Introduction à la logique

Théorie de la démonstration

René David, Karim Nour, Christophe Raffalli

Introduction à la logique

Théorie de la démonstration

2e ÉDITION

Préface de Pierre-Louis Curien

DUNOD

Illustration de couverture
akibostanci © istockphoto.com

11, rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-080632-4

Préface

La logique mathématique est certainement presque aussi ancienne que les mathématiques elles-mêmes, comme en attestent des expressions telles que *modus ponens* qui ont survécu dans les présentations actuelles des règles de raisonnement. Plus près de nous, les paradoxes et controverses du début du XX^{e} siècle ont permis l'essor de la logique mathématique comme une branche des mathématiques, qui a elle-même engendré des sous-branches : on reconnaît aujourd'hui principalement la théorie des modèles, la théorie des ensembles, et la théorie de la démonstration. Cette dernière a pour objet d'étude les systèmes de preuves formelles, et vise à une compréhension des notions de preuve et de calcul, si possible indépendante de telle ou telle représentation précise. La théorie de la démonstration connaît aujourd'hui un essor considérable en raison de rapports très étroits avec l'informatique, et plus particulièrement avec le développement de logiciels sûrs.

Ce manuel est une introduction à la logique qui prend le point de vue de la théorie de la démonstration. Il n'est pas en premier lieu destiné à des étudiants souhaitant ensuite se spécialiser en logique, bien qu'il puisse bien servir ce rôle également. Il cherche d'abord à éduquer le lecteur (typiquement, un étudiant en licence) à la rédaction de preuves rigoureuses, et en même temps lisibles. De nombreux exemples montrent qu'il y a souvent peu de chose qui différencie une preuve rédigée avec des phrases et une preuve arborescente ne contenant que des formules et des règles. Sauf que bien sûr, dans la plupart des preuves rédigées en français, de nombreux détails sont laisses de côté, d'ailleurs qui les lirait ? L'ordinateur se révèle là une aide précieuse. Le troisième auteur du livre a développé lui-même un assistant de preuve, conçu notamment comme un outil pouvant accompagner les étudiants dans le processus d'acquisition de compétences de base en algèbre, analyse ou géométrie. Aujourd'hui, les assistants de preuve commencent à être utilisés industriellement aux fins de certification de logiciels. Et, pour revenir aux mathématiques, on a pu récemment formaliser entièrement la preuve du théorème des quatre couleurs (qui n'avait pu déjà être démontré qu'à l'aide de l'ordinateur). Ingénieurs, étudiants et chercheurs sont donc tous des clients potentiels de ces outils, qui doivent encore gagner en flexibilité, en efficacité, en facilité d'utilisation.

Ce livre est d'autant plus bienvenu qu'il n'existe aujourd'hui à ma connaissance que quelques livres de référence sur la théorie de la démonstration, mais (tout au moins en langue française) aucun livre d'introduction. La rédaction en est remarquable par la limpidité et la richesse des exemples, qui campent la logique mathématique non comme une branche isolée, voire ésotérique, des mathématiques, mais comme un domaine autonome en interaction vivante avec le reste des mathématiques, ainsi, comme nous l'avons déjà souligné, qu'avec l'informatique. Les plus accrochés trouveront des résultats (tels que la décidabilité de l'arithmétique de Presburger), des exemples (tels que la suite de Goodstein) ou des remarques plus avancées (par exemple sur la logique linéaire). Cet excellent ouvrage contient donc beaucoup de matériel, exposé dans un style très progressif, ainsi qu'un stock considérable d'exercices. Rappelons ici qu'il ne faut pas regarder trop vite les corrigés !

Pierre-Louis Curien

Avant-propos

Ce livre résulte de la conjonction de nos expériences d'enseignants de mathématiques, pas uniquement de logique, et de chercheurs en théorie de la démonstration :

- René David et Karim Nour donnent, depuis plusieurs années, des cours et des TD dans divers modules de la licence et la maîtrise de mathématiques à l'Université de Savoie. L'une des difficultés majeures des étudiants réside dans l'absence de maîtrise du raisonnement. Le cours de logique de licence s'est ainsi, peu à peu, transformé en une introduction à la théorie de la démonstration : le but étant d'aider les étudiants à apprendre à raisonner.
- L'assistant de démonstration PhoX écrit par Christophe Raffalli a été testé avec les étudiants de licence : cela les a aidé à mieux comprendre les raisonnements qu'ils « subissaient » en algèbre, en analyse, etc.

Il nous a alors semblé intéressant de rédiger un cours qui puisse, à la fois :

- être utile aux étudiants qui veulent mieux connaître la « grammaire » du raisonnement dans les mathématiques « usuelles » ;
- être une introduction à la théorie de la démonstration qui est une théorie mathématique comme une autre avec ses définitions et ses théorèmes.

À notre connaissance, un tel livre n'existait ni en Français ni en Anglais.

Merci à toux ceux qui nous ont aidé : nos collègues Noël Bernard et Jacques Doyen, Jean-François Pabion dont les nombreuses remarques, en particulier celles à caractère plus philosophique, ont toujours été fort pertinentes et surtout Yves Bertini et René Cori qui ont fait une lecture très attentive de notre manuscrit et sont la source directe de nombreuses corrections et améliorations. Bien sûr, les erreurs qui peuvent rester sont de notre seule responsabilité.

Merci à Anne Bourguignon pour son efficacité dans l'édition.

Écrire un livre est un travail redoutable mais très enrichissant. Cela a demandé beaucoup d'énergie et de temps aux auteurs... et beaucoup de patience à leurs épouses Sophie, Souhad et Rebecca. Elles savent ce que nous leur devons et nous leur dédions ce livre.

A propos de la nouvelle édition

Dans cette seconde édition nous avons corrigé quelques erreurs qui avaient échappé à notre vigilance. Nous avons aussi tenu compte des innombrables remarques faites par Chantal Berline : nous espérons ainsi avoir pu préciser ou clarifier de nombreux points. Nous en avons, enfin, profité pour ajouter une vingtaine d'exercices.

Merci à Chantal Berline pour sa lecture très attentive et à Pierre Louis Curien pour avoir accepté de préfacer cette seconde édition.

Chambéry, mai 2003.

Table des matières

Introduction

Quand un étudiant arrive en licence, il a surtout appris à calculer, à utiliser des algorithmes : les opérations élémentaires à l'école primaire, les bases du calcul algébrique (règle des signes,...) au collège et les bases de l'analyse (équations, limites, dérivées,...) au lycée.

Il a très peu appris à raisonner : cela a été fait essentiellement en géométrie. C'est en effet là qu'on trouve des phrases du genre « montrer que ce triangle est isocèle », « montrer que ces droites sont parallèles », etc. Pour tous ces raisonnements, le support visuel est fort, et les élèves qui ne « voient pas » qu'en traçant telle ou telle droite la solution apparaît ou qui ne « voient pas » dans l'espace sont très pénalisés.

Au lycée, on manipule déjà des objets « inconnus », mais c'est surtout pour faire des calculs, et quand on raisonne sur des objets représentés par des lettres, on peut les remplacer « visuellement » par un réel, un vecteur, etc. En licence, on demande aux étudiants de raisonner sur des structures plus abstraites, et donc de travailler sur des objets inconnus qui sont des éléments d'un ensemble lui même inconnu, par exemple les éléments d'un groupe quelconque. Ce support visuel n'existe alors plus.

On demande aux étudiants de raisonner, de démontrer des propriétés, mais personne ne leur a jamais appris à raisonner, à écrire des preuves. Si on demande à un étudiant de licence de mathématiques ce qu'est une démonstration, il a quelque difficulté à répondre. Il peut dire que c'est un texte dans lequel on trouve des « mots clés » : « donc », « parce que », « si », « si et seulement si », « prenons un x tel que », « supposons que », « cherchons une contradiction », etc. Mais il est incapable de donner la grammaire de ces textes ni même ses rudiments, et d'ailleurs, ses enseignants, s'ils n'ont pas suivi eux-mêmes de cours de logique, en seraient probablement incapables eux aussi.

Pour parler, un enfant n'a pas besoin de connaître la grammaire. Il imite son entourage et cela marche très bien : un enfant de six ans sait utiliser des phrases déjà compliquées quant à la structure grammaticale sans avoir jamais fait de grammaire. La plupart des enseignants ne connaissent pas non plus la grammaire du raisonnement mais, chez eux, le processus d'imitation a bien marché et ils raisonnent correctement. L'expérience de tous les enseignants d'université montre que ce processus d'imitation marche bien chez les très bons étudiants, et alors il est suffisant, mais il marche beaucoup moins bien, voire pas du tout, chez beaucoup d'autres.

Tant que le degré de complexité est faible (notamment lors d'un raisonnement de type « équationnel »), la grammaire ne sert à rien, mais quand il augmente ou quand on ne comprend pas pourquoi quelque chose est faux, il devient nécessaire de faire un peu de grammaire pour pouvoir progresser. Les enseignants et les étudiants connaissent bien la situation suivante : dans un devoir, le correcteur a barré toute une page d'un grand trait rouge et mis « faux » dans la marge. Quand l'étudiant demande ce qui est faux, le correcteur ne peut que dire des choses du genre « ça n'a aucun rapport avec une démonstration », « rien n'est juste », ..., ce qui n'aide évidemment pas l'étudiant à comprendre. Cela vient, en partie, du fait que le texte rédigé par l'étudiant utilise les mots voulus mais dans un ordre plus ou moins aléatoire et qu'on ne peut donner de sens à l'assemblage de ces mots. De plus, l'enseignant n'a pas les outils nécessaires pour pouvoir expliquer ce qui ne va pas. Il faut donc les lui donner !

Ces outils existent mais sont assez récents. La théorie de la démonstration est une branche de la logique mathématique dont l'origine est la crise des fondements : il y a eu un doute sur ce qu'on avait le « droit » de faire dans un raisonnement mathématique. Des paradoxes sont apparus. Par exemple (ce paradoxe est connu sous le nom de « paradoxe de Russel ») : soit a l'ensemble des ensembles ne se contenant pas eux-mêmes. Il est facile de montrer que a appartient à a si et seulement si a n'appartient pas à a, ce qui est manifestement contradictoire. Il a alors été nécessaire de préciser les règles de démonstration et de vérifier que ces règles ne sont pas contradictoires. Cette théorie est apparue il y a environ un siècle, ce qui est très peu puisque l'essentiel des mathématiques enseignées en premier et deuxième cycle universitaire est connu (avec un formalisme éventuellement différent) depuis deux ou trois cents ans.

Plan du livre

Le but de ce livre est double.

Dans la première partie, on présente la logique comme un outil pour aider l'étudiant qui débute des études de mathématiques.

La seconde partie est une introduction à la théorie de la démonstration pour l'étudiant qui veut apprendre la logique en tant que discipline mathématique.

L'équilibre entre ces deux buts n'est pas simple : trop de formalisation pourra décourager le lecteur qui veut seulement apprendre à raisonner, pas assez donnera l'impression d'un manque de sérieux. Nous avons essayé de trouver le juste milieu.

La première partie (chapitres 1 à 3) correspond au cours de logique de la licence de mathématiques à l'université de Savoie.

Le chapitre 1 présente les bases de la théorie de la démonstration : qu'est-ce qu'une formule, une démonstration ? Quelles en sont les règles ? On présente en détail ce qu'on appelle la *déduction naturelle* mais d'autres formalisations sont possibles. Des exemples concrets (issus des mathématiques courantes) sont donnés. Il s'agit donc de :

- donner à l'étudiant les outils (i.e. la grammaire) pour pouvoir écrire correctement des preuves. Le but n'est évidemment pas de vouloir tout formaliser complètement : ce n'est pas faisable et cela ne sert à rien. Il faut cependant qu'il puisse faire suffisamment de preuves en détail pour que les principes en soient acquis : au collège, pour comprendre la règle des signes, il faut en faire usage ;
- l'aider à analyser suffisamment la preuve pour pouvoir la rédiger et pouvoir, éventuellement, lui montrer les erreurs faites ;
- l'aider à comprendre ce qu'on lui demande (dans le cours d'algèbre, de topologie, etc.) quand il fait une démonstration.

Le chapitre 2 donne la sémantique, c'est-à-dire la notion générale d'interprétation (ou de structure) et de vérité d'une formule. Le résultat essentiel en est le théorème de complétude : on peut démontrer une formule si et seulement si elle est vraie quelle que soit l'interprétation des symboles utilisés.

Le chapitre 3 donne des exemples de théories mathématiques. Après un rappel de quelques théories classiques, on introduit deux théories particulièrement importantes :

- l'arithmétique de Peano, la base de la théorie des nombres ;
- la théorie des ensembles de Zermelo Fraenkel, le fondement habituel de toutes les mathématiques.

La seconde partie (chapitres 4 à 7) est au niveau du master.

Le chapitre 4 introduit la logique intuitionniste : ses règles de démonstration, sa sémantique et son théorème de complétude. La logique intuitionniste se distingue de la logique « habituelle » (on appelle cette dernière la logique *classique*) par l'interdiction du raisonnement par l'absurde. L'intérêt de cette logique est de donner des preuves constructives : si on a prouvé qu'il existe un entier vérifiant telle propriété, la preuve donne explicitement cet entier alors qu'avec un raisonnement par l'absurde on sait simplement qu'il n'est pas possible qu'il n'y ait pas d'entier vérifiant la propriété. Le lien entre cette logique et la preuve de programmes informatiques est un domaine très actif de recherche.

Au chapitre 5, on donne une autre formalisation de la notion de démonstration : le calcul des séquents. Elle est moins naturelle (et donc moins utilisable dans la pratique quotidienne) que celle qu'on a présentée au chapitre 1, mais elle permet de montrer plus facilement des propriétés importantes des démonstrations et, en particulier, que les règles ne sont pas contradictoires.

Le chapitre 6 est consacré à des extensions de la logique introduite au chapitre 1. Celle-ci ne permet pas de quantifier sur plusieurs types d'objets (par exemple les réels et les ensembles de réels ou les fonctions de $\mathbb{R}$ dans $\mathbb{R}$). Il s'agit donc d'étendre son pouvoir d'expression pour pouvoir le faire : on introduit pour cela les logiques d'ordre supérieur et les logiques multi sorte. La plupart des logiciels d'aide à la démonstration

sont basés sur des logiques d'ordre supérieur. Ce chapitre est donc aussi destiné aux étudiants qui veulent connaître les bases dans ce domaine.

Enfin, le chapitre 7 est une courte introduction aux techniques de base de la démonstration automatique. On y donne les résultats essentiels sur l'unification, la méthode des tableaux et la résolution.

Indépendance des chapitres

Les sections 1 à 5 du chapitre 1 et du chapitre 2 sont des préalables à tous les autres chapitres. Les chapitres 3 à 6 et les sections 1, 2 et 4 du chapitre 7 peuvent, dans une large mesure, être lus indépendamment les uns des autres. La section 3 du chapitre 7 nécessite une lecture préalable des sections 1 et 2 du chapitre 5.

Ce livre est destiné :

1. Aux étudiants de mathématiques et d'informatique.

- En second cycle, ils trouveront dans ce livre un cours de base de logique ainsi que de nombreux exercices corrigés. Ce livre pourra aussi être utile aux étudiants de troisième cycle en logique.
- En classes préparatoires et en licence de mathématiques, le chapitre 1 pourra être lu et travaillé avec beaucoup de profit par les étudiants qui souhaitent mieux comprendre ce qu'est un raisonnement mathématique.

2. Aux enseignants de mathématiques et d'informatique, du lycée à l'université.

- Ceux qui souhaitent apprendre, pour eux-mêmes, les bases de la logique et, en particulier, la théorie de la démonstration. Faut-il rappeler que depuis 1999 il y a au programme du CAPES de mathématiques un chapitre de logique ?
- Mais aussi tous les autres. Comment répondre aux questions des étudiants qui demandent comment on fait une démonstration quand on n'a pas soi-même une idée un tant soit peu précise sur le sujet ? Les auteurs de ce livre ont fait, à plusieurs reprises, des exposés d'introduction à la logique face à un public d'universitaires : les nombreuses questions posées montrent, à l'évidence, que cette introduction n'est pas inutile.

Utilisation d'un assistant de démonstration

Depuis l'apparition des ordinateurs, les logiciens ont cherché à faire des preuves sur machines, l'un des buts étant de pouvoir vérifier (avec un grand degré de fiabilité) des preuves longues et difficiles. Cette vérification est utile : il se peut, en effet, que dans de telles preuves on n'ait pas « vu » une difficulté. Des logiciels d'aide à la démonstration ont été développés qui permettent d'écrire des démonstrations complètes et sûres. Ils sont utilisés, en particulier, pour la vérification de logiciels.

Depuis plusieurs années, nous enseignons en licence, en maîtrise et en DEA de mathématiques. Dans le cours de logique de la licence de mathématiques de l'université de Savoie, nous avons fait travailler les étudiants sur le logiciel PhoX écrit par C. Raffalli. Cette expérience a été très positive et il semble que

cela ait permis aux étudiants de progresser. Nous avons, par exemple, fait démontrer le théorème des valeurs intermédiaires et cela nous a permis de mieux comprendre où étaient les difficultés rencontrées par les étudiants : si un enseignant rappelle la preuve de ce théorème en 3 $^{\text{e}}$ année de licence (elle est supposée connue dès les 2 premières années), il n'y passera que quelques petites minutes. Et pourtant, quand on demande aux étudiants de l'écrire un peu en détail (« j'applique la définition de la continuité avec $\varepsilon = ...$, je trouve α tel que etc.), on se rend compte que tout cela est un peu vide de sens pour beaucoup d'entre eux.

L'aide de l'ordinateur devient indispensable quand on veut formaliser complètement des preuves un peu longues. Les outils informatiques dont on peut disposer commencent à devenir performants.

L'assistant de démonstration PhoX (*cf.* annexe p. 261), sur lequel travaillent les auteurs du livre, permet cette formalisation et peut donc aider le lecteur à appréhender la notion de démonstration. PhoX est disponible gratuitement, ainsi que la correction de certains exercices réalisée avec PhoX, à partir du site Web :

`www.lama.univ-savoie.fr/~raffalli/dnr.html`

Nous souhaitons pouvoir développer une banque de preuves entièrement formalisées qui serait disponible sur ce site Web. Les lecteurs intéressés peuvent évidemment participer à son élaboration en nous écrivant à l'adresse ci-dessous. De même, le lecteur ne doit pas hésiter à nous signaler, par e-mail, les erreurs qui subsistent dans le livre :

`dnr@univ-savoie.fr`

Notations et abréviations couramment utilisées

Les notations utilisées dans ce livre sont standard. Quelques précisions peuvent être utiles :

- $\{x \,/\, P(x)\}$ ou $\{x : P(x)\}$ désigne l'ensemble des éléments x qui satisfont la propriété P ;

- $\subset$ représente l'inclusion large, $\subsetneq$ l'inclusion stricte ;

- $A - B$ représente le complémentaire de B dans A ;

- Les deux symboles nouveaux $\vdash$ et $\models$, beaucoup utilisés en logique et donc dans ce livre, sont définis pages 24 et 70 ;

- Si f est une fonction, $f \upharpoonright A$ représente la restriction de f à l'ensemble A ;

- Les lettres Γ et Δ seront réservées pour représenter des ensembles de formules. On s'autorisera souvent à utiliser l'expression « soient Γ, A des formules » pour signifier que Γ est un ensemble de formules et A une formule ;

- Le symbole $=$ est utilisé à plusieurs niveaux qu'il est quelquefois important de ne pas confondre :

1. Au niveau qu'on peut appeler « méta » : quand on définit quelque chose (par exemple, soit $E = \{x \,/\, ...\}$) ou quand on prouve un résultat (par exemple, quand on affirme que, pour tout x, $(x+1)^2 = x^2 + 2x + 1$).

2. Au niveau objet : quand on écrit « soit F la formule $\forall x\,(x.y = y.x)$ », le signe $=$ est un symbole du langage qui sert à écrire l'objet mathématique qu'est la formule F.

• On dira souvent « preuve par induction » à la place de « preuve par récurrence ». Ces deux mots seront utilisés indifféremment.

• On utilisera les abréviations *i.e.*, *c. à d.*, *ssi* et *resp.* pour *id est*, *c'est-à-dire* et *respectivement* : les deux premières sont synonymes. *ssi* signifie *si et seulement si*. La dernière est utilisée pour éviter la répétition de deux phrases presque identiques.

• Il est fréquent qu'une définition commence par une phrase telle que « un... est un... si... ». Par exemple, « une fonction f est un morphisme si... ». Dans une telle phrase on utilisera indifféremment *si* et *ssi*.

Quelques rappels sur la cardinalité et les ordres

1) Cardinalité

On dit qu'un ensemble est *dénombrable* s'il est en bijection avec $\mathbb{N}$ et au plus dénombrable s'il est fini ou dénombrable. On rappelle que la réunion d'une famille dénombrable d'ensembles dénombrables est dénombrable.

Le théorème de Cantor-Bernstein affirme que si A et B sont deux ensembles, s'il existe une injection de A dans B et une injection de B dans A, alors il existe une bijection de A dans B. En particulier, pour montrer que A est dénombrable, il suffit de donner une injection de $\mathbb{N}$ dans A et une injection de A dans un ensemble dénombrable.

2) Ordres

Un ordre sur E est *bien fondé* s'il n'existe pas de suite infinie strictement décroissante. Un *bon ordre* est un ordre total et bien fondé ou, de manière équivalente, un ordre tel que toute partie non vide de E possède un plus petit élément.

L'ordre *lexicographique* sur les n-uplets d'entiers est défini par : $(x_1, ..., x_n) < (y_1, ..., y_n)$ ssi il existe i tel que $x_i < y_i$ et $x_j = y_j$, pour tout $j < i$. C'est un ordre bien fondé qui permet de faire des preuves par récurrence sur, par exemple, un triplet d'entiers. Cela signifie que, pour montrer une propriété P qui dépend de trois paramètres n, m, p, il suffit de montrer l'analogue du cas de récurrence c'est-à-dire : si P est vraie pour tout triplet $(n, m, p) < (n', m', p')$ alors P est également vraie pour (n', m', p').

Une *chaîne* d'un ensemble ordonné E est un sous-ensemble de E qui est totalement ordonné.

3) La structure d'*arbre* est omniprésente en mathématique mais on l'utilise rarement de manière explicite. On ne présentera pas ici de définition formelle (voir section 4.4.3) : les exemples donnés seront plus parlants et suffiront sans doute pour la compréhension. De même, on ne définira pas les notions de racine, de feuille, de nœud, de branche, qui parlent d'elles-mêmes.

Avertissement

Ce livre a été écrit par trois personnes. Telle section, écrite par l'un, a été revue, retravaillée par les autres, mais chacun a son style, son vocabulaire. Nous avons essayé,

au maximum, d'harmoniser notations, styles, etc. Il se peut qu'il reste des différences que, malgré les nombreuses relectures, nous n'avons pas vues. Si cela peut paraître gênant, c'est quelquefois aussi une richesse...

Code de lecture

Les références sont données en valeur *absolue* : par exemple, à l'intérieur du chapitre 3, la référence à la section 2.4 signifie la section 4 du chapitre 2 et non pas la sous-section 3.2.4.

Les panneaux suivants indiquent :

: un passage difficile ;

: il est indispensable de s'arrêter et de bien comprendre ;

: un passage difficile sur lequel il n'est pas nécessaire de s'attarder

Chapitre **1**

Formules et démonstrations de la logique du premier ordre

1.1 INTRODUCTION

Dans un cours de mathématiques (d'algèbre, d'analyse, de géométrie, ...), on démontre les propriétés de différents types d'objets (entiers, réels, matrices, suites, fonctions continues, courbes, ...). Pour pouvoir prouver ces propriétés, il faut bien sûr que les objets sur lesquels on travaille soient clairement définis (qu'est-ce qu'un entier, un réel, ... ?).

En *logique du premier ordre* et, en particulier, en théorie de la démonstration, les objets que l'on étudie sont les *formules* et leurs *démonstrations*. Il faut donc donner une définition précise de ce que sont ces notions. Les termes et les formules forment la grammaire d'une langue, simplifiée à l'extrême et construite exactement pour dire ce que l'on veut sans ambiguïté et sans détour inutile.

Les termes. Ils représentent les objets dont on veut prouver des propriétés.

– En algèbre, les termes peuvent désigner les éléments d'un groupe (ou anneau, corps, espace vectoriel, etc). On manipule aussi des ensembles d'objets (sous-groupe, sous-espace vectoriel, etc). Les termes qui désignent ces objets, d'un autre type, seront appelés *termes du second ordre*. Ils ne seront pas considérés avant le chapitre 6.

– En analyse, les termes pourront désigner les réels ou (par exemple, si on se place dans des espaces fonctionnels) des fonctions.

Les formules. Elles représentent les propriétés des objets que l'on étudie.

– En algèbre, on pourra écrire des formules pour exprimer que deux éléments commutent, qu'un sous-espace vectoriel est de dimension 3, etc.

– En analyse, on écrira des formules pour exprimer la continuité d'une fonction, la convergence d'une suite, etc.

– En théorie des ensembles, les formules pourront exprimer l'inclusion de deux ensembles, l'appartenance d'un élément à un ensemble, ...

Les démonstrations. Elles permettent d'établir qu'une formule est *vraie*. Le sens précis de ce mot aura lui aussi besoin d'être défini (*cf.* chapitre 2). Plus exactement, elles sont des déductions sous hypothèses, elles permettent de « mener du vrai au vrai », la question de la vérité de la conclusion étant alors renvoyée à celle des hypothèses, laquelle ne regarde pas la logique mais repose sur la connaissance que nous avons des choses dont nous parlons.

De même que la notion de corps est une formalisation de la notion de nombre, les démonstrations sont une formalisation du raisonnement que l'on pratique d'habitude en mathématique.

Il est important de noter que, dans ce chapitre, on n'étudie que le côté qu'on appellera *syntaxique* des formules et des démonstrations, i.e. il n'y aura pas *ici* de notion de *vrai* ni de *faux*. La notion de vérité (qu'on appellera le côté *sémantique*) n'apparaîtra que dans le chapitre suivant. Bien sûr, pour pouvoir lire une formule, comprendre une démonstration, on a besoin d'y mettre un sens, mais ce sens n'intervient pas. Un ordinateur (qui ne comprend rien à ce qu'il fait !) peut, sans problème, manipuler des démonstrations sur les groupes, les espaces vectoriels, ... sans avoir, dans son imaginaire, des exemples pour fonder son intuition. Dans les exemples donnés dans ce chapitre, on n'hésitera pas à faire appel à cette intuition mais il doit être clair qu'ici ce ne sera qu'une aide.

Dans ce chapitre, on prouve des résultats sur les termes, les formules et les démonstrations. On va donc écrire des formules (on pourrait les appeler *méta-formules*) qui expriment leurs propriétés et on les démontrera. On fera donc des méta-démonstrations.

Il faut prendre garde à ne pas mélanger le niveau où vivent les termes, les formules et les démonstrations, qui sont les objets de notre étude, et le niveau *méta* où l'on énonce et prouve les propriétés de ces objets.

Pour faciliter la tâche du lecteur et lui éviter de confondre ces deux niveaux, on utilisera plutôt les mots du tableau ci-dessous qui donne la correspondance entre le niveau « objet » et le niveau « *méta* ».

Niveau	**Mots utilisés**
objet	formule, démonstration (ou dérivation)
méta	propriété, preuve

On réservera également, dans la plupart des cas, la notation symbolique au niveau objet. Par exemple, on n'utilisera pas l'abréviation $\forall$ pour « pour tout » au niveau méta.

1.2 LES FORMULES

1.2.1 Le langage

En mathématique, on utilise, suivant le domaine, différents langages qui se distinguent par les symboles utilisés (*cf.* exemple 1.2.2). La définition ci-dessous exprime simplement qu'il suffit de donner la liste de ces symboles pour préciser le langage.

Définition 1.2.1 Un *langage* (du premier ordre) est la donnée d'une famille (pas nécessairement finie) de symboles. On en distingue trois sortes :

- les symboles de *constante* ;
- les symboles de *fonction*. À chaque symbole est associé un entier strictement positif qu'on appelle son *arité* : c'est le nombre d'arguments de la fonction. Si l'arité est 1 (resp. 2, ..., n), on dit que la fonction est *unaire* (resp. *binaire*, ..., n-aire) ;
- les symboles de *relation*. De la même manière, à chaque symbole est associé un entier positif ou nul (son arité) qui correspond à son nombre d'arguments et on parle de *relation unaire*, *binaire*, ..., n-aire.

On utilise quelquefois le mot *vocabulaire* ou le mot *signature* à la place du mot langage.

Le mot *prédicat* peut être utilisé à la place du mot relation. On parle alors de *calcul des prédicats* au lieu de *logique du premier ordre*.

Remarques.

1. Un symbole de constante peut être vu comme un symbole de fonction à 0 argument.
2. On considérera (sauf mention contraire) que chaque langage contient le symbole de relation binaire $=$ et un symbole de relation à 0 argument dénoté $\bot$ (on lit *bottom* ou *absurde*) qui représentera le faux. Dans la description d'un langage, on omettra donc souvent de les mentionner.
3. Le symbole $\bot$ est souvent redondant. On peut en effet, sans l'utiliser, écrire une formule qui est toujours fausse. Il permet cependant de représenter le faux d'une manière canonique et donc d'écrire des règles de démonstration générales.
4. Le rôle des fonctions et des relations est très différent. Comme on le verra plus loin, les symboles de fonction et de constante sont utilisés pour construire les termes (i.e. les objets du langage) et les symboles de relation pour construire les formules (i.e. les propriétés de ces objets).

Exemple 1.2.2

1. Le langage $\mathcal{L}_1$ de la *théorie des groupes* contient les symboles :
   - constantes : e (pour représenter l'élément neutre)
   - fonctions : $*$ (binaire, pour l'opération du groupe), $.^{-1}$ (unaire, pour l'inverse)
   - relation : $=$
2. Le langage $\mathcal{L}_2$ de la *théorie des corps ordonnés* contient les symboles :
   - constantes : 0, 1
   - fonctions : $+$, $\times$, $-$, $.^{-1}$. On utilise en fait deux symboles $-$ que l'on confond. L'un est unaire, l'autre binaire (remarquer que, sur les calculatrices, il y a deux touches distinctes)
   - relations : $=$, $\leqslant$
3. Le langage $\mathcal{L}_3$ de la *théorie des espaces vectoriels* sur $\mathbb{R}$ contient les symboles :
   - constantes : 0
   - fonctions : $+$, $(f_\lambda)_{\lambda\in\mathbb{R}}$. Ici l'ensemble des symboles est infini. f_λ correspond à la multiplication par le scalaire λ. On reconnaît ainsi que l'addition est une opération interne alors que la multiplication par un scalaire correspond à une opération externe
   - relation : $=$
4. Le langage $\mathcal{L}_4$ de la *théorie des ensembles* contient les symboles :
   - constante : $\varnothing$
   - fonctions : $\cap, \cup, .^c$ (pour le complémentaire)
   - relations : $=, \in, \subset$
5. Le langage $\mathcal{L}_5$ de l'*analyse réelle* contient les symboles :
   - constantes : $0, 1, ..., e, \pi$, ...
   - fonctions : $+$, $\times, |.|, \sin$, $\ln$, ...
   - relations : $=$, $\leqslant$, ...

1.2.2 Les termes

Dans toute la suite, on se donne un ensemble (infini) $\mathcal{V}$ de *variables*. Les variables seront notées : $x, y, z, ...$ (éventuellement indexées : $x_1, ...$).

Les *termes* (sous-entendu *du premier ordre*) représentent les objets associés au langage. Formellement :

Définition 1.2.3 Soit $\mathcal{L}$ un langage.

1. L'ensemble $\mathcal{T}$ des termes sur $\mathcal{L}$ est le plus petit ensemble contenant les variables, les constantes et stable par *l'application* des symboles de fonctions de $\mathcal{L}$ à des termes.

2. Un terme *clos* est un terme qui ne contient pas de variables.

3. Pour obtenir une définition plus formelle, on peut écrire :
$\mathcal{T}_0 = \{t \ / \ t$ est une variable ou un symbole de constante$\}$ et, pour tout $k \in \mathbb{N}$, $\mathcal{T}_{k+1} = \mathcal{T}_k \cup \{f(t_1, ..., t_n) \ / \ t_i \in \mathcal{T}_k$ et f symbole de fonction d'arité $n\}$.
On pose alors $\mathcal{T} = \bigcup_{k\in\mathbb{N}} \mathcal{T}_k$.

4. On appellera *hauteur* d'un terme t le plus petit k tel que $t \in \mathcal{T}_k$.

Remarques.

1. La définition signifie que les variables et les constantes sont des termes et que si f est un symbole de fonction n-aire et $t_1, ..., t_n$ sont des termes alors $f(t_1, ..., t_n)$ est un terme.

2. La définition ne fait que donner des règles d'écriture. Il faut donc la comprendre sous la forme : si f est un symbole, on peut *écrire* $f(t_1, ..., t_n)$. Le choix de cette écriture n'est évidemment pas neutre puisque son *sens* (*cf.* définition 2.2.3) en sera l'application d'une fonction à ses arguments.

3. On sera amené souvent à donner ce genre de définition. Dans la suite, on la formulera de la manière suivante : l'ensemble $\mathcal{T}$ des termes est défini par la *grammaire* :

$$\mathcal{T} = \mathcal{V} \mid \mathcal{S}_C \mid \mathcal{S}_F(\mathcal{T}, ..., \mathcal{T})$$

$\mathcal{S}_C$ étant l'ensemble des symboles de constantes et $\mathcal{S}_F$ l'ensemble des symboles de fonctions du langage.
Cette expression se lit de la manière suivante : un élément de l'ensemble $\mathcal{T}$ que l'on est en train de définir est soit un élément de $\mathcal{V}$, soit un élément de $\mathcal{S}_C$, soit l'application d'un élément $f \in \mathcal{S}_F$ d'arité n à n éléments de $\mathcal{T}$.
Attention : le fait que f soit de la bonne arité est seulement implicite dans cette notation. De plus, l'écriture $\mathcal{S}_F(\mathcal{T}, ..., \mathcal{T})$ ne signifie pas que tous les arguments d'une fonction sont identiques mais simplement que ces arguments sont des éléments de $\mathcal{T}$.

4. On peut formaliser de façon analogue beaucoup de situations courantes en mathématiques. Par exemple, dans un espace vectoriel, le sous-espace F engendré par une partie A peut être défini par la grammaire :

$$F = A \mid F + F \mid f_\lambda(F) \quad \text{pour} \quad \lambda \in \mathbb{R}$$

5. Pour les fonctions binaires, la notation infixée (par exemple $x+y$ ou $A \cap B$) sera souvent préférée à la notation préfixée ($+(x, y)$ ou $\cap(A, B)$) mais cela nécessite, pour une meilleure lisibilité, l'usage de parenthèses.

Exemple 1.2.4 Les langages utilisés ci-dessous sont définis dans l'exemple 1.2.2.

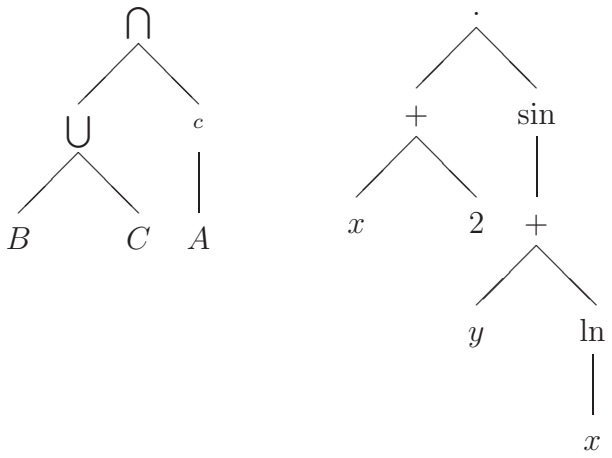
1. Sur le langage $\mathcal{L}_1$: $x * y^{-1}$ et $x^{-1} * (y * x^{-1})^{-1}$ sont des termes. $e^{-1} * e$ est un terme clos.

2. Sur le langage $\mathcal{L}_2$: $(x - (1 + (x^{-1})^{-1})) \times (0 + y^{-1})$ est un terme. Par ailleurs $(1 \times (0 + 1)^{-1}) + ((1 \times 0) + 0))$ est un terme clos.

3. Sur le langage $\mathcal{L}_3$: $f_{\sin(1)}(f_{-2}(x+y)+f_{1/2}(y+f_{\ln(7)}(x)))$ est un terme. $f_\pi(0+f_{\sqrt{2}}(0))$ est un terme clos. Par contre, $f_x(x)$ n'est pas un terme du langage $\mathcal{L}_3$.
4. Sur le langage $\mathcal{L}_4$: $(x \cup y)^c$ et $(x^c \cap (\varnothing \cup y))^c$ sont des termes. $\varnothing \cap (\varnothing^c \cup \varnothing)^c$ est un terme clos.
5. Sur le langage $\mathcal{L}_5$: $\sin(\ln(x) \times \cos(|y+e|))$ est un terme. $\ln(|\cos(e) - \sin(e)|)$ est un terme clos.

Remarques.

1. On l'a déjà dit, on ne cherche pas, dans ce chapitre, à donner un sens aux termes et donc, par exemple, 0^{-1} et $\ln(0)$ sont bien des termes sur, respectivement, les langages $\mathcal{L}_2$ et $\mathcal{L}_5$.
2. Il est souvent commode de voir un terme comme un arbre dont chaque nœud est étiqueté par un symbole de fonction et chaque feuille par une variable ou une constante. Par exemple les termes $(B \cup C) \cap A^c$ et $(x+2).\sin(y+\ln(x))$ sont représentés par les arbres :



3. Dans la suite on va sans cesse définir des notions (ou prouver des résultats) par *récurrence* sur la structure d'un terme.
   a) Pour prouver une propriété P sur les termes, il suffit de prouver P pour les variables et les constantes et de prouver $P(f(t_1,...,t_n))$ à partir de $P(t_1),...,P(t_n)$. On fait ici une preuve par induction sur la hauteur d'un terme.
   b) Pour définir une fonction Φ sur les termes, il suffit de la définir sur les variables et les constantes et de dire comment on obtient $\Phi(f(t_1,...,t_n))$ à partir de $\Phi(t_1),...,\Phi(t_n)$. On fait ici une définition par induction sur la hauteur d'un terme. Voici, à titre d'exemple, une telle définition.

Définition 1.2.5 La *taille* (on dit aussi la *longueur*) d'un terme t (notée $\tau(t)$) est le nombre de symboles de fonction apparaissant dans t. Formellement :

- $\tau(x) = \tau(c) = 0$ si x est une variable et c est une constante ;
- $\tau(f(t_1, ..., t_n)) = 1 + \sum_{1 \leqslant i \leqslant n} \tau(t_i)$.

Remarque. La preuve par induction sur la hauteur d'un terme sera parfois insuffisante. On pourra alors prouver une propriété P sur les termes en supposant la propriété vraie pour tous les termes de taille $p < n$ et en la démontrant pour les termes de taille n. Il s'agira alors d'une preuve par récurrence sur la taille du terme.

1.2.3 Les formules

Les formules sont construites à partir des formules dites *atomiques* en utilisant des *connecteurs* et des *quantificateurs*. On utilisera les connecteurs et les quantificateurs suivants :

– *connecteur unaire* : $\neg$ qui se lit *non*.

– *connecteurs binaires* :

$\wedge$ qui se lit *et*. Penser à $x \in A \cap B$ qui signifie $x \in A$ et $x \in B$.

$\vee$ qui se lit *ou*. Penser à $x \in A \cup B$ qui signifie $x \in A$ ou $x \in B$.

$\rightarrow$ qui se lit *implique* ou *flèche*.

– *quantificateurs* : $\exists$ qui se lit *il existe* et $\forall$ qui se lit *pour tout*.

Cette notation des connecteurs est standard. Elle est utilisée pour éviter les confusions entre les formules (objets étudiés dans ce livre) et le langage courant (le métalangage).

On peut aussi trouver, dans certains livres de logique, les notations suivantes :

– $\rightarrow$ est noté $\supset$. Penser au fait que la conclusion est incluse dans l'hypothèse.

– $\forall$ est noté $\bigwedge$. C'est une généralisation du *et*.

– $\exists$ est noté $\bigvee$. C'est une généralisation du *ou*.

Le choix des connecteurs est relativement arbitraire, i.e. on pourrait en choisir d'autres. Par exemple, on pourrait ajouter $\leftrightarrow$ (pour l'équivalence), $\nmid$ (la *barre de Schaeffer* utilisée, en particulier, en électronique), etc.

Définition 1.2.6 Soit $\mathcal{L}$ un langage.

1. Les formules *atomiques* de $\mathcal{L}$ sont les formules de la forme : $R(t_1, ..., t_n)$ où R est un symbole de relation n-aire de $\mathcal{L}$ et $t_1, ..., t_n$ sont des termes de $\mathcal{L}$. On note $\mathcal{A}tom$ l'ensemble des formules atomiques. Si on note $\mathcal{S}_R$ l'ensemble des symboles de relation, on peut écrire :
$$\mathcal{A}tom = \mathcal{S}_R(\mathcal{T}, ..., \mathcal{T})$$
2. L'ensemble $\mathcal{F}$ des *formules* (de la logique du premier ordre) de $\mathcal{L}$ est défini par la grammaire (où x parcourt l'ensemble des variables) :
$$\mathcal{F} = \mathcal{A}tom \mid \mathcal{F} \vee \mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \rightarrow \mathcal{F} \mid \neg\mathcal{F} \mid \exists x\, \mathcal{F} \mid \forall x\, \mathcal{F}$$

Remarques.

1. On rappelle que cela veut dire que l'ensemble des formules est le plus petit ensemble contenant les formules atomiques et tel que si F_1 et F_2 sont des formules alors $F_1 \vee F_2$, etc. sont des formules.
2. Pour les relations binaires, la notation infixée (par exemple $x = y$ et $x \leqslant y$) sera souvent utilisée à la place de la notation préfixée $(=(x, y)$ et $\leqslant (x, y))$.
3. On fera attention à ne pas confondre termes et formules : $\sin(x)$ est un terme, $x = 3$ est une formule, mais $\sin(x) \wedge (x = 3)$ n'est rien ; on ne peut, en effet, mettre un connecteur entre un terme et une formule.

Exemple 1.2.7 Les langages utilisés ci-dessous ont été définis dans l'exemple 1.2.2.

1. Sur le langage $\mathcal{L}_1$: $\forall x \exists y\ ((x * y = e) \wedge (\neg(y * x = e)))$ et $\forall x\ ((x = e) \vee \exists y\ ((\neg(y = e)) \wedge (y * x = e)))$ sont des formules.
2. Sur le langage $\mathcal{L}_2$: $\forall x \forall y\ (x \times y = 0 \rightarrow ((x = 0) \vee (y = 0)))$ et $\exists x\ ((\neg(x = 0)) \wedge (x \times 0 = 1))$ sont des formules.
3. Sur le langage $\mathcal{L}_3$: $\forall x \forall y\ (f_2(x + y) = f_2(x) + f_2(x))$ et $\forall x \forall y\ (f_2(x + y) = 0 \rightarrow ((x = 0) \wedge (y = 0)))$ sont des formules.
4. Sur le langage $\mathcal{L}_4$: $\forall x \forall y\ ((x \cup y)^c = (x^c \cap y^c))$ et $\forall x \forall y\ (x \cap y = \varnothing \rightarrow (x = \varnothing \wedge y = \varnothing))$ sont des formules.
5. Sur le langage $\mathcal{L}_5$: $\forall x\ (x > 0 \rightarrow \exists y\ ((y > 0) \wedge (x = y \times y)))$ et $\exists x\ (\sin(x) = 0 \wedge \cos(x) = 0)$ sont des formules.

Remarque. On notera que certaines des formules écrites ci-dessus sont *intuitivement* vraies et d'autres fausses.

Notation 1.2.8

1. On utilise des parenthèses, crochets et accolades : $(,), [,], \{,\}$ pour faciliter la lecture ou pour lever les ambiguïtés. Sans parenthèses, la formule $\neg A \wedge B$ est ambiguë : elle peut être lue soit comme $(\neg A) \wedge B$, soit comme $\neg(A \wedge B)$. Pour alléger les notations, on utilisera des règles de priorité suivantes :

   0 : les symboles de relations du langage

   1 : $\neg$, $\forall$, $\exists$

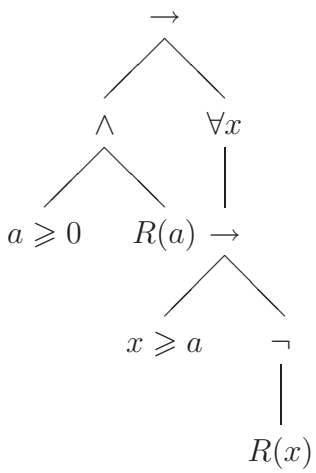
   2 : $\wedge$, $\vee$

   3 : $\rightarrow$

   Ainsi $\forall x\ A \wedge B \rightarrow \neg C \vee D$ est la formule $((\forall x\ A) \wedge B) \rightarrow ((\neg C) \vee D)$.
2. On peut montrer que, si on utilisait une notation préfixée (par exemple $\vee F_1 F_2$), les parenthèses ne seraient pas nécessaires, i.e. il n'y aurait qu'une manière de lire une formule. Ceci est également valable pour les termes. Si on ne le fait pas c'est parce que, dans la pratique, la notation préfixée est beaucoup plus difficile à lire.
3. On utilise aussi les abréviations suivantes :
   - $\forall x_1, ..., x_n\ A$ pour $\forall x_1 ... \forall x_n\ A$.

- $\forall x > 0\ A$ pour $\forall x\ (x > 0 \to A)$ et $\exists x > 0\ A$ pour $\exists x\ (x > 0 \wedge A)$. On emploiera cette notation avec tout autre symbole de relation binaire en notation infixée, par exemple $\forall x \in y\, F[x, y]$. Noter que, dans le cas de $\forall$, on a une implication, dans le cas de $\exists$ c'est une conjonction.
- $A \wedge B \wedge C$ pour $A \wedge (B \wedge C)$. On pourra changer si l'on veut la place des parenthèses $(A \wedge B) \wedge C$ puisque l'on verra que ces formules sont équivalentes. On aura la même convention avec $\vee$. On note, dans la suite, $\bigwedge_{1 \leqslant i \leqslant n} A_i$ la formule $A_1 \wedge ... \wedge A_n$ et $\bigvee_{1 \leqslant i \leqslant n} A_i$ la formule $A_1 \vee ... \vee A_n$.
- $A_1 \to A_2 \to ... \to A_n \to A$ pour $A_1 \to (A_2 \to (...(A_n \to A)...))$.
- $A \leftrightarrow B$ pour $(A \to B) \wedge (B \to A)$.
- $t \neq u$ pour $\neg(t = u)$.

Remarque. Comme pour les termes, une formule peut être vue comme un arbre dont les nœuds sont étiquetés par des connecteurs ou des quantificateurs et les feuilles par des formules atomiques. Par exemple la formule : $(a \geqslant 0 \wedge R(a)) \to \forall x\ (x \geqslant a \to \neg R(x))$ est représentée par l'arbre :



Exemple 1.2.9

1. La formule $\forall x_0\ \forall \varepsilon > 0\ \exists \alpha > 0\ \forall x\ (|x - x_0| < \alpha\ \to\ |f(x) - f(x_0)| < \varepsilon)$
est l'abréviation de la formule :
$\forall x_0 \forall \varepsilon\ \{\varepsilon > 0\ \to \exists \alpha\ [\alpha > 0\ \wedge \forall x\ (|x - x_0| < \alpha\ \to\ |f(x) - f(x_0)| < \varepsilon)]\}$
Elle représente la continuité de la fonction f.

2. La formule
$H(e) \wedge \forall x, y\ \{H(x) \wedge H(y) \to H(x.y^{-1})\} \wedge \forall x, y\ \{H(y) \to H(x.y.x^{-1})\}$
représente le fait que H (ou plus exactement l'ensemble des éléments qui ont la propriété H) est un sous-groupe distingué du groupe que l'on considère.

Définition 1.2.10

1. Une *sous-formule* d'une formule F est l'un de ses « composants », i.e. une formule à partir de laquelle F est construite. Formellement, on définit l'ensemble $\mathrm{SF}(F)$ des sous-formules de F par :
   - Si F est atomique, $\mathrm{SF}(F) = \{F\}$.
   - Si $F = F_1 \oplus F_2$ avec $\oplus \in \{\vee, \wedge, \rightarrow\}$, $\mathrm{SF}(F) = \{F\} \cup \mathrm{SF}(F_1) \cup \mathrm{SF}(F_2)$.
   - Si $F = \neg F_1$ ou $Qx\ F_1$ avec $Q \in \{\forall, \exists\}$, $\mathrm{SF}(F) = \{F\} \cup \mathrm{SF}(F_1)$.
2. Une formule F de $\mathcal{L}$ n'utilise qu'un nombre fini de symboles de $\mathcal{L}$. Ce sous-ensemble est appelé *le langage de la formule* et noté $\mathcal{L}(F)$.

Exemple 1.2.11 Les formules $\varepsilon > 0$ et $\forall x\ \{|x - x_0| < \alpha \ \rightarrow\ |f(x) - f(x_0)| < \varepsilon\}$ ou encore $|x - x_0| < \alpha$ sont, par exemple, des sous-formules de la première formule de l'exemple ci-dessus. Le langage de cette formule est formé de la constante 0, des symboles de fonction unaire f et $|.|$ et des symboles de relation binaire $>$ et $<$.

Remarque. Si une formule F est vue comme un arbre, une sous-formule de F est une formule correspondant à un sous-arbre dont la racine est un nœud de l'arbre de F.

Comme pour les termes, on sera amené, dans la suite, à définir des notions (ou à prouver des résultats) par *récurrence* sur la taille d'une formule.

Définition 1.2.12 La *taille* (ou la *longueur*) d'une formule F (notée $\tau(F)$) est le nombre de connecteurs ou de quantificateurs apparaissant dans F. Formellement :

- $\tau(F) = 0$ si F est une formule atomique ;
- $\tau(F_1 \oplus F_2) = 1 + \tau(F_1) + \tau(F_2)$ où $\oplus \in \{\vee, \wedge, \rightarrow\}$;
- $\tau(\neg F_1) = \tau(Qx\ F_1) = 1 + \tau(F_1)$ avec $Q \in \{\forall, \exists\}$).

Remarques.

1. Pour définir une fonction Φ sur les formules, il suffit de définir Φ sur les formules atomiques et de dire comment on obtient $\Phi(F_1 \oplus F_2)$ (resp. $\Phi(\neg F_1), \Phi(Qx\ F_1)$) à partir de $\Phi(F_1)$ et $\Phi(F_2)$ (resp. $\Phi(F_1)$).
2. Pour prouver une propriété P sur les formules, il suffit de prouver P pour les formules atomiques et de prouver $P(F_1 \oplus F_2)$ (resp. $P(\neg F_1), P(Qx\ F_1)$) à partir de $P(F_1)$ et $P(F_2)$ (resp. $P(F_1)$).
3. Pour prouver une propriété P sur les formules, il suffit de supposer la propriété vraie pour toutes les formules de taille $p < n$ et de la démontrer pour les formules de taille n.

C'est la traduction concrète d'une définition ou d'une preuve par récurrence sur la taille de la formule.

Définition 1.2.13 *L'opérateur principal* (on dit aussi le *connecteur principal*) d'une formule est défini par :

- Si A est atomique, alors elle n'a pas d'opérateur principal.
- Si $A = \neg B$, alors $\neg$ est l'opérateur principal de A.
- Si $A = B \oplus C$ où $\oplus \in \{\wedge, \vee, \rightarrow\}$, alors $\oplus$ est l'opérateur principal de A.
- Si $A = Qx\ B$ où $Q \in \{\forall, \exists\}$, alors Q est l'opérateur principal de A.

1.2.4 Variables libres et variables liées

La présence des quantificateurs $\forall$ et $\exists$ pose deux problèmes concernant les noms donnés aux variables.

1. On considère habituellement que, par exemple, les formules $\forall x\ (x.z = z.x)$ et $\forall y\ (y.z = z.y)$ sont les mêmes. Par contre, les formules $\forall x\ (x.y = y.x)$ et $\forall x\ (x.z = z.x)$ ne sont pas les mêmes, car l'une exprime une propriété de l'objet y et l'autre de l'objet z. Cela signifie que l'on considère que deux formules sont égales au renommage près de certaines variables (les variables qu'on appelle *liées* ou *muettes*). On travaille donc dans un quotient. Cela est très classique en mathématique, mais la définition formelle de la relation d'équivalence n'est pas aussi triviale qu'on peut le penser. Les formules $\forall x\ (x.z = z.x)$ et $\forall z\ (z.z = z.z)$ ne sont pas les mêmes : on ne peut remplacer x par z !
2. On est souvent amené à remplacer (ou substituer) une variable par un terme dans un autre terme ou dans une formule. Par exemple, si l'on définit $Z[z] : \forall x\ (x.z = z.x)$ et qu'ensuite on veut écrire $Z[2x]$, il faudra renommer la variable muette x et écrire $\forall y\ (y.2x = 2x.y)$.

Pour ne pas rendre la lecture de ce premier chapitre trop pénible, on ne formalisera pas ces deux notions (renommage et substitution). On se contentera d'en donner une définition informelle et on se reposera sur l'intuition que le lecteur a de ces notions que l'on manipule en permanence en mathématique.

Définition 1.2.14 Soit F une formule. L'ensemble $\mathrm{VL}(F)$ des *variables libres* de F est défini par récurrence sur $\tau(F)$:

- Si $F = R(t_1, ..., t_n)$ est atomique : $\mathrm{VL}(F)$ est l'ensemble des variables apparaissant dans les t_i
- Si $F = F_1 \oplus F_2$ où $\oplus \in \{\vee, \wedge, \rightarrow\}$: $\mathrm{VL}(F) = \mathrm{VL}(F_1) \cup \mathrm{VL}(F_2)$
- Si $F = \neg F_1$: $\mathrm{VL}(F) = \mathrm{VL}(F_1)$
- Si $F = Qx\ F_1$ avec $Q \in \{\forall, \exists\}$: $\mathrm{VL}(F) = \mathrm{VL}(F_1) - \{x\}$

Exemple 1.2.15

1. Soit $F : \forall x\ (x.y = y.x)$ alors $\mathrm{VL}(F) = \{y\}$.
2. Soit $G : \{\forall x \exists y\ (x.z = z.y)\} \wedge \{x = z.z\}$ alors $\mathrm{VL}(G) = \{x, z\}$.
3. Soit $H : \forall x\ (y = 0)$ alors $\mathrm{VL}(H) = \{y\}$.

On peut aussi définir ces notions de la manière suivante :

1. Une *occurrence* d'une variable x dans une formule F est une *position* de cette variable dans la formule F. Ne pas confondre avec l'*objet* qu'est la variable elle-même.
2. Une occurrence de la variable x est liée (ou muette) dans F si, dans la branche qui aboutit à la feuille où se trouve cette occurrence, on trouve $\forall x$ ou $\exists x$. Dans le cas contraire, on dit que l'occurrence est libre.
3. Une variable x est libre dans F si elle a au moins une occurrence libre. Une variable muette ou liée est une variable qui figure et n'est pas libre. On peut vérifier en exercice que : x est libre dans F ssi $x \in \mathrm{VL}(F)$.

Remarques.

1. On notera (*cf.* l'exemple ci-dessus) qu'une variable peut avoir à la fois des occurrences libres et des occurrences liées.
2. Une variable peut être liée dans F mais libre dans une sous-formule de F, par exemple y est libre dans $x.y = y.x$ mais liée dans $\forall y\ (x.y = y.x)$.
3. Le quantificateur qui lie une variable doit apparaître dans la branche de l'arbre avant l'occurrence et pas simplement avant (dans l'écriture linéaire de la formule). Par exemple : la deuxième occurrence de x est libre dans la formule $(\forall x\, x^2 > 0) \wedge x > 3$ (car le $\forall$ ne porte que sur le premier x).

Définition 1.2.16 On dit que les formules F et G sont *α-équivalentes* si elles sont (syntaxiquement) identiques à un renommage près des occurrences liées des variables.

Exemple 1.2.17 $\forall y\ (x.y = y.x)$ et $\forall z\ (x.z = z.x)$ sont α-équivalentes mais $\forall y\ (x.y = y.x)$ et $\forall y\ (z.y = y.z)$ ne le sont pas.

Remarque. On ne peut pas renommer y en x dans la formule $\forall y\ (x.y = y.x)$ et obtenir la formule $\forall x\ (x.x = x.x)$: la variable x serait *capturée*. La définition précédente est informelle et incomplète car on ne peut pas renommer les occurrences liées sans précaution : il faut éviter de *capturer* des occurrences libres. On trouvera une définition formelle en regardant l'exercice 5.9.

Pour les logiciels d'aide à l'écriture de preuves, le renommage des variables pose de difficiles problèmes. Au lieu d'utiliser des noms pour les variables, une solution consiste à utiliser des entiers appelés indices de *de Bruijn*. L'occurrence d'une variable x est remplacée par le nombre de quantificateurs qu'il faut traverser (dans la branche qui va de cette occurrence à la racine de la formule) avant de trouver le quantificateur qui lie cette variable. Malheureusement, cette notation, très commode en informatique, rend les formules illisibles pour un être humain.

Convention Désormais, on travaillera à α-équivalence près, c'est-à-dire que deux formules α-équivalentes seront considérées comme identiques. Afin de contourner le problème de capture, on évitera donc d'écrire des formules où une même variable a, simultanément, des occurrences libres et des occurrences liées. Cela est courant en mathématique : il est prudent de ne pas écrire, dans une même formule, $\sin(t)$ et $\int_0^1 cos(t)dt$.

Notation 1.2.18 Pour préciser les variables libres *possibles* d'une formule, on notera $F[x_1, ..., x_n]$. Cela signifie que les variables libres de F sont *parmi* $x_1, ..., x_n$ i.e. si y est libre dans F, alors y est l'un des x_i mais tous les x_i n'ont pas nécessairement d'occurrence libre dans F.

Définition 1.2.19

1. Une formule *close* est une formule sans variables libres. Les formules de l'exemple 1.2.7 sont toutes closes.
2. Soit F une formule dont les variables libres sont $x_1, ..., x_n$. La *clôture* (universelle) de F est la formule close $\forall x_1, ..., x_n\ F$. Il y a ici formellement un abus : on a fait comme si l'ordre des variables était fixé. Cela n'est pas gênant : en choisissant un autre ordre on obtiendrait une formule différente mais équivalente.

Remarque importante

La notion de formule définie ici est la notion de formule de la logique du premier ordre. Certaines formules (qu'on utilise couramment en mathématique) ne sont pas des formules du premier ordre. Par exemple :

- On ne peut pas écrire, par une formule du premier ordre, que « tout idéal est principal ». On serait en effet amené à quantifier à la fois sur les objets du premier ordre (les éléments de l'anneau) et sur des objets du second ordre (les parties de l'anneau).
- La formule commençant par : pour tout entier n, pour tous $x_1, ..., x_n$... (c'est, par exemple, le début de la formule qui exprime le fait qu'une partie d'un espace vectoriel est libre) n'est pas non plus une formule du premier ordre : en effet, le nombre de $\forall$ d'une formule ne saurait varier en fonction d'une variable alors qu'ici il dépend de la valeur de n. D'autre part, on devrait à nouveau quantifier sur deux types d'objets différents : les entiers et les éléments d'un espace vectoriel.

Pour pouvoir quantifier sur divers types d'objets, on est amené à introduire les logiques d'*ordre supérieur* ou les logiques *multisortes* (*cf.* chapitre 6). Cependant, les difficultés essentielles, aussi bien pour ce qui est du raisonnement que pour la théorie de la démonstration, se trouvent au niveau de la logique du premier ordre.

1.2.5 Substitutions

Une formule $F[x]$ représente une propriété de l'objet x. On veut pouvoir remplacer dans F la variable x par le terme t (on dit aussi substituer t à x) ce qu'on notera

$F[x := t]$ ou plus simplement $F[t]$ si le contexte est suffisamment clair. Deux difficultés apparaissent :

1. Il est clair que seules les occurrences libres de x doivent être remplacées. Soit $F[x]$ la formule $(\forall x\ x^2 > 0) \wedge (x < 1)$. F signifie : le carré de tout élément est positif et l'objet x est plus petit que 1. Si $t = \sin(y), F[t]$ signifie donc : le carré de tout élément est positif et $\sin(y)$ est plus petit que 1. Écrire pour $F[t]$ la formule $(\forall x\ \sin^2(y) > 0) \wedge\ \sin(y) < 1$ ne correspond donc pas à ce que l'on souhaite. Écrire $(\forall \sin(y)\ \sin^2(y) > 0) \wedge\ \sin(y) < 1$ serait pire ! (immédiatement après $\forall$ il ne peut y avoir qu'une variable).
2. Il y a, là encore, un problème de capture. Un étudiant sait bien que si la fonction f est définie par $f(x) = \int_0^1 e^{(x+y)^2}\ dy$ alors $f(y^2)$ n'est pas $\int_0^1 e^{(y^2+y)^2}\ dy$ mais c'est $\int_0^1 e^{(y^2+z)^2}\ dz$. Un changement de nom de la variable d'intégration a été nécessaire pour éviter la capture de la variable y (qui doit bien sûr être libre dans le résultat) par le lieur qu'est le signe d'intégration. On a ici exactement le même problème : si $F[x]$ est $\forall y\ (y.x = x.y)$, $F[y^{-1}]$ n'est pas $\forall y\ (y.y^{-1} = y^{-1}.y)$ mais $\forall z\ (z.y^{-1} = y^{-1}.z)$.

Définition 1.2.20 Soit F une formule, x une variable et t un terme. $F[x := t]$ est la formule obtenue en remplaçant dans F toutes les occurrences libres de x par t, après renommage éventuel des occurrences de variables liées de F qui apparaissent libres dans t.

Remarques.

1. Le renommage a pour but d'éviter la capture de variables. Il est clair que les seules captures possibles sont celles décrites dans la définition ci-dessus. S'il n'y a pas de capture, il n'est bien sûr pas nécessaire de faire un renommage.
2. Pour éviter les captures, il suffit d'appliquer la convention de la page 21 à l'expression $F[x := t]$. On prend donc soin de renommer les variables liées de F pour qu'aucune variable n'ait des occurrences libres et d'autres liées.
3. On peut aussi définir la substitution simultanée de $t_1, ..., t_n$ à $x_1, ..., x_n$. On la notera $F[x_1 := t_1, ..., x_n := t_n]$ ou, plus simplement, $F[t_1, ..., t_n]$ s'il n'y a pas d'ambiguïté.
4. *Attention* : Les formules $F[x_1 := t_1, x_2 := t_2]$ et $F[x_1 := t_1][x_2 := t_2]$ ne sont pas, en général, équivalentes (voir lemme 1.6.5). La première substitution est simultanée, la deuxième correspond à deux substitutions consécutives.

1.2.6 Le calcul propositionnel

Si les seuls symboles de relation du langage sont des relations d'arité 0 (même le symbole = est absent), les quantificateurs sont alors inutiles (puisqu'une formule ne peut pas contenir de variables). On obtient alors le *calcul propositionnel* défini ci-dessous.

Définition 1.2.21 L'ensemble $\mathcal{C}_P$ des formules du *calcul propositionnel* est défini par la grammaire (où $\mathcal{V}_P$ est l'ensemble des relations d'arité 0) :

$$\mathcal{C}_P = \mathcal{V}_P \mid \bot \mid \neg\mathcal{C}_P \mid \mathcal{C}_P \vee \mathcal{C}_P \mid \mathcal{C}_P \wedge \mathcal{C}_P \mid \mathcal{C}_P \to \mathcal{C}_P$$

Remarque. On rappelle que cette notation signifie que les formules sont obtenues à partir des variables et de $\bot$ par utilisation de $\neg$, $\vee$, etc. Les relations d'arité 0 sont souvent appelées *variables propositionnelles*. Ce n'est pas très judicieux puisqu'on utilise ainsi le mot « variable », qui représente un objet, pour quelque chose qui est une formule. On adoptera quand même cette terminologie très classique.

Exemple 1.2.22 Les formules ci-dessous sont des formules du calcul propositionnel ayant comme variables propositionnelles X, Y et Z.

$$(X \to Y \vee Z) \wedge \{(X \to \bot) \vee (X \to \neg Z)\}$$
$$(X \wedge \neg Y \to Z) \vee \{(\bot \to Z) \vee (Y \to Z)\}$$

1.3 LES DÉMONSTRATIONS EN DÉDUCTION NATURELLE

1.3.1 Introduction

Les démonstrations qu'on trouve dans les livres de mathématiques (et donc, en particulier, dans celui-ci) sont des assemblages de symboles mathématiques et de phrases contenant des « mots clés » tels que : *donc, parce que, si, si et seulement si, il est nécessaire que, il suffit de, prenons un x tel que, supposons que, cherchons une contradiction, etc.* Ces mots sont supposés être compris par tous de la même manière, ce qui n'est, en fait, pas toujours le cas. On verra, par exemple, plus loin que l'expression « soit x tel que » peut vouloir dire deux choses très différentes et qu'il peut être très fâcheux de se tromper.

Dans un livre ou un cours, le but d'une démonstration est de convaincre le lecteur de la vérité de l'énoncé. Suivant le niveau du lecteur, cette démonstration sera plus ou moins détaillée : quelque chose qui pourra être considéré comme évident dans un cours de maîtrise pourra ne pas l'être en licence. Tout comme en collège on détaillera un calcul, alors qu'au lycée ce calcul sera fait de tête et ne nécessitera pas d'explication.

Dans un devoir, le correcteur sait que le résultat demandé à l'étudiant est vrai et il en connaît la démonstration. L'étudiant doit convaincre le correcteur qu'il sait démontrer (correctement) le résultat demandé. Le niveau de détail qu'il doit donner dépend donc de la confiance qu'aura le correcteur : dans une bonne copie, une « preuve par une récurrence évidente » passera bien, alors que dans une copie où il y a eu auparavant un « évident », qui était évidemment... faux, ça ne passera pas !

Pour pouvoir gérer convenablement le niveau de détail, il faut savoir ce qu'est une démonstration complète. Ce travail de formalisation a été fait au début du siècle. On donne ici la notion de démonstration en *déduction naturelle*. On verra d'autres formalisations à la fin de ce chapitre et au chapitre 5.

Plusieurs choses peuvent paraître surprenantes.

– Il n'y a qu'un nombre fini de règles : deux pour chacun des connecteurs (et l'égalité) plus trois règles générales. Il n'était pas du tout évident, *a priori*, qu'un nombre *fini* de règles soit suffisant pour démontrer tout ce qui est vrai. On montrera ce résultat (c'est, essentiellement, le théorème de complétude) et on en précisera le sens dans le chapitre 2. La preuve n'en est pas du tout triviale.

– Ce sont les mêmes règles pour toutes les mathématiques : algèbre, analyse, géométrie, etc. Cela veut dire qu'on a réussi à isoler tout ce qui est général dans un raisonnement. On verra plus loin qu'une démonstration est un assemblage de couples (Γ, A), où Γ est un ensemble de formules (les hypothèses) et A une formule (la conclusion). Quand on fait de l'arithmétique, de la géométrie ou de l'analyse réelle, on utilise, en plus des règles, des hypothèses que l'on appelle des *axiomes*. Ceux-ci expriment les propriétés particulières des objets qu'on manipule. Par exemple, dans $\mathbb{N}$, tout élément a un successeur ; dans $\mathbb{R}$, entre deux éléments, il y en a toujours un autre. Quelques exemples de *théories* seront étudiés au chapitre 3.

1.3.2 Les séquents

On démontre, en général, des formules en utilisant un ensemble d'hypothèses, et cet ensemble peut varier au cours de la démonstration : quand on dit « supposons F et montrons G », F est alors une nouvelle hypothèse que l'on pourra utiliser pour montrer G. Pour formaliser cela, on introduit la notion de séquent.

Définition 1.3.1 Un *séquent* est un couple (noté $\Gamma \vdash F$) où :

- Γ est un ensemble *fini* de formules. Γ représente les hypothèses que l'on peut utiliser. Cet ensemble s'appelle aussi le *contexte* du séquent.
- F est une formule. C'est la formule que l'on veut montrer. On dira que cette formule est la *conclusion* du séquent.

Remarque. Il est important de noter qu'on *ne demande pas* que les formules Γ, F soient *closes* : on verra en effet, par exemple, que pour prouver la formule close $\forall x\, F[x]$, il faut, si x n'est pas libre dans les hypothèses, prouver la formule, non close, $F[x]$.

Notation 1.3.2 Si $\Gamma = \{A_1, ..., A_n\}$ on pourra noter $A_1, ..., A_n \vdash F$ au lieu de $\Gamma \vdash F$. Le signe $\vdash$ se lit *thèse* ou *démontre*. On notera $\vdash F$ un séquent dont l'ensemble d'hypothèses est vide et $\Gamma_1, ..., \Gamma_n \vdash F$ un séquent dont l'ensemble d'hypothèses est $\bigcup_{1 \leqslant i \leqslant n} \Gamma_i$.

On notera, en particulier, que dans le séquent $\Gamma, A \vdash B$ la formule A peut être déjà dans Γ, puisque ce contexte est $\Gamma \cup \{A\}$.

Définition 1.3.3 Un séquent $\Gamma \vdash F$ est *prouvable* (ou *démontrable* ou *dérivable*) s'il peut être obtenu par une application finie de règles décrites dans la section suivante. Une formule F est *prouvable* si le séquent $\vdash F$ est prouvable.

Remarques.

1. « $\Gamma \vdash F$ » représente à la fois le séquent et la phrase « le séquent $\Gamma \vdash F$ est prouvable ». Il n'y aura, en général, pas d'ambiguïté.
2. On écrira $\Gamma \nvdash F$ pour dire « le séquent $\Gamma \vdash F$ n'est pas prouvable ».
3. Il existe des systèmes de démonstration qui n'utilisent pas ce principe hypothèses/conclusion. On en présentera un dans la section 1.7.

1.3.3 Les règles de démonstration

Les règles de démonstration sont les briques qui permettent de construire les dérivations. Une dérivation formelle est un assemblage fini (et correct !) de règles. Cet assemblage n'est pas linéaire (ce n'est pas une suite) mais un arbre. On est en effet souvent amené à faire des branchements : par exemple, pour prouver $A \wedge B$, on doit faire deux choses (prouver A et prouver B).

On présente ci-dessous un choix de règles. On aurait pu en présenter d'autres (à la place ou en plus) qui donneraient la même notion de prouvabilité. Celles que l'on a choisies sont « naturelles » et correspondent aux raisonnements que l'on fait habituellement en mathématique. Dans la pratique courante on utilise, en plus des règles ci-dessous, beaucoup d'autres règles mais celles-ci peuvent se déduire des précédentes. On les appellera *règles dérivées*. On en verra quelques exemples dans la section 1.3.8.

Il est de tradition d'écrire la racine de l'arbre (le séquent conclusion) en bas, les feuilles en haut : la nature est faite ainsi ! Pourtant, on construit souvent l'arbre en allant de la racine vers les feuilles. Comme il est également de tradition d'écrire, sur une feuille de papier, de haut en bas, il ne serait pas déraisonnable d'écrire la racine en haut et les feuilles en bas. Il faut faire un choix : on a adopté ici le choix le plus répandu dans les livres de logique.

a) Comment lire les règles ?

1. Une règle se compose :
   - d'un ensemble de *prémisses* : chacune d'elles est un séquent. Il peut y en avoir zéro, une ou plusieurs et leur ordre est sans importance ;
   - du séquent *conclusion* de la règle ;
   - d'une barre horizontale séparant les prémisses (en haut) de la conclusion (en bas). Sur la droite de la barre, on indiquera le nom de la règle.

   Exemple :
   $$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} \to_{\mathrm{e}}$$
   Cette règle a deux prémisses ($\Gamma \vdash A \to B$ et $\Gamma \vdash A$) et une conclusion ($\Gamma \vdash B$). Le nom abrégé de cette règle est $\to_{\mathrm{e}}$.
2. Chaque règle peut se lire de deux manières :
   - *de bas en haut* : si on veut prouver la conclusion, il suffit, par utilisation de la règle, de prouver les prémisses. C'est ce qu'on fait quand on cherche une démonstration. Cela correspond à *l'analyse* ;

– *de haut en bas* : si on a prouvé les prémisses, alors on a aussi prouvé la conclusion. C'est ce qu'on fait quand on rédige une démonstration. Cela correspond à la *synthèse*.

Le texte informel qui suit chaque règle en donne le sens intuitif et, éventuellement, quelques commentaires. Il est tantôt rédigé pour une lecture de haut en bas, tantôt de bas en haut. Cela vient du fait que certaines règles sont plus facilement compréhensibles dans un sens que dans l'autre.

3. À chaque symbole logique correspondent deux types de règles :

– les *règles d'introduction* qui permettent de *prouver* une formule ayant ce symbole comme opérateur principal ;

– les *règles d'élimination* qui permettent d'*utiliser* une formule ayant ce symbole comme opérateur principal.

4. L'égalité ayant un statut particulier, on lui associe également deux règles. On verra au chapitre 3 (*cf.* section 3.2.1) une autre manière d'utiliser le statut particulier de ce symbole.

5. Il y a, en plus de ces règles, trois autres règles.
Les deux premières (axiome et affaiblissement) ne correspondent à aucun connecteur. La première, la seule à ne pas avoir de prémisses, correspond donc à une brique terminale du morceau de démonstration qui la contient (une feuille de l'arbre). La deuxième exprime simplement le fait que, dans un morceau de démonstration, certaines hypothèses peuvent ne pas servir.
La troisième (absurdité classique) a été mise avec les deux règles concernant le connecteur $\neg$. Elle a pourtant un statut particulier : c'est elle, et elle seule, qui correspond au raisonnement par l'absurde. Son rôle apparaîtra mieux dans le chapitre 4 où on étudiera une notion de démonstration sans cette règle.

b) Les règles

Axiome

$$\frac{}{\Gamma, A \vdash A}\ \text{ax}$$

Si la conclusion du séquent est l'une des hypothèses, alors le séquent est prouvable.

Affaiblissement

$$\frac{\Gamma \vdash A}{\Gamma, B \vdash A}\ \text{aff}$$

De haut en bas : si on peut démontrer A sous les hypothèses Γ alors, en ajoutant des hypothèses supplémentaires, on peut encore démontrer A.
De bas en haut : il y a des hypothèses qui peuvent ne pas servir.

Introduction de l'implication

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_{\text{i}}$$

De bas en haut : pour montrer $A \to B$, on suppose A (c'est-à-dire qu'on l'ajoute aux hypothèses) et on démontre B.

Élimination de l'implication Cette règle est connue depuis très longtemps sous le nom de *modus ponens* :

$$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} \to_{\mathrm{e}}$$

De bas en haut : pour démontrer B, si on connaît un théorème de la forme $A \to B$ ou si on peut démontrer le lemme $A \to B$, il suffit de démontrer A.

Introduction de la conjonction

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge_{\mathrm{i}}$$

De bas en haut : pour montrer $A \wedge B$, il suffit de montrer A et de montrer B.

Élimination de la conjonction

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge_{\mathrm{e}}^{g} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge_{\mathrm{e}}^{d}$$

De haut en bas : de $A \wedge B$, on peut déduire A (élimination gauche) et B (élimination droite).

Introduction de la disjonction

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee_{\mathrm{i}}^{g} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vee_{\mathrm{i}}^{d}$$

De bas en haut : pour démontrer $A \vee B$, il suffit de démontrer A ou de démontrer B. Cette règle peut paraître bizarre car la conclusion de la règle est plus faible que la prémisse. Les exemples suivants devraient aider à la comprendre.

Imaginons qu'on ait un théorème de la forme $\forall x \ \{(x < 4) \vee (x = 4) \to A\}$ et que pour prouver un résultat il suffit d'appliquer ce théorème avec $x = \pi$. On a, formellement, besoin de vérifier que π est inférieur ou égal à 4. On sait que π est (strictement) inférieur à 4, mais la seule chose qui importe ici pour pouvoir appliquer le théorème, est qu'il est inférieur ou égal à 4. C'est, formellement, la règle $\vee_{\mathrm{i}}^{d}$ qui permet de le faire. Imaginons qu'on doive prouver une propriété de la forme : pour tout entier n, $u_n \leqslant 4$ et, pour le prouver, on est amené à distinguer le cas n pair et le cas n impair (c'est le cas, par exemple, dans l'étude d'une suite récurrente). Il se peut qu'on ait $u_{2p+1} = 4$ et qu'on prouve $u_{2p} < 4$ par récurrence sur p mais que la récurrence $u_{2p} \leqslant 4$ ne marche pas ! Cette règle est alors également nécessaire.

Élimination de la disjonction

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C} \vee_{\mathrm{e}}$$

De bas en haut : si on veut montrer C et qu'on sait qu'on a $A \vee B$, il suffit de le montrer, d'une part en supposant A, d'autre part en supposant B.

C'est un raisonnement par cas. Par exemple, si on a à montrer une propriété sur un entier n, on pourra distinguer les cas n pair et n impair. On pourra aussi distinguer n premier et n composé.

Introduction de la négation

$$\frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg_{\mathrm{i}}$$

De bas en haut : pour montrer $\neg A$, on suppose A et on démontre l'absurde. Voir aussi le commentaire après la règle $\bot_c$.

Élimination de la négation

$$\frac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash \bot}\ \neg_{\mathrm{e}}$$

De haut en bas : si on a montré $\neg A$ et A, alors on a montré l'absurde. Voir aussi le commentaire après la règle $\bot_c$.

Absurdité classique

$$\frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ \bot_c$$

De bas en haut : pour démontrer A, il suffit de démontrer l'absurde en supposant $\neg A$.

Les deux règles $\neg_{\mathrm{e}}$ et $\neg_{\mathrm{i}}$ mettent en valeur le rôle du symbole $\bot$ qui représente, intuitivement, le faux « universel » indépendamment du langage qu'on utilise.

Quand, dans une démonstration, on dit qu'on a obtenu une contradiction, c'est qu'on a montré quelque chose de « manifestement faux » : une formule et sa négation (*cf.* la règle $\neg_{\mathrm{e}}$), $0 = 1$ ou tout autre formule liée au langage utilisé. On verra plus loin (*cf.* remarque page 31) que $\neg A$ est équivalent à $A \to \bot$: on pourrait donc ne pas utiliser le connecteur $\neg$ et supprimer les deux règles correspondantes.

La règle $\bot_c$ correspond au raisonnement par l'absurde. On verra (*cf.* lemme 4.3.3) que cette règle revient dire que $\neg\neg A$ est équivalent à A : « A est vraie ssi il est faux que A soit fausse ». Cette règle ne va pas de soi : elle est nécessaire pour prouver certains résultats, i.e. il y a des résultats qu'on ne peut pas prouver si on n'a pas le droit de faire un raisonnement par l'absurde.

On notera enfin que, contrairement à beaucoup d'autres, cette règle peut être appliquée à tout moment : on peut, en effet, toujours dire « pour prouver A, je suppose $\neg A$ et je vais chercher une contradiction ». Cela rend la recherche de preuve (un peu plus) difficile.

Introduction du quantificateur universel

$$\frac{\Gamma \vdash A \quad x \text{ n'est pas libre dans les formules de } \Gamma}{\Gamma \vdash \forall x\ A}\ \forall_{\mathrm{i}}$$

De bas en haut : pour démontrer $\forall x\ A$, il suffit de montrer A en ne faisant aucune hypothèse sur x.

Pour montrer $\forall x\ A$, il faut montrer A pour un objet x quelconque. La condition « x n'est pas libre dans les formules de Γ » est la manière formelle de dire « x est quelconque ». Elle signifie qu'on n'a rien supposé sur x puisque x n'a pas le droit d'apparaître dans les hypothèses.

Cette condition est cruciale, et l'oubli de cette vérification est souvent source d'erreurs.

Si cette condition n'est pas satisfaite, il faut alors se souvenir qu'une formule n'est définie qu'au renommage près de ses variables liées : la formule $\forall y\ A[x := y]$ est identique à $\forall x\ A$. Il suffit donc de renommer x, dans A, en une autre lettre y qui satisfera la condition et prouver $A[x := y]$.

Élimination du quantificateur universel

$$\frac{\Gamma \vdash \forall x\ A}{\Gamma \vdash A[x := t]}\ \forall_{\mathrm{e}}$$

De haut en bas : de $\forall x\ A$, on peut déduire $A[x := t]$ pour n'importe quel terme t. Ce qu'on peut aussi dire sous la forme : si on a prouvé A pour tout x, alors on peut utiliser A avec n'importe quel objet t. On rappelle que les termes désignent les objets sur lesquels on travaille et que, dans la substitution, il faut, si nécessaire, renommer les variables liées de A pour éviter la capture des variables de t. Ceci est également valable pour la règle suivante.

Introduction du quantificateur existentiel

$$\frac{\Gamma \vdash A[x := t]}{\Gamma \vdash \exists x\ A}\ \exists_{\mathrm{i}}$$

De bas en haut : pour démontrer $\exists x\ A$, il suffit de trouver un objet (i.e. un terme) t pour lequel on sait montrer $A[x := t]$.

Élimination du quantificateur existentiel

$$\frac{\Gamma \vdash \exists x\ A \quad \Gamma, A \vdash C \quad x \text{ n'est libre ni dans les formules de } \Gamma, \text{ ni dans } C}{\Gamma \vdash C}\ \exists_{\mathrm{e}}$$

De bas en haut : quand on a une hypothèse de la forme $\exists x\ A$, on peut utiliser cette hypothèse en « prenant » un x qui satisfait A. Formellement, « prendre » signifie qu'on lui donne un nom.

La condition sur x est la formalisation du fait que l'objet dont l'hypothèse affirme l'existence n'a aucune raison d'être l'un des objets qui ont déjà été introduits au cours de la démonstration. Elle est donc cruciale et son oubli est souvent source d'erreurs. Comme dans le cas de la règle $\forall_{\mathrm{i}}$, si la condition n'est pas satisfaite, il suffira de faire un renommage de x dans la formule $\exists x\ A$, i.e. de donner un nouveau nom à x.

Remarques.

Les deux règles $\forall_{\mathrm{i}}$ et $\exists_{\mathrm{e}}$ sont les seules qui permettent d'introduire de nouveaux objets.

La condition sur la variable dans les règles $\forall_{\mathrm{i}}$ et $\exists_{\mathrm{e}}$ peut s'écrire dans les deux cas « x n'est pas libre dans le séquent conclusion de la règle ».