Université de Tours

Logique pour l'informatique

Jean-Yves Antoine, Jeanne Villaneau

Sommaire

1. Introduction

La logique : quoi ? pourquoi ? pour qui ?

OBJECTIFS LIES AU CHAPITRE 1 - INTRODUCTION

1.1.1. Vérité et validité
1.1.2. Langages formels
1.1.3. Approche formelle (théorie de la preuve): axiome, théorème, conséquence syntaxique
1.1.4. Approche sémantique (théorie des modèles): interprétation, modèle, tautologie, équivalence
1.1.5. Déduction sémantique : conséquence sémantique et validité d'un raisonnement
1.1.6. Approches formelles et sémantiques : quelles relations (consistance, complétude).
1.1.7. Décidabilité

1. QUELLE LOGIQUE POUR QUELLES SCIENCES ?

Le livre de logique que vous avez entre les mains est avant tout destiné à des étudiants se préparant à une carrière dans l'un des multiples domaines d'activités liés à l'informatique. Les notions que nous allons découvrir ensemble constituent en effet une base théorique et conceptuelle utile, voire incontournable, dans de nombreux champs d'intervention de l'informatique. Elles constituent de ce fait un prérequis pour divers enseignements que vous serez amenés à rencontrer au cours de votre scolarité supérieure. Ce sera par exemple le cas des bases de données, comme vous pourrez le voir brièvement à la fin de cet ouvrage.

Pourtant, vous découvrirez que la modélisation logique emploie souvent des notations qui sont proches de celles utilisées en mathématiques. Cela n'est pas étonnant. A l'origine, la logique est en effet avant tout une science du raisonnement. Elle permet d'analyser les principes sur lesquels reposent les démonstrations mathématiques. Il vous faut également savoir que tout étudiant de philosophie est amené à suivre dans son cursus un enseignement de logique : l'utilité de la logique dans ce cas est de servir de support au raisonnement humain. Le philosophe se doit ainsi de maîtriser des notions que vous allez découvrir dans la première partie de ce livre (*logique des propositions*). Généralement, il mobilisera pour son raisonnement des méthodes graphiques (liées à une vision ensembliste du calcul de la vérité) là où nous vous ferons découvrir d'autres techniques plus adaptées à l'informaticien. La logique des propositions englobe également sous le terme de logique booléenne (du nom du mathématicien du XIX^e siècle George Boole) des notions qui intéressent aussi bien l'informaticien que l'électronicien.

Informatique, électronique, mathématiques, philosophie : la logique constitue une science qui sert de socle à de nombreuses disciplines. Tout en rendant compte de toute la richesse de la théorie, ce livre est toutefois orienté vers les besoins de l'informaticien. À ce titre, il est important de noter que l'ouvrage privilégie des choix de terminologie ou de représentation adaptés au futur ingénieur en informatique. Il ne s'agit donc pas d'un livre de logique mathématique, et le pur logicien ne trouverait sans doute pas complètement son compte dans cette présentation adaptée de la théorie logique.

Pour l'informaticien, les connaissances utiles liées à la théorie logique restent toutefois très étendues. On observe en effet que la logique intervient dans de nombreux pans de la discipline :

- La logique est ainsi au centre de l'architecture des ordinateurs, machines binaires manipulant des bits (0 ou 1). La conception d'un ordinateur repose en effet sur la logique booléenne, dont nous étudierons un équivalent lors de la première partie de ce livre (cf. chapitre 2),
- Les bases de données qui sont du développement de l'informatique du XXIe siècle et du *Big Data*, reposent sur un modèle mathématique appelé algèbre relationnelle. Nous verrons à la fin de ce livre que ce modèle relationnel peut également être interprété dans une perspective

logique, que l'on appelle le calcul des tuples et que nous étudierons (cf. chapitre 5),

- Le Web sémantique, sur lequel repose le développement d'un internet de la connaissance, et ses langages tels que OWL-DL et OWL-Lite, ne saurait exister sans les logiques de description. Ces logiques constituent un sous-langage de la logique de prédicats du premier ordre, que nous étudierons dans la seconde partie de ce livre (cf. chapitre 3),
- L'Intelligence Artificielle, qui reste un champ d'investigation actif de l'informatique, repose de même fréquemment sur le paradigme de la logique de prédicats. Nous en verrons un très bref aperçu dans ce livre, dans une partie applicative consacrée à une découverte de la programmation logique (cf. chapitre 4).

Bien d'autres champs d'application de la logique pourraient être cités. Cette brève introduction n'a qu'un but : vous persuader que l'acquisition de connaissances en logique constitue un des fondements théoriques essentiels que doit acquérir le futur ingénieur en informatique. Remarquons d'ailleurs qu'il en va de même de bien d'autres domaines des mathématiques, sans lesquels l'informatique n'existerait pas. On oublie ainsi fréquemment que l'informatique est une fille des mathématiques : ses pères fondateurs, tels Alex Turing pour la théorie des calculateurs ou John Von Neumann pour l'architecture des ordinateurs, étaient des mathématiciens.

2. QU'EST-CE QUE LA LOGIQUE ?

2.1. UNE DEFINITION HISTORIQUE

Rappelant que le terme *logique* est issu étymologiquement du grec *logos*, qui signifie *raison*, le dictionnaire Larousse en donne la définition suivante :

Logique (n.f) *science du raisonnement en lui-même, abstraction faite de la matière à laquelle il s'applique*.

Cette définition recouvre le propos de cette discipline, qui revient à étudier ce qui fait la validité d'un raisonnement. En précisant « *abstraction faite de la matière à laquelle elle s'applique* », cette définition explique par ailleurs en quoi la logique peut intéresser plusieurs disciplines différentes.

De fait, la philosophie a fondé la logique à la suite d'Aristote (IVe s. avant JC). Ce qui intéressait les philosophes était la validité du raisonnement humain, ainsi que la question de la vérité, deux notions que nous distinguerons ultérieurement. La logique est restée le domaine de la philosophie jusqu'au XVIIIe siècle. Ainsi Leibniz, qui était par ailleurs physicien et mathématicien, ne l'applique-t-il qu'à ses recherches en philosophie.

Ce n'est qu'au XIXe siècle que la logique s'émancipe de la philosophie avec les travaux de mathématiciens tels que Boole et Morgan, puis surtout, au tournant du XXe siècle, ceux de Frege, Hilbert, Russell et enfin Gödel. Jusqu'ici, les démonstrations mathématiques reposaient encore fréquemment sur des arguments d'autorité ou des déductions acceptées intuitivement. Il a fallu l'émergence de la logique mathématique pour que soit rigoureusement posée la question de la validité de ces démonstrations. Ces recherches fondent la logique formelle, *étude générale des raisonnements déductifs* selon le Larousse, dans lequel tout énoncé mathématique doit être démontré comme un théorème à partir d'un certain nombre d'axiomes.

La dernière grande discipline qui s'est intéressée à la logique est celle qui va nous concerner dans cet ouvrage : l'informatique. On le sait, l'ordinateur est une machine binaire qui fonctionne en ne considérant que deux types de valeurs unitaires : les bits 0 ou 1. En identifiant ces 2 valeurs à la vérité et son contraire, on peut interpréter tout calcul informatique comme un calcul logique. Mais d'une manière plus générale, la logique peut servir de système de représentation des connaissances à un ordinateur. L'informatique utilise ainsi des champs assez variés de la théorie logique. Dans tous les cas, le raisonnement qui intéresse l'informaticien est un raisonnement automatique, celui du traitement de l'information. C'est cette perspective que nous allons étudier dans cet ouvrage, tout en notant que la plupart des notions que nous aborderons font également sens pour le philosophe ou le mathématicien.

2.2. NOTIONS DE BASE : VALIDITÉ ET VERITÉ

Nous venons de présenter la logique comme une science du raisonnement. Pour la logique, un raisonnement peut être défini comme l'articulation de plusieurs jugements. Chaque jugement étant une **affirmation** à laquelle on associe une valeur de vérité unique : vrai ou faux.

> **Valeur de vérité logique** – Le sens d'un énoncé en logique se limite à sa valeur de vérité, à savoir *vrai* (noté *V*) ou *faux* (*F*).

Considérons quelques exemples d'énoncés :

- *Les oiseaux sont des animaux* proposition de valeur de vérité ***V***
- *2 est un nombre négatif* proposition de valeur de vérité ***F***

Pour ceux qui trouveraient que la logique n'a pas le sens des nuances, les logiciens ont toutefois élaboré des systèmes plus subtils tels que la logique floue qui sera étudiée dans le chapitre [REF CHAP]. Au contraire, nous étudions ici la logique classique qui est une logique binaire. Par ailleurs et dans tous les cas, les énoncés comme *Viens-tout de suite* ! ou *Paris brule-t-il ?* ne peuvent se voir associer de valeur de vérité. Ils ne seront pas considérés par les systèmes logiques que nous étudierons.

C'est sur ces énoncés dotés d'une valeur de vérité que l'on construit des raisonnements logiques, qui consistent à associer un ensemble d'hypothèses, appelées également **prémisses** du raisonnement, à une **conclusion** qui doit logiquement découler des hypothèses.

La logique a pour objectif de nous donner les moyens de savoir si un raisonnement donné est valide. Quel lien existe-t-il entre la notion de vérité, et celle, essentielle de **validité** d'un raisonnement ? Et comment peut-on définir cette notion de validité ? Pour nous aider à clarifier cette question, nous allons étudier quelques exemples de raisonnements. Considérons tout d'abord le raisonnement suivant :

Raisonnement 1	
Prémisses	*Tous les chevaux ont des oreilles*
	Les poneys sont des chevaux
Conclusion	*Donc les poneys ont des oreilles*

Ce raisonnement semble valide a priori (nous saurons le démontrer lorsque nous aurons avancé dans notre découverte de la théorie logique). On remarque que les prémisses sont toutes vraies dans une interprétation standard de l'univers, de même que la conclusion.

Un raisonnement valide correspondrait-il dès lors à sa capacité à tirer une conclusion vraie de prémisses vraies ? Avant de conclure, continuons cette étude par un second exemple :

Raisonnement 2	
Prémisses	*Tous les oiseaux volent*
	L'autruche est un oiseau
Conclusion	*Donc l'autruche vole*

La validité de ce raisonnement est déjà moins évidente, car il conduit à une conclusion fausse (l'autruche ne vole pas). Pourtant, on peut montrer que ce raisonnement est valide au sens de la logique classique. On observe que la première prémisse n'est pas vraie (tous les oiseaux ne volent pas, puisque l'autruche ou l'émeu ne savent pas voler). Le jugement de validité sur le raisonnement semble donc cohérent : d'une prémisse fausse, on arrive à une conclusion fausse.

La validité d'un raisonnement dépendrait-elle dès lors de sa capacité à propager la vérité, ou la fausseté ? L'exemple ci-dessous va nous montrer que la réalité est plus complexe.

Raisonnement 3

Prémisse	*Tout nombre pair est divisible par 2* .
Conclusion	*Donc tout nombre divisible par 2 est pair*

Dans ce cas, prémisse et conclusion sont toutes les deux vraies. Pourtant, et même si ce n'est pas évident à première vue, ce raisonnement n'est pas valide, nous saurons le démontrer plus tard.

Cherchons à nous persuader de l'incorrection du raisonnement. Pour cela, opérons une petite modification sur ce dernier : remplaçons « *nombre pair* » par « *français* » et « *divisible par 2* » par « *européen* ». Le raisonnement devient alors :

Raisonnement 4

Prémisse	*Tout français est européen* .
Conclusion	*Donc tout européen est français*

Ici, l'incorrection du raisonnement est évidente. Il semble donc le raisonnement précédent était déjà erroné, quand bien même ses prémisses et conclusions étaient vraies. Nous allons en discuter dans le paragraphe suivant. Notons déjà cette conclusion importante :

Indépendance vérité – validité

On ne peut identifier la validité d'un raisonnement à la vérité des propositions qui le composent.

2.3. POSTULAT DE LA LOGIQUE FORMELLE

Le passage que nous avons opéré entre le raisonnement (3) et le raisonnement (4) n'a rien d'anodin. Nous avons en effet changé le sens des propositions, au point que la conclusion qui était vraie pour le raisonnement (3), est erronée dans le (4). Dès lors, pourquoi peut-on affirmer qu'il s'agit du même raisonnement ? Cette affirmation repose sur un postulat fort de la logique, qui est que la validité d'un raisonnement ne dépend que de sa forme, ou structure (i.e. l'articulation de ses propositions) et non du sens des propositions qui le composent. C'est la raison pour laquelle on parle de **logique formelle**.

De fait, pour la logique formelle, les raisonnements (3) et (4) sont rigoureusement équivalents au schéma de déduction général suivant :

Raisonnement 5

Prémisse	*Tout X est Y* .
Conclusion	*Donc tout Y est X*

Le lecteur est plus prêt à accepter l'identité des raisonnements (3) et (4) une fois passé à cette forme générale. Celle-ci est au centre du postulat de la logique formelle que l'on définit ainsi :

Définition de la logique formelle

La logique formelle est l'étude de la validité des raisonnements en considérant uniquement leur forme et en faisant abstraction du contenu des propositions qui les composent.

Ce postulat fort, mais cohérent avec le sens commun, permet la mise en place d'une étude objective de la validité des raisonnements. Cette étude peut toutefois être abordée suivant deux approches que nous allons présenter maintenant.

3. APROCHE FORMELLE ET APPROCHE SEMANTIQUE

La logique formelle étudie ainsi la validité des raisonnements à partir de leur forme. Le mathématicien s'attache ainsi à la validité formelle de ses démonstrations.

A l'opposé, ce qui intéresse souvent l'informaticien est d'obtenir de nouvelles informations à partir d'informations connues. Par exemple, un programme prend en entrée des connaissances pour en déduire de nouvelles. De même, on peut vouloir obtenir une information déjà présente dans une base de données (interrogation de la base) ou même en déduire de nouvelles informations (base de données déductives). On peut ainsi distinguer deux approches de la logique, sachant que la première intéresse également certains pans de l'informatique

Approche syntaxique (formelle) – Dans ce cas, les raisonnements sont étudiés uniquement du point de vue de leur forme.

Approche sémantique – Ici, les raisonnements sont étudiés du point de vue de la propagation de la vérité ou de la fausseté entre prémisses et conclusion (cf § 2.2)

Dans l'approche sémantique, on s'intéresse donc au contenu (vrai/faux) des propositions qui composent le raisonnement. Il ne faut toutefois pas croire que cette démarche est contradictoire avec le postulat de la logique formelle. En effet, la propagation de la vérité ou de la fausseté dans un raisonnement ne dépendent, on le montrera ultérieurement, que de la forme du raisonnement.

En travaillant dans une approche sémantique, on ne se préoccupe donc pas explicitement d'aspects formels. Mais les conclusions auxquelles on arrive dans cette approche ne dépendent en fait (de manière certes masquée) que de considérations purement formelles. Nous reviendrons sur cette question à la fin de ce chapitre (§ 4.3. sur « complétude et consistance »).

Au final, ces deux approches répondent toutes deux du postulat de la logique formelle. C'est pourquoi, quelle que soit l'approche envisagée, un raisonnement devra avoir été représenté dans un langage formel, basé sur un vocabulaire, pour pouvoir être étudié. Précisons ces notions :

Vocabulaire – On appelle vocabulaire tout ensemble fini de symboles.

Langage formel – Soit un vocabulaire donné $\mathcal{V}$. On appelle langage formel $\mathcal{L}=<\mathcal{R},\mathcal{V}>$ tout ensemble, potentiellement infini, de séquences de symboles de $\mathcal{V}$ respectant des règles de bonne formation $\mathcal{R}$. Tout élément du langage est appelé **formule bien formée (fbf)** de $\mathcal{L}$.

Exemple – Soit le vocabulaire $\mathcal{V} = \{a,b\}$ et les règles de bonne formation $\mathcal{R}=\{\mathcal{R}1,\mathcal{R}2\}$ suivantes :

$\mathcal{R}1$ $\quad$ $\mathcal{F}$ est une fbf de $\mathcal{L}=<\mathcal{R},\mathcal{V}>$ si elle est formée d'un seul symbole appartenant à $\mathcal{V}$
$\mathcal{R}2$ $\quad$ $\mathcal{F}$ est une fbf de $\mathcal{L}=<\mathcal{R},\mathcal{V}>$ si $\mathcal{F} = a\mathcal{G}$ avec $\mathcal{G}$ fbf de $\mathcal{L}$

On observe par construction que le langage $\mathcal{L}$ est infini puisque la règle $\mathcal{R}2$ consiste à ajouter récursivement un *a* devant toute formule déjà bien formée : $\mathcal{L} = \{a,b,aa,ab,aaa,aab,aaaa,aaab,\ldots\}$. Si on prend comme convention que a* représente tout suite éventuellement vide du symbole a, on peut alors représenter ce langage sous la forme $\mathcal{L} = \{a^*a, a^*b\}$

Question de cours (réponses en fin d'ouvrage)

Question 1.1. Mon premier langage formel (objectif 1.1.2.)
On considère le vocabulaire fini constitué des trois lettres *a*, *b* et *c* : $\mathcal{V} = \{a,b,c\}$. Soit le langage $\mathcal{L}$ construit sur $\mathcal{V}$ à l'aide des règles de bonne formation suivantes :

$\mathcal{R}1$ $\quad$ *bac* et *abc* sont des fbf de $\mathcal{L}$

$\mathcal{R}2$ $\quad$ si $\mathcal{F}$ est une fbf de la forme x$\mathcal{G}$, avec $\mathcal{G}$ quelconque et x $\in \mathcal{V}$, alors $\mathcal{G}\boldsymbol{x}$ est une fbf de $\mathcal{L}$

a) *cba* appartient-elle au langage ?

b) montrer que *bbb* n'appartient pas au langage, à savoir qu'on ne pas construire cette formule en partant de $\mathcal{R}1$ et en appliquant la règle $\mathcal{R}2$.

c) donner l'ensemble des éléments du langage. Celui-ci est-il infini ?

Sur un langage donné, on peut définir plusieurs systèmes de raisonnement logiques qui répondent chacun à un objectif donné. Chacun de ces cadres de raisonnement est appelé *théorie logique* :

Théorie logique – On nomme théorie logique tout cadre de raisonnement spécifique construit sur un langage donné. Dans une approche syntaxique, une théorie logique, aussi appelée alors **théorie axiomatique**, est définie par un ensemble d'axiomes et de règles d'inférence. Dans une approche sémantique, elle est donnée par une interprétation particulière des éléments du langage.

Exemple – Sur un même langage formel, il est possible de construire des théories logiques différentes. Par exemple, le symbole formel d'addition + n'aura pas la même interprétation en arithmétique classique où l'on a 1+1=2, et dans le calcul booléen (que nous étudierons plus tard) où l'on a 1+1 = 1.

Dans cet ouvrage, nous n'étudierons en détail que l'approche sémantique. Nous allons toutefois survoler dans le paragraphe 4.1 des notions liées à l'approche syntaxique, pour étudier les parallèles qui peuvent exister entre ces deux démarches. Ce sera l'occasion de découvrir le vocabulaire de base qui va être utilisé tout au long de cet ouvrage, et également de retrouver avec un nouvel éclairage certaines définitions mathématiques.

3.1. APPROCHE SYNTAXIQUE DE LA LOGIQUE : THEORIE DE LA PREUVE

Dans cette approche, les raisonnements sont considérés comme la dérivation formelle de nouvelles conclusions à partir de prémisses, sans s'interroger sur la valeur de vérité de ces dernières. La notion – sémantique – de vérité est complètement ignorée ici : on admet l'existence d'un certain nombre de connaissances initiales, les axiomes, et on cherche à démontrer de nouvelles propositions logiques à partir de ces axiomes et de règles de raisonnement bien définies. Mais jamais on ne se pose de question concernant la véracité des axiomes et de leurs propositions dérivées.

Cette approche répond au besoin de formalisation des mathématiciens en termes de démonstration : c'est une raison pour laquelle cette approche est souvent nommée « *théorie de la preuve* ». Il n'est donc pas étonnant qu'ils utilisent au quotidien les notions et définitions que nous allons présenter :

Axiome – Un axiome est une formule logique qui est choisie a priori.

Un axiome T se note : $\vdash T$

Règle d'inférence – Une règle d'inférence est un schéma de raisonnement minimal (i.e. non décomposable en raisonnements plus simples) qui permet de déduire formellement de nouvelles formules C à partir de formules Pi qui sont des axiomes ou qui ont déjà été démontrées par ailleurs. On dit que la conclusion est la **conséquence formelle** (ou **syntaxique**) des prémisses.

On note : $P1, P2, \ldots, Pn \vdash C$

Dans une telle règle, les Pi sont appelées **prémisses** ou encore **hypothèses**, tandis que C est appelée **conclusion**.

Exemple : le *modus ponens* (également appelé *détachement*) est une règle d'inférence qui met en jeu l'implication. Très intuitif, le schéma de raisonnement qui lui correspond est le suivant :

Modus ponens		
Prémisses	(P1)	*Si A alors B*
	(P2)	*A* .
Conséquence	(C)	$\vdash$ *B*

Le *modus ponens* dit ainsi que si A est déjà établi (prémisse P2) et que l'on a déjà établi que si A est démontré alors on peut démontrer B (prémisse P1), alors on démontre par conséquence formelle la formule B. Remarquons à cette occasion que cette règle d'inférence ne nous dit rien du contenu de A et B : elle s'applique ainsi à tout raisonnement qui suit ce schéma. Par exemple :

(P1) *S'il fait beau alors je vais à la plage*
(P2) *Il fait beau*

(C) *Je vais à la plage*

Avec ici les propositions logiques : A = *il faut beau* et B = *je vais à la plage*

Nous démontrerons la validité de ce schéma minimal de raisonnement (au sens sémantique du terme) dans le cadre de l'étude de la logique des propositions [chapitre 2]. Dans l'immédiat, continuons notre parcours des définitions classiquement utilisées dans une approche purement formelle de la logique.

Théorème – On appelle théorème toute formule logique qui est, soit un axiome, soit le résultat d'une application successive de plusieurs règles d'inférences (démonstration) à partir d'un ensemble initial d'axiomes (prémisses). Lorsqu'il a ainsi été démontré, un théorème se note $\vdash T$

Démonstration – Une démonstration formelle est l'application successive de plusieurs règles d'inférences à partir d'un ensemble d'axiomes ou de théorèmes déjà démontrés. Elle produit systématiquement de nouveaux théorèmes qui sont la conséquence formelle de ses prémisses.

Pour aller plus loin : syllogismes et pensée logique antique

L'étude des raisonnements logiques a été initiée dès l'Antiquité par le philosophe Aristote (384 – 322 av. JC). C'est en particulier lui qui a le premier cherché à recenser un certain nombre de figures basiques de raisonnements et à juger de leur validité. Il a ainsi défini le terme de syllogisme (συλλογισμός en grec), qui peut être traduit littéralement par « parole qui va avec une autre ». On retrouve ici l'idée d'affirmations mises en correspondance que nous venons de découvrir. Les schémas de raisonnement qui ont intéressé Aristote se composaient en effet toujours de deux prémisses (prémisse majeure et mineure) et d'une conclusion, suivant un exemple bien connu :

Tous les hommes sont mortels	(prémisse majeure)
Socrate est un homme	(prémisse mineure)
Socrate est mortel	(conclusion)

Tout au long de l'antiquité, et ensuite jusqu'aux Temps Modernes, les évolutions de la théorie logique seront limitées : elles consisteront en effet avant tout à découvrir de nouvelles figures de raisonnement, à les classifier et à les comparer entre elles.

Ainsi, le *modus ponens*, forme de déduction des plus simples et souvent classée avec les syllogismes hypothétiques, a-t-il progressivement émergé de la pensée antique, en particulier chez les philosophes stoïciens.

Pour en savoir plus – Susanne Bobzien (2002). The Development of Modus Ponens in Antiquity, *Phronesis* 47(4), 359-394.

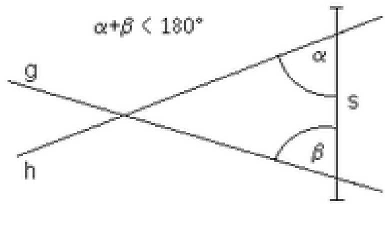
Remarque : théorème et vérité – Les définitions que nous avons données ci-dessus ignorent la question de la vérité des prémisses ou des conclusions : ce qui importe est la cohérence formelle de la démonstration. Cette remarque concerne également les axiomes : rien n'impose que ces axiomes et ses théorèmes soient des vérités. Les mathématiques se sont ainsi emparées très tôt de la possibilité de considérer des axiomes ou théorèmes dont la vérité n'est pas établie.

Les géométries non euclidiennes sont un bel exemple de modèle mathématique dans lequel les théorèmes ne sont pas vrais (du moins dans notre représentation courante du monde). Lorsqu'il a proposé ses *Eléments*, le mathématicien grec Euclide a eu besoin de cinq axiomes. De tout temps, les mathématiciens se sont demandés si le dernier de ces axiomes n'était pas en fait une conséquence formelle des autres. Il s'agit de l'axiome des parallèles, qui peut être énoncé ainsi :

« *Par un point donné, on peut mener une et une seule parallèle à une droite donnée* »

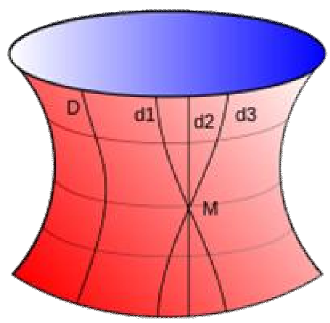
Pour aller plus loin : le vrai axiome d'Euclide

L'axiome des parallèles d'Euclide n'a pas été formulé sous la forme qui est enseignée à tout écolier. Euclide avait en effet affirmé que : « *si une droite coupe deux autres droites en déterminant deux angles internes dont la somme soit différente de deux angles droits, alors les deux droites se coupent dans le demi-plan pour lequel la somme est inférieure à deux angles droits* » (voir figure).



C'est cette formulation complexe qui a fait croire que l'énoncé pouvait être déduit des autres axiomes euclidiens. C'est en cherchant à le démontrer que le mathématicien Proclus de Lycie a montré au Ve siècle après J.C. qu'il était équivalent à l'axiome plus simple que nous connaissons.

Il a fallu près de deux millénaires pour démontrer que ce postulat était vraiment une proposition axiomatique non démontrable. Ces travaux ont toutefois incité Carl Friedrich Gauss et Nikolaï Lobatchevski à s'interroger sur le développement d'une géométrie excluant cet axiome. Il en a résulté le développement de géométries dites non euclidiennes parmi lesquelles les géométriques hyperboliques et sphériques dans lesquelles l'espace est courbe. Sur la figure ci-contre, les « droites » D1, D2 et D3 sont parallèles à la « droite » D tout en étant plusieurs à passer par le point M !



Une telle situation ne se rencontre jamais au quotidien dans notre perception géométrique de notre environnement habituel. Les géométries non euclidiennes proposent donc des théorèmes qui ne sont pas vrais, dans notre expérience classique du monde réel. Pourtant, les géométries non euclidiennes reprennent toute leur pertinence dans le cadre de la théorie générale de la relativité énoncée par Albert Einstein, dont une conséquence est la courbure de l'espace-temps.

Ainsi, nous pouvons retenir qu'un axiome ou un théorème (au sens logique du terme) ne sont pas nécessairement des vérités du sens commun. Ce sont deux notions a priori indépendantes : nous étudierons au cours du paragraphe 4.3 les rapports qu'entretiennent théorème et vérité.

3.2. APPROCHE SEMANTIQUE DE LA LOGIQUE : THEORIE DES MODÈLES

L'approche sémantique s'intéresse à la valeur de vérité des énoncés logiques et à leur propagation de la vérité en se basant uniquement sur la forme des raisonnements. C'est sur l'analyse de cette propagation qu'est fondée l'approche sémantique, qui met en jeu les concepts suivants :

Sens – Le sens d'un énoncé logique se réduit strictement à sa valeur de vérité : vrai (noté V) ou faux (noté F).

Equivalence logique – Deux énoncés A et B sont dits équivalents s'ils ont le même sens logique (donc s'ils ont toujours la même valeur de vérité quelle que soit l'interprétation que l'on considère). On note : $A \equiv B$

Exemples P = *La capitale de la France est Lyon* SENS(P) = $\mathcal{F}$, soit P ≡ $\mathcal{F}$
P = *La capitale des Gaules était Lyon* SENS(P) = $\mathcal{V}$, soit P ≡ $\mathcal{V}$

On le voit, la notion de sens est bien plus restrictive en logique que dans le langage courant, puisqu'elle se limite à une valeur binaire : la valeur de vérité *vrai* ou *faux*. Cette valeur de vérité est indépendante de la forme des énoncés. C'est ainsi que les quatre énoncés ci-dessous ont rigoureusement la même valeur de vérité. Ils sont donc équivalents logiquement :

La France a pour capitale Paris	$P \equiv \mathcal{V}$
La Ville Lumière est la capitale de la France	$P \equiv \mathcal{V}$
The capital of France is the well-known city of Paris	$P \equiv \mathcal{V}$
Paris ist die Hauptstadt von Frankreich	$P \equiv \mathcal{V}$

On voit qu'ici, la logique ne fait pas dans la nuance ! Autre exemple de l'indépendance entre forme et sens en logique, un énoncé d'une forme donnée peut au contraire avoir plusieurs sens. Prenons l'exemple suivant :

L'homme est le lointain descendant d'une bactérie

Le sens de cet énoncé sera vrai pour un biologiste, au vu de la théorie de l'évolution. Alors qu'il est faux dans le cas d'une lecture littérale de la Bible, du Coran ou de la Torah. On dit que l'on a deux **interprétations** différentes du même énoncé. Formellement, on définit ainsi la notion d'interprétation.

Interprétation – Soit L l'ensemble des énoncés correspondant à un système logique donné. On appelle interprétation sur L toute application I associant une valeur de vérité vraie ou fausse aux énoncés de L :

$$\mathsf{I} : \mathcal{L} \rightarrow \{\mathcal{V},\mathcal{F}\}$$
$$\Phi \quad \mathsf{I}(\Phi)$$

$\mathsf{I}(\Phi)$ est le sens de Φ dans l'interprétation I.

Modèle et satisfaisabilité – On dit qu'une interprétation I est un **modèle** pour une formule Φ si elle rend vraie cette dernière : $I(\Phi)$. On dit par ailleurs qu'une formule Φ est **satisfaisable** si elle admet au moins un modèle.

Considérons ce nouvel énoncé : P = *Au niveau de la mer, l'eau bout à 100e degré*

L'énoncé P est faux pour un anglais, pour qui l'eau bout à 212 degré … Fahrenheit, tandis qu'elle est vraie pour à peu près tout autre européen qui apprend, à la suite du Système International d'Unités (SI en abrégé), que l'eau entre en ébullition à 100 degrés … Celsius. L'énoncé est donc satisfaisable, puisque nous lui avons trouvé un modèle : l'interprétation SI. Par rapport à ce que nous avons étudié plus haut, on pourrait de même construire un exemple d'énoncé satisfaisable qui est vrai en géométrie hyperbolique et faux en géométrie euclidienne (par exemple « la somme des angles d'un triangle est toujours égale à 180 degrés) ou inversement.

Au vu de ces exemples, il pourrait paraître évident qu'un énoncé logique est toujours satisfaisable : il suffirait de trouver une interprétation qui fonctionne bien. Il n'en est rien, et certains énoncés ont ainsi la propriété remarquable d'être toujours vrais (ou toujours faux) quelle que soit l'interprétation que l'on puisse imaginer.

Tautologie – Un énoncé T est une tautologie (ou une vérité universelle) si et seulement s'il est vrai pour toute interprétation.

On le note : $\models T$

Contradiction – A l'opposé, un énoncé C est dit contradictoire (ou **insatisfaisable**) si et seulement s'il est faux pour toute interprétation.

Il est assez facile de construire de tels énoncés :

- *La pièce est sur le côté pile et le côté face* : contradiction, on ne peut être à la fois quelque chose et son contraire (si votre pièce n'appartient pas à un tricheur !).
- *La pièce est sur le côté pile ou le côté face* : tautologie, on a quelque chose ou son contraire.

Les définitions que nous venons de découvrir concernaient le sens des énoncés logiques. Intéressons-nous désormais aux raisonnements construits sur ces énoncés.

> **Conséquence sémantique** – Soient deux énoncés logiques A et B. On dit que B est la conséquence sémantique de A si et seulement si tout modèle de A est un modèle de B. En d'autres termes, si B est vrai pour toute interprétation qui rend A vraie.
>
> On note : $A \models B$

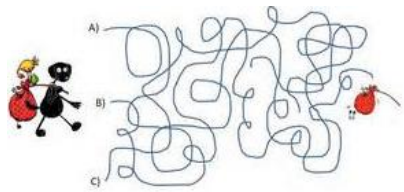
Remarque – On notera le rôle central de la notion de modèle dans cette définition sémantique de la déduction logique. C'est pourquoi l'approche sémantique est souvent dénommée sous le terme de théorie des modèles.

> **Raisonnement valide** – Soit un raisonnement formé d'un ensemble de prémisses P1, …, Pn et d'une conclusion C. On dit que le raisonnement est valide si et seulement si C est la conséquence sémantique des prémisses.
>
> On note : $P1, \ldots, Pn \models C$

Question de cours (réponses en fin d'ouvrage)

Question 1.2. B.A.BA logique (objectif 1.1.3 & 1.1.4)

Sauras-tu aider notre amie la fourmi à relier les termes présentés à gauche avec les définitions données sur la droite ?



modèle	☐	☐	proposition logique admise a priori
tautologie	☐	☐	énoncé faux pour toute interprétation
axiome	☐	☐	énoncé vrai pour toute interprétation
contradiction	☐	☐	interprétation qui rend vrai un énoncé logique
satisfaisable	☐	☐	proposition démontrée par raisonnement formel
théorème	☐	☐	énoncé vrai au moins pour une interprétation

Remarque – En d'autres termes, un raisonnement est valide si et seulement si sa conclusion est vraie à chaque fois qu'une interprétation rend vraie l'ensemble de ses prémisses : on retrouve ici la contrainte de propagation de la vérité qui avait été étudiée en introduction (cf § 2.2).

Notons bien que la validité d'un raisonnement ne dépend pas du choix d'une interprétation particulière : comme le suggère cette définition, l'approche sémantique devra considérer **toutes** les interprétations imaginables pour vérifier la validité d'un raisonnement.

3.3. APPROCHES FORMELLE ET SEMANTIQUE : CONSISTANCE ET COMPLETUDE[1]

Nous avons découvert deux approches pour étudier une théorie logique : la théorie de la preuve (approche formelle ou syntaxique) regarde exclusivement la forme des énoncés, tandis que la théorie des modèles (approche sémantique) s'intéresse à leur contenu. Les logiciens se sont interrogés dès le début du XXe siècle sur les correspondances entre ces deux démarches. En particulier, ils se sont demandé si les « bonnes propriétés » que l'on peut associer à une théorie logique sont équivalentes dans les deux approches. Deux propriétés ont en particulier retenu leur attention : la consistance et la complétude :

- La **consistance** va nous permettre de déterminer si une théorie logique forme un système de raisonnement cohérent, i.e. sans contradiction
- La **complétude** va de son côté nous assurer qu'une théorie logique offre un système de raisonnement qui couvre l'ensemble de la réalité qu'elle se propose de décrire.

[1] Les paragraphes 3.3, 4.1 et 4.2 de ce chapitre font appel à des notions plus complexes qui peuvent être omises dans un premier temps (**hors programme L1)**: le lecteur intéressé pourra y revenir après avoir lu les chapitres 2 et 3.

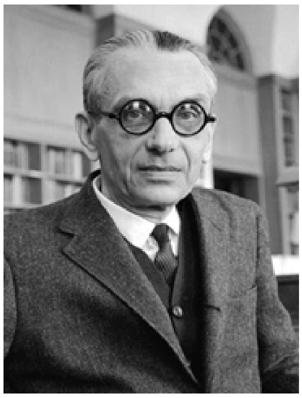
Ces notions sont parfois difficiles à appréhender car leur définition varie suivant l'approche envisagée. Ainsi, on distingue la consistance syntaxique de la consistante sémantique :

Consistance sémantique – Dans l'approche sémantique, on dit qu'une théorie logique est consistante (ou encore **satisfaisable**) si et seulement si elle possède au moins un modèle. Dans le cas contraire, la théorie est dite inconsistante. Il va de soi qu'une théorie inconsistante est considérée comme de peu d'intérêt

Consistance syntaxique – Dans l'approche syntaxique, on dit qu'une théorie logique est consistante (ou encore **cohérente** ou **non contradictoire**) s'il n'existe pas de formule de la théorie que l'on peut démontrer en même temps que sa négation[2]. A l'opposé, dans une théorie incohérente, on peut montrer tout et son contraire et toute formule est un théorème.

Deux définitions pour une même notion, voilà qui était un peu difficile à comprendre. Le mathématicien allemand Kurt Gödel, une des figures centrales de l'étude de la théorie logique (voir encart), a précisément étudié les rapports entre ces deux formes de consistance. Ainsi, il a montré que les deux notions coïncident dans la logique des prédicats du premier ordre. Cette logique, essentielle en informatique, sera étudiée plus loin dans ce cours.

Pour aller plus loin : Kurt Gödel, génie et bourreau de la théorie logique



Kurt Gödel (1906-1978) est un mathématicien et logicien né à Brno en 1906, alors en Autriche-Hongrie, naturalisé Tchécoslovaque à 12 ans, puis autrichien à 23 ans, qui devient allemand à 32 ans et obtient la double nationalité austro-américaine à 42 ans, après avoir quitté l'Allemagne nazie.

Sa thèse et un article publié en 1931 lui donnent une réputation internationale et mettent fin aux espoirs de Hilbert d'axiomatiser complètement les mathématiques. Il travaille ensuite à la théorie des ensembles puis il s'intéresse à la relativité, en relation avec son ami Einstein. Il s'éloigne du monde scientifique à la fin des années 1950.

Gödel a connu des successions d'épisodes dépressifs dès 1930. Il devient progressivement hypocondriaque et paranoïaque. Persuadé que l'on cherche à l'empoisonner, il refuse presque totalement de s'alimenter. Il en meurt le 14 janvier 1978.

Si ce premier résultat publié par Gödel avait de quoi rassurer les mathématiciens, il en fut tout autrement de ses deux célèbres théorèmes dits d'incomplétude qui ruinèrent les espoirs du mathématicien David Hilbert. Ce dernier en effet, l'un des pères de la logique moderne, était convaincu qu'il devait être possible de toujours prouver la cohérence ou l'incohérence de n'importe quel système formel, capacité que l'on qualifie de complétude syntaxique.

Complétude syntaxique (ou **complétude forte**) – Une théorie logique (axiomatique) est fortement complète si et seulement si, pour toute formule Φ de la théorie, soit il existe une preuve de Φ, soit ajouter Φ aux axiomes rend la théorie inconsistante.

La complétude syntaxique est une propriété très importante puisqu'elle traduit le fait que les formules que l'on ne peut pas démontrer sont en contradiction avec l'ensemble des axiomes. Ce que ne savait pas alors Hilbert, c'est qu'il s'agissait malheureusement d'une propriété que ne possèdent que des systèmes au pouvoir d'expression très limité, tels que la logique des propositions (la démonstration de la complétude forte du calcul des propositions n'est pas simple et elle ne sera pas abordée dans cet ouvrage). A l'opposé, dans son premier théorème, Gödel

[2] Ceci signifie également que dans une théorie cohérente, certaines formules ne sont pas prouvables. En effet, si toutes les formules étaient prouvables, alors leur négation le serait aussi, ce qui va à l'encontre de la définition.

montre que tout système assez puissant pour modéliser l'arithmétique (l'ensemble des entiers naturels muni de ses opérations) est incapable de prouver sa propre consistance.

Les conséquences du second théorème d'incomplétude de Gödel, prouvé en 1931, sont encore plus troublantes et elles ont encore davantage ébranlé les convictions des mathématiciens formalistes tels que Hilbert. Ce théorème fait en effet appel à la notion de complétude sémantique, et questionne la notion de la prouvabilité des vérités.

Complétude sémantique ou **complétude faible** – Soit une théorie logique T dont on peut donner une définition axiomatique ou sémantique. T est sémantiquement complète (on dit aussi faiblement complète) si et seulement si toute tautologie (approche sémantique) est démontrable (approche syntaxique) dans T.

La complétude sémantique[3] traduit le fait que toutes les formules vraies peuvent être démontrées. Elle permet d'identifier le système formel à son modèle. Considérons maintenant une théorie sémantiquement incomplète. Cela signifie que l'on a identifié au moins une formule F vraie mais qu'on ne peut pas démontrer. Suivant une démarche suggérée par David Hilbert, on peut supposer qu'en rajoutant cette formule dans les axiomes, le problème soit résolu et reproduire ainsi le processus avec toute autre formule se trouvant dans une situation analogue. Malheureusement, il s'avère que ce processus de complétion ne peut fonctionner dans tous les cas. Le **second théorème d'incomplétude de Gödel** nous dit en effet que, dans toute théorie axiomatique consistante syntaxiquement et capable de modéliser l'arithmétique, on pourra toujours construire une formule vraie que l'on ne peut pas démontrer. Plus précisément, Gödel établit clairement que la démarche suggérée par Hilbert n'est pas une solution dans la mesure où la théorie complétée serait elle-même incomplète et il faudrait donc répéter l'opération de complétion à l'infini. Il existe donc une différence entre prouvabilité et vérité, alors que Hilbert voulait obtenir des preuves syntaxiques pour éliminer le recours à l'intuition qui intervient toujours dans les preuves sémantiques.

Hilbert ou le rêve brisé

David Hilbert (1862-1943), mathématicien allemand, est un des plus grands noms de l'histoire de la logique, au sens où il a longtemps porté le flambeau d'une théorie logique du Grand Tout, i.e. explicative de toute vérité, qu'elle soit mathématique ou autre.

Figure emblématique des mathématiques du XXième siècle, Hilbert a durablement influencé les recherches dans sa discipline, en particulier par ses 23 problèmes, proposés à la communauté des mathématiciens en 1900 et dont 3 sont encore ouverts. Ainsi, avant que les persécutions nazies ne démantèlent son équipe, il avait fait de l'Université de Göttingen un centre mondial des mathématiques. Sa volonté de comprendre et d'expliquer est résumée dans sa célèbre formule qui lui sert d'épitaphe : *« Wir müssen wissen, wir werden wissen »* (*« nous devons savoir, nous allons savoir »*).

Malgré tout, les deux premières théories logiques que nous étudierons dans cet ouvrage (la logique des propositions et la logique des prédicats du premier ordre) sont des théories sémantiquement complètes : la complétude faible du calcul des prédicats est un résultat qui a été démontré par Gödel (encore !) dans sa thèse de doctorat en 1929.

[3] Lorsque le mot complétude est utilisé sans précision, c'est de complétude sémantique dont il est question

En dépit de son caractère affaibli, la complétude sémantique est importante au sens où elle relie les deux approches sémantique et syntaxique sur ce qui intéressera le plus l'informaticien : les énoncés logiques porteurs de vérité, donc d'information utile.

Pour qu'une théorie logique soit utile en informatique, il faut pourtant qu'elle possède une autre propriété : il est nécessaire qu'il soit possible de construire un algorithme qui permette de prouver en toute situation une formule (ou sa négation). Cette propriété est couverte par la notion de décidabilité, qui là encore relève de plusieurs définitions.

Question de cours (réponses en fin d'ouvrage)

Question 1.3. Peut-on réellement croire un prof de maths ? (objectif 1.1.6)

Tout au long de votre scolarité, vos enseignants de mathématiques vous ont causé de théorèmes. Mais en utilisant cette notion, ne vous aurait-on pas menti par omission ? Que pensez-vous en particulier des assertions suivantes :

1 — *Pour être un théorème il suffit qu'un énoncé mathématique soit toujours vrai (tautologie)*

☐ vrai ☐ faux

2 — *Un théorème peut être faux*

☐ vrai ☐ faux

3 — *Tout théorème est une tautologie*

☐ vrai ☐ faux

4 — *Une tautologie n'est pas forcément un théorème*

☐ vrai ☐ faux

4. DÉCIDABILITÉ

Il convient en effet de distinguer deux types différents de décidabilité : la décidabilité logique qui s'applique aux formules d'une théorie, et la décidabilité algorithmique qui s'applique aux théories ou aux problèmes. Cette seconde notion n'est pas propre à la logique mais concerne toute classe de problèmes informatiques.

4.1. DÉCIDABILITÉ LOGIQUE D'UN ÉNONCÉ

La notion de décidabilité logique a été étudiée avant tout dans une approche formelle (théorie de la preuve), même si sa définition, donnée ci-dessous, peut facilement être étendue à la théorie des modèles.

Décidabilité logique – Un énoncé est (logiquement) décidable dans une théorie axiomatique si on peut le démontrer ou démontrer sa négation dans le cadre de cette théorie.

De même, un énoncé est indécidable dans une théorie mathématique si l'on ne peut pas le déduire de cette théorie, non plus que sa négation, à partir des axiomes posés. L'énoncé est alors dit aussi indépendant du système d'axiome.

4.2. DÉCIDABILITÉ ALGORITHMIQUE

La notion de décidabilité algorithmique est intimement liée à la théorie informatique puisqu'elle pose la question des fonctions calculables par une machine de Turing. Il semble important d'en parler dans cet ouvrage, tout d'abord pour bien distinguer décidabilité algorithmique (informatique) et décidabilité logique (mathématiques). Mais également parce que nombre de problèmes formalisés sous forme logique vont donner lieu à une implémentation informatique.

Avant d'aller plus loin, il faut rappeler qu'un ordinateur actuel est une machine de Turing, c'est-à-dire que ses capacités sont équivalentes à un modèle abstrait d'automate programmable défini par ce grand mathématicien anglais en 1936. Tout algorithme mis en œuvre par un ordinateur peut ainsi être représenté dans une machine de Turing, ainsi que l'avait montré Church. Tout problème informatique peut alors être formalisé suivant un problème de décision au sens de Turing, à savoir un problème auquel on peut répondre par oui ou par non. Par exemple, créer un programme qui calcule la racine carrée de 4 pour donner 2 en résultat correspond au problème de décision suivant : "*2 est-il oui ou non la racine carrée de 4 ?*". Bien que cela soit moins évident à première vue, il en serait de même pour tout autre problème tel que, par exemple, l'affichage d'une page Web sur un navigateur !

La décidabilité algorithmique va justement consister à se demander si, pour un problème donné, il est possible à un automate programmable tel qu'un ordinateur de répondre à la question en un temps fini. Et ceci, quels que soient les talents du programmeur qui a imaginé un algorithme pour répondre au problème. On la définit comme suit :

Décidabilité algorithmique – Un problème de décision est dit décidable s'il existe un algorithme qui se termine en un nombre fini d'étapes et qui répond par oui ou par non à la question posée par le problème. Si un tel algorithme n'existe pas, le problème est dit indécidable.

On voit aisément que cette définition concerne également la théorie logique et la décidabilité logique. On peut ainsi se demander si, pour une théorie axiomatique donnée, il existe une procédure effective qui prend en entrée une formule du système et qui rend en sortie en un temps fini une réponse positive ou négative à la question "*la formule est-elle un théorème du système*". Etant donné qu'une démonstration logique est toujours finie, la décidabilité logique d'une théorie implique sa décidabilité algorithmique. Nous avons vu par ailleurs qu'un système syntaxiquement complet est décidable logiquement. Dès lors, il vient :

Complétude et décidabilité – Une théorie logique syntaxiquement complète est décidable (logiquement et algorithmiquement). En effet, étant donné un énoncé, énumérer les axiomes et les preuves est un algorithme qui permet de répondre "oui, cet énoncé est prouvable" ou "non, cet énoncé n'est pas prouvable". Il est connu sous le nom d'algorithme du British Museum[4].

Si un système est syntaxiquement incomplet, on n'est par contre plus du tout sûr que la procédure de décision sur une formule F se termine. La non-terminaison ne peut par ailleurs pas servir de conclusion : si la procédure ne se termine pas, on ne saura en effet jamais si c'est sa preuve est très très longue et s"il faut encore attendre. D'une manière plus générale, un problème qui ne se termine pas quelle que soit la réponse attendue est dit **indécidable**. Certaines situations sont légèrement plus favorables, mais ne permettent pas toujours de conclure :

Semi-décidabilité algorithmique – Un problème de décision est dit semi-décidable s'il existe un algorithme qui se termine en un nombre fini d'étapes lorsqu'il doit répondre par oui, mais dont ne connait pas la réponse quant à l'arrêt de cet algorithme dans tous les cas (i.e. en cas de réponse négative).

Dans le chapitre 3 ce cet ouvrage, nous étudierons précisément une théorie logique déjà puissante, la logique des prédicats du premier ordre, qui est semi-décidable. En pratique, un tel système logique ne peut donner lieu à une implémentation informatique efficace. Nous verrons heureusement que certains fragments de cette théorie (la logique monadique du 1[er] ordre, la

[4] Ce nom d'algorithme, étrange pour une production informatique, a été attribué par les trois informaticiens Allen Newell, Herbert A. Simon et Cliff Shaw qui déclare à son sujet : "*it seemed* (…) *as sensible as placing monkeys in front of typewriters in order to reproduce all the books in the British Museum*".

logique des clauses de Horn) sont décidables. Un langage de programmation logique que nous étudierons, Prolog, s'appuie précisément sur cette restriction aux clauses de Horn.

5. PLAN DE L'OUVRAGE

Cet ouvrage comporte quatre parties successives. Les deux premières sont consacrées à la présentation de deux modèles logiques de représentation, qui pourraient être aussi bien étudiés en mathématiques qu'ici. La première, consacrée à la **logique des propositions** (abrégée en **LP0**), nous permettra de découvrir les notions de base liées à la modélisation logique, au calcul de la vérité d'énoncés appelés propositions, mais aussi de la validité de raisonnements basés sur ses propositions. Nous en profiterons pour découvrir l'intérêt d'une logique équivalente à la LP0 (**logique booléenne**) pour la conception électronique des systèmes informatiques.

Nous verrons toutefois que la logique des propositions est insuffisante pour répondre complètement aux besoins de la formalisation de problèmes informatiques. Ceci nous amènera à étudier dans la partie suivante la **logique des prédicats du premier ordre** (abrégée en **LP1**), qui étend la logique des propositions tout en l'englobant.

Bien d'autres modèles logiques existent, comme les logiques d'ordre supérieures. Ils dépassent le propos de cet ouvrage, mais nous étudierons tout de même le formalisme de la logique floue pour lequel la distinction vérité/fausseté n'est plus binaire mais progressive afin de rendre compte d'une incertitude sur cette décision. Nous les ignorerons pour aborder dans les parties suivantes deux applications directes de la théorie logique à l'informatique :

- La **programmation logique**, qui sera étudiée avec le langage Prolog comme support. Nous nous contenterons ici d'une présentation du noyau logique du langage Prolog (appelé parfois Prolog pur).
- Les **bases de données**, dont nous donnerons ici une interprétation logique qui est équivalente à l'algèbre relationnelle et qui est souvent employée comme porte d'entrée à la modélisation des bases de données. Cette partie tissera précisément des parallèles entre modélisation logique ou relationnelle des bases de données.

Introduction - Exercices & Problèmes

EXERCICES

Exercice 1.1 — Mon premier langage formel (objectif 1.1.2)

On considère le langage formel $\mathcal{L} = \{\mathcal{V},\mathcal{R}\}$ défini sur le vocabulaire $\mathcal{V}$ = {a,b,c,d,e,f,g,h,i,j,k,l, m,n,o,p,q,r,s,t,u,v,w,x,y,z} avec R défini comme suit :

R1 : si $x \in \mathcal{V}$ alors $F = x\text{a}$, $G = x\text{i}$ et $H = x\text{o}$ sont des fbf $\in \mathcal{L}$
R2 : si $F \in \mathcal{L}$, $x \in \mathcal{L}$ et $y \in \{\text{a},\text{i}\}$ alors $G = xyF$ est un fbf $\in \mathcal{L}$

1. Donnez deux exemples de formules bien formées de $\mathcal{L}$ comportant respectivement 2 et 4 symboles de $\mathcal{V}$.

2. Dire si les formules ci-dessous sont ou non des formules bien formées du langage $\mathcal{L}$

a	☐ oui	☐ non	☐ on ne peut dire
harakiri	☐ oui	☐ non	☐ on ne peut dire
yamamoto	☐ oui	☐ non	☐ on ne peut dire
diniz	☐ oui	☐ non	☐ on ne peut dire

Exercice 1.2 — Langages formels : Apapou (objectif 1.1.2)

On considère un langage formel $\mathcal{L} = \langle\mathcal{V},\mathcal{R}\rangle$ formé sur le vocabulaire $\mathcal{V}$ = {a,o,p,u} et les règles $\mathcal{R}$ de bonne formation suivantes :

R1 $\mathcal{F}$ est une fbf de $\mathcal{L}$ si $\mathcal{F}$ = pa
R2 $\mathcal{F}$ est une fbf de $\mathcal{L}$ si $\mathcal{F}$ = pou
R3 $\mathcal{F}$ est une fbf de $\mathcal{L}$ si $\mathcal{F}$ = pou$\mathcal{G}$pa avec $\mathcal{G}$ est une fbf de $\mathcal{L}$

Question. Donnez deux exemples de fbf de $\mathcal{L}$ composés de plus de 9 symboles de $\mathcal{V}$

Réponses
Un exemple de réponse parmi d'autres : poupoupapapa ou poupoupoupapa

Exercice 1.3 — Théorie de la preuve (objectif 1.1.3)

Cet ouvrage ne s'attardera pas sur les approches syntaxiques de la logique. Cet exercice vous donne toutefois une idée de cette démarche. On considère, dans une approche syntaxique, une théorie logique définie à partir du quadruplet S = < $\mathcal{V}$, $\mathcal{L}$, A, R > suivant :

- Vocabulaire $\mathcal{V}$ = { i, m, e } ;

- Langage $\mathcal{L}$ = chaînes de caractères quelconques construites sur V.
- Axiomes $\vdash$ `Xme` où `X` représente une chaîne éventuellement vide de `i`.
- Règles d'inférences (1) `PmQeR` $\vdash$ `PmiQeQR` (avec P, Q, R chaînes de `i`)
  (2) `PmQeR` $\vdash$ `PimQePR`

1. Les chaînes suivant appartiennent-elles au langage ?

1) `mie` 2) `iii` 3) `m`

2. A-t-on $\vdash$ `iiimieii` ?

3. Y-a-t-il plusieurs démonstrations de $\vdash$ `imiiei` ?

Réponses

1. oui puisque toute chaîne de caractères construite sur {i,m,e} est une fbf.
2. $\vdash$ `iime` est un axiome et on a $\vdash$ `iime` $\vdash$ `iimie` $\vdash$ `iiimieii`
3. On a deux démonstrations $\vdash$ `me` $\vdash$ `mie` $\vdash$ `miiei` $\vdash$ `imiiei` et $\vdash$ `me` $\vdash$ `ime` $\vdash$ `imie` $\vdash$ `imiiei`

PROBLEMES

Problème 1.1 — Théorie de la preuve (objectif 1.1.3)

On considère un langage $\mathcal{L}$ formé sur un vocabulaire formé des quatre lettres `a`,`b`,`c` ou `d`. $\mathcal{L}$ est l'ensemble des formules de la forme `FdG` où `F` et `G` sont des composantes définies comme suit :

- `F` est une composante si elle correspond à une lettre `a` ou `b`
- `F` est une composante si elle est de la forme `Gca` ou `Gcb` avec `G` elle-même composante.

Question 1. Formules bien formées du langage – Les séquences de lettres qui appartiennent au langage sont appelée formules bien formées. Dire si les suites de lettres suivantes sont ou non des formules bien formées du langage $\mathcal{L}$:

1) `aca` 2) `adbca` 3) `adcb` 4) `adbcada` 5) `bcbdbcbcb`

Question 2. Théorèmes d'une théorie logique basée sur le langage – On définit une théorie logique sur ce langage à l'aide des axiomes et des règles de déduction suivantes :

Axiomes $\vdash$ `ada` et $\vdash$ `bdb`

Règle d'inférence $F_i \vdash F_{i+1}$ avec F_i formule bien formée de $\mathcal{L}$ et F_{i+1} une formule bien formée obtenue à partir de F_i en remplaçant dans cette formule :

- une occurrence de `a` par `aca`, `acb` ou `bca`.
- ou bien une occurrence de `b` par `bcb`

2.1. Démontrer que `bcbcada` et `bcbcadacbcb` sont des théorèmes de la théorie.

2.2. Montrer que `bcbdacb` n'est pas un théorème.

Question 3. Approches syntaxiques et sémantiques : interprétations de la théorie – On considère l'interprétation suivante de cette théorie logique :

a → 0 b → 1 c → * d → =

3.1. Donnez l'interprétation des axiomes et de la règle d'inférence dans ce cas.

3.2. Montrez que tous les théorèmes de la théorie sont vrais (au sens de l'arithmétique) dans cette interprétation.

3.3. Réciproquement, peut-on assurer que toute formule vraie appartenant au langage de la forme est un théorème du système ?

3.4. Pouvez-vous préciser, en justifiant votre réponse, si cette théorie est syntaxiquement complète et si elle est décidable logiquement.

3.5. Mêmes questions avec l'interprétation suivante :

$a \rightarrow 0 \qquad b \rightarrow 1 \qquad c \rightarrow * \qquad d \rightarrow \leq$

Réponses

1. non, oui, non, non, oui

2.1. ⊢ `ada` ⊢ `bcada` ⊢ `bcbcada` (formule 1) ⊢ `bcbcadacb` ⊢ `bcbcadacbcb` (formule 2)

2.2. `bcbdacb` n'est pas un théorème car `bcb` à gauche de `d` ne peut s'obtenir qu'en remplaçant un `b` par `bcb`, `acb` à droite de `d` ne peut provenir que d'un remplacement d'un `a` par `acb`. La formule initiale serait alors `bda` qui n'est pas un théorème.

3.1. Les axiomes deviennent 0=0 et 1=1. La règle d'inférence revient toujours à remplacer 0 par une valeur nulle.

3.2. Les axiomes sont vrais (0=0 et 1=1) et remplacer 0 par 0*1, 0*0 ou 1*0 ne change pas la valeur des entiers dont l'égalité est testée. Donc tous les théorèmes sont vrais dans cette interprétation

3.3. Par ailleurs, si l'on a une formule de la forme voulue, elle exprime que 0=0 ou 1=1. Dans le cas de 0=0, on pourra exhiber une preuve à partir de l'axiome `ada`, et dans l'autre cas, à partir de `bdb`.

3.4. Cette théorie axiomatique est syntaxiquement complète. En effet, étant donnée une formule de la théorie, on peut dire si elle est ou non un théorème du système. Elle est donc également décidable. Pour savoir si une formule de longueur n est ou non un théorème, on peut

- soit utiliser la première interprétation, dans laquelle théorème et formule vraie sont identifiables,
- soit énumérer toutes les preuves qui, à partir des axiomes, permettent d'avoir des théorèmes de longueur n; ce qui est simple (mais peut-être long) puisque chacune des règles ajoute 2 caractères à la formule à laquelle elle est appliquée.

3.5. Dans cette seconde interprétation, tous les théorèmes sont également vrais. En revanche, les formules qui se ramènent à 0<=1 (l'exemple le plus simple étant `adb`) qui sont vraies dans l'interprétation ne sont pas prouvables : ce ne sont pas des théorèmes de la théorie.