$\forall E$ 4, $f(x)$
1,3	8	$p(f(f(x)))$	$\Rightarrow E$ 6,7
1,3	9	$\exists x p(f(f(x)))$	$\exists I$, x
1	10	Donc $p(x) \Rightarrow \exists x p(f(f(x)))$	$\Rightarrow I$ 3,9
1	11	$\exists x p(f(f(x)))$	$\exists E$ 2,10
	12	Donc $\exists x p(x) \wedge \forall x(p(x) \Rightarrow p(f(x))) \Rightarrow \exists x p(f(f(x)))$	$\Rightarrow I$ 1,12

3. Prouvons que pour tout entier naturel n , $\exists x p(x), \forall x(p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x))$.

(a) Preuve en déduction naturelle de :

$$(*) \forall x(p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x)) \Rightarrow \exists x p(f^{n+1}(x))$$

environnement			
référence		formule	
i		$\forall x(p(x) \Rightarrow p(f(x)))$	
contexte	numéro	preuve	règle
1	1	Supposons $\exists x p(f^n(x))$	
1,2	2	Supposons $p(f^n(x))$	
1,2	3	$p(f^n(x)) \Rightarrow p(f^{n+1}(x))$	$\forall E i, f^n(x)$
1,2	4	$p(f^{n+1}(x))$	$\Rightarrow E 2,3$
1,2	5	$\exists x p(f^{n+1}(x))$	$\exists I 4, x$
1	6	Donc $p(f^n(x)) \Rightarrow \exists x p(f^{n+1}(x))$	$\Rightarrow I 2,5$
1	7	$\exists x p(f^{n+1}(x))$	$\exists E 1,6$
	8	Donc $\exists x p(f^n(x)) \Rightarrow \exists x p(f^{n+1}(x))$	$\Rightarrow I 1,7$

(b) Notons $P(n)$ la propriété :

$$\exists x p(x), \forall x(p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x))$$

Montrons qu'elle est vraie pour tout entier naturel n .

La propriété $P(0)$ est vraie car la conclusion $\exists x p(x)$ fait partie des hypothèses

Supposons $P(n)$ vraie. D'après (*) et par monotonie de \vdash , on a :

$$(**) \exists x p(x), \forall x(p(x) \Rightarrow p(f(x))) \vdash \exists x p(f^n(x)) \Rightarrow \exists x p(f^{n+1}(x))$$

D'après $P(n), (**) et la composition, $P(n+1)$ est aussi vraie.$

Donc $P(n)$ est vraie pour tout entier naturel n .

Exercice 106, page 149

1. $\neg \forall x P(x) \vee \neg \exists y Q(y) \Rightarrow \neg (\forall x P(x) \wedge \exists y Q(y))$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg \forall x P(x) \vee \neg \exists y Q(y)$	
1,2	2	Supposons $\forall x P(x) \wedge \exists y Q(y)$	
1,2,3	3	Supposons $\neg \forall x P(x)$	
1,2,3	4	$\forall x P(x)$	$\wedge E 1,2$
1,2,3	5	\perp	$\Rightarrow E 3,4$
1,2	6	Donc $\neg \forall x P(x) \Rightarrow \perp$	$\Rightarrow I 3,5$
1,2,7	7	Supposons $\neg \exists y Q(y)$	
1,2,7	8	$\exists y Q(y)$	$\wedge E 2,2$
1,2,7	9	\perp	$\Rightarrow E 3,4$
1,2	10	Donc $\neg \exists y Q(y) \Rightarrow \perp$	$\Rightarrow I 3,5$
1,2	11	\perp	$\vee E 1,6,10$
1	12	Donc $\neg (\forall x P(x) \wedge \exists y Q(y))$	$\Rightarrow I 2,11$
	13	Donc $\neg \forall x P(x) \vee \neg \exists y Q(y) \Rightarrow \neg (\forall x P(x) \wedge \exists y Q(y))$	$\Rightarrow I 1,12$

2. $\forall x \forall y (P(y) \Rightarrow R(x)) \Rightarrow \exists y P(y) \Rightarrow \forall x R(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x \forall y (P(y) \Rightarrow R(x))$	
1,2	2	Supposons $\exists y P(y)$	
1,2	3	$\forall y (P(y) \Rightarrow R(x))$	$\forall E$ 1, x
1,2	4	$P(y) \Rightarrow R(x)$	$\forall E$ 3, y
1,2,5	5	Supposons $P(y)$	
1,2,5	6	$R(x)$	$\Rightarrow E$ 4,5
1,2,5	7	$\forall x R(x)$	$\forall I$ 6
1,2	8	Donc $P(y) \Rightarrow \forall x R(x)$	$\Rightarrow I$ 5,7
1,2	9	$\forall x R(x)$	$\exists E$ 2,8
1	10	Donc $\exists y P(y) \Rightarrow \forall x R(x)$	$\Rightarrow I$ 2,9
	11	Donc $\forall x \forall y (P(y) \Rightarrow R(x)) \Rightarrow \exists y P(y) \Rightarrow \forall x R(x)$	$\Rightarrow I$ 1,10

3. $\neg \forall x \neg P(x) \Rightarrow \exists x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\neg \forall x \neg P(x)$	
1,2	2	Supposons $\neg \exists x P(x)$	
1,2,3	3	Supposons $P(x)$	
1,2,3	4	$\exists x P(x)$	$\exists I$ 3
1,2,3	5	\perp	$\Rightarrow E$ 2,4
1,2	6	Donc $\neg P(x)$	$\Rightarrow I$ 3,5
1,2	7	$\forall x \neg P(x)$	$\forall I$ 6
1	8	\perp	$\Rightarrow E$ 7,1
1	9	Donc $\neg \neg \exists x P(x)$	$\Rightarrow I$ 8,2
1	10	$\exists x P(x)$	RAA 9
	11	Donc $\neg \forall x \neg P(x) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 1,10

Exercice 107, page 149

1. $\exists x (Q(x) \Rightarrow P(x)) \wedge \forall x Q(x) \Rightarrow \exists x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x (Q(x) \Rightarrow P(x)) \wedge \forall x Q(x)$	
1	2	$\exists x (Q(x) \Rightarrow P(x))$	$\wedge E$ 1 1
1	3	$\forall x Q(x)$	$\wedge E$ 2 1
1,4	4	Supposons $Q(x) \Rightarrow P(x)$	
1,4	5	$Q(x)$	$\forall E$ 3, x
1,4	6	$P(x)$	$\Rightarrow E$ 4,5
1,4	7	$\exists x P(x)$	$\exists I$ 6, x
1	8	Donc $(Q(x) \Rightarrow P(x)) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 4,7
1	9	$\exists x P(x)$	$\exists E$ 2,8
	10	Donc $\exists x (Q(x) \Rightarrow P(x)) \wedge \forall x Q(x) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 1,9

2. $\forall x \forall y (R(x,y) \Rightarrow \neg R(y,x)) \Rightarrow \forall x \neg R(x,x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x \forall y (R(x,y) \Rightarrow \neg R(y,x))$	
1	2	$\forall y (R(x,y) \Rightarrow \neg R(y,x))$	$\forall E$ 1, x
1	3	$R(x,x) \Rightarrow \neg R(x,x)$	$\forall E$ 2, x
1,2	4	Supposons $R(x,x)$	
1,2	5	$\neg R(x,x)$	$\Rightarrow E$ 3,4
1,2	6	\perp	$\Rightarrow E$ 5,4
1	7	Donc $\neg R(x,x)$	$\Rightarrow I$ 4,6
1	8	$\forall x \neg R(x,x)$	$\forall I$ 7
	9	Donc $\forall x \forall y (R(x,y) \Rightarrow \neg R(y,x)) \Rightarrow \forall x \neg R(x,x)$	$\Rightarrow I$ 1,8

Exercice 108, page 149

1. $\exists x(P(x) \vee Q(x)) \wedge \forall x \neg Q(x) \Rightarrow \exists x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x(P(x) \vee Q(x)) \wedge \forall x \neg Q(x)$	
1,2	2	Supposons $P(x) \vee Q(x)$	
1,2,3	3	Supposons $P(x)$	
1,2,3	4	$\exists x P(x)$	$\exists I$ 3, x
1,2	5	Donc $P(x) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 3,4
1,2,6	6	Supposons $Q(x)$	
1,2,6	7	$\forall x \neg Q(x)$	$\wedge E$ 1
1,2,6	8	$\neg Q(x)$	$\forall E$ 7, x
1,2,6	9	\perp	$\Rightarrow E$ 6,8
1,2,6	10	$\exists x P(x)$	$Ef q$ 9
1,2	11	Donc $Q(x) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 6,10
1,2	12	$\exists x P(x)$	$\vee E$ 2,5,11
1	13	Donc $P(x) \vee Q(x) \Rightarrow \exists x P(x)$	$\Rightarrow I$ 2,12
1	14	$\exists x(P(x) \vee Q(x))$	$\wedge E$ 1
1	15	$\exists x P(x)$	$\exists E$ 13,14
	16	Donc $\exists x(P(x) \vee Q(x)) \wedge \forall x \neg Q(x) \Rightarrow \exists x P(x)$	\Rightarrow 1,15

2. $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x(P(x) \wedge R(x)) \Rightarrow \exists x(Q(x) \wedge R(x))$.

contexte	numéro	preuve	règle
1	1	Supposons $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x(P(x) \wedge R(x))$	
1,2	2	Supposons $P(x) \wedge R(x)$	
1,2	3	$P(x)$	$\wedge E$ 1
1,2	4	$\forall x(P(x) \Rightarrow Q(x))$	$\wedge E$ 1
1,2	5	$P(x) \Rightarrow Q(x)$	$\forall E$ 4, x
1,2	6	$Q(x)$	$\Rightarrow E$ 3,5
1,2	7	$R(x)$	$\wedge E$ 2
1,2	8	$Q(x) \wedge R(x)$	$\wedge I$ 6,7
1,2	9	$\exists x(Q(x) \wedge R(x))$	$\exists I$ 8, x
1	10	Donc $P(x) \wedge R(x) \Rightarrow \exists x(Q(x) \wedge R(x))$	$\Rightarrow I$ 2,9
1	11	$\exists x(P(x) \wedge R(x))$	$\wedge E$ 1
1	12	$\exists x(Q(x) \wedge R(x))$	$\exists E$ 10,11
	13	Donc $\forall x(P(x) \Rightarrow Q(x)) \wedge \exists x(P(x) \wedge R(x)) \Rightarrow \exists x(Q(x) \wedge R(x))$	$\Rightarrow I$ 1,12

3. $\exists x \neg(P(x) \vee \neg P(x)) \Rightarrow \forall x P(x)$.

contexte	numéro	preuve	règle
1	1	Supposons $\exists x \neg(P(x) \vee \neg P(x))$	
1,2	2	Supposons $\neg(P(x) \vee \neg P(x))$	
1,2,3	3	Supposons $\neg P(x)$	
1,2,3	4	$P(x) \vee \neg P(x)$	$\vee I2\ 3$
1,2,3	5	\perp	$\Rightarrow E\ 2,4$
1,2	6	Donc $\neg \neg P(x)$	$\Rightarrow I\ 3,5$
1,2	7	$P(x)$	$RAA\ 6$
1,2	8	$P(x) \vee \neg P(x)$	$\vee I1\ 7$
1,2	9	\perp	$\Rightarrow E\ 2,8$
1	10	Donc $\neg(P(x) \vee \neg P(x)) \Rightarrow \perp$	$\Rightarrow I\ 2,9$
1	11	\perp	$\exists E\ 1,10$
1	12	$\forall x P(x)$	$Efq\ 11$
	13	Donc $\exists x \neg(P(x) \vee \neg P(x)) \Rightarrow \forall x P(x)$	$\Rightarrow I\ 1,12$

Bibliographie

- [1] Collin Allen and Michael Hand. *Logic Primer*. MIT, 2001.
- [2] Peter B. Andrews. *An introduction to mathematical logic : to truth through proof*. Academic Press, 1986.
- [3] Franz Baader and Wayne Snyder. Unification theory. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning (in 2 volumes)*, pages 445–532. Elsevier and MIT Press, 2001.
- [4] Alonzo Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1(1) :40–41, 1936.
- [5] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.
- [6] René Cori and Daniel Lascar. *Logique mathématique Cours et exercices I. Calcul propositionnel, algèbres de Boole, calcul des prédicats*. Masson, 1993.
- [7] René Cori and Daniel Lascar. *Logique mathématique Cours et exercices II. Fonctions récursives, théorème de Gödel, théorie des ensembles, théories des modèles*. Masson, 1993.
- [8] René David, Karim Nour, and Christophe Raffali. *Introduction à la logique*. Dunod, 2001.
- [9] Martin Davis, George W. Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Communications of The ACM*, 5 :394–397, 1962.
- [10] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM (JACM)*, 7(3) :201–215, 1960.
- [11] Herbert B. Enderton. *A mathematical Introduction to Logic*. Academic Press, 2001.
- [12] Gerhard Gentzen. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, 39 :176–210, 1935. 10.1007/BF01201353.
- [13] Gerhard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39 :405–431, 1935. 10.1007/BF01201363.
- [14] Roger Godement. *Cours d'algèbre*. Hermann, 1973.
- [15] Gérard P. Huet. unification in typed lambda calculus. In Corrado Böhm, editor, *Lambda-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, March 25-27, 1975*, volume 37 of *Lecture Notes in Computer Science*, pages 192–212. Springer, 1975.
- [16] Gérard P. Huet. Higher order unification 30 years later. In Victor Carreño, César Muñoz, and Sofiène Tahar, editors, *Theorem Proving in Higher Order Logics, 15th International Conference, TPHOLs 2002, Hampton, VA, USA, August 20-23, 2002, Proceedings*, volume 2410 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2002.
- [17] Michael Huth and Mark Ryan. *Logic in Computer Science*. Cambridge University Press, 2004.
- [18] Jacquemin. *Logique et Mathématiques 109 exercices corrigées*. Masson, 1994.
- [19] Stephen C. Kleene. *Logique Mathématique*. Armand Colin, 1971.
- [20] Thierry Lucas, Isabelle Berlinger, and Isabelle De Greef. *Initiation à la logique formelle*. De Boeck, 2003.
- [21] Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Trans. Program. Lang. Syst.*, 4 :258–282, April 1982.
- [22] William McCune. Prover9 and mace4. <http://www.cs.unm.edu/~mccune/prover9/>, 2005–2010.
- [23] J. Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 12(1) :23–41, January 1965.
- [24] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42 :230–265, 1937.