

DÉDUCTION NATURELLE

À la fin de ce chapitre, je sais :

- ─ lire un séquent
- ─ décrire les règles d'introduction et d'élimination
- ─ justifier les principaux raisonnements de la logique classique
- ─ construire un arbre de preuve démontrant une formule simple

A Déduction naturelle

Explorer l'arbre syntaxique d'une formule logique s'avère être une tâche dont la complexité est exponentielle dans le pire des cas, $O(2^n)$ si la formule comporte n variables. Vérifier qu'une formule logique est une tautologie est faisable mais pas toujours en temps humain. On cherche donc un moyen de prouver qu'une formule logique est vraie non pas en testant toutes les valuations possibles mais en construisant une preuve, c'est-à-dire une suite d'opérations purement logiques.

La déduction naturelle a été développée dans ce but et est un des premiers systèmes de preuve ayant été mis en œuvre. Il s'agit d'établir des règles d'inférence élémentaires permettant de prouver des formules d'une logique donnée, élémentaires à analyser et à implémenter, dans le but d'obtenir un programme d'assistant de preuve, voire de preuve automatique.

■ **Définition 1 — Séquent ou jugement.** Soit \mathcal{F} l'ensemble des formules logiques, Γ une partie de \mathcal{F} (les hypothèses) et a une formule logique (la conclusion). Un séquent est une relation binaire entre l'ensemble $\mathcal{P}(\mathcal{F})$ et \mathcal{F} . On la note ainsi :

$$\Gamma \vdash a \tag{1}$$

Elle signifie que l'on peut déduire a en utilisant uniquement les hypothèses Γ : de Γ on peut conclure a .

■ **Exemple 1 — Séquent simple.** Voici un exemple de séquent valide :

$$\forall x \in \mathbb{R}, x^2 - 10x + 21 = 0 \vdash x = 3 \vee x = 7 \tag{2}$$

Voici un exemple de séquent non valide, car la conclusion n'est pas vérifiée pour ces hypothèses :

$$\forall x \in \mathbb{R}_+, x^2 - 4x - 21 = 0 \vdash x = -3 \tag{3}$$

Par contre, ce dernier est valide :

$$\forall x \in \mathbb{R}_+, x^2 - 4x - 21 = 0 \vdash x = 7 \tag{4}$$

■ **Définition 2 — Déduction naturelle.** La déduction naturelle est un système de déduction qui permet de déterminer si des séquents sont **prouvables** ou non. Elle met en valeur le raisonnement «naturel» d'une preuve mathématiques et s'appuie sur une ensemble de **règles** qu'il s'agit de définir afin de pouvoir construire les preuves comme des emboitements de règles d'inférence.

La déduction naturelle organise une démonstration sous la forme d'un **arbre** dont la racine est le séquent à démontrer. Les nœuds de l'arbre se déduisent les uns des autres pas à pas, de manière quasi-évidente via l'introduction ou l'élimination de règles d'inférence élémentaires : la conclusion d'une branche devient une hypothèse du niveau inférieur. Les feuilles sont des axiomes, des introductions de constantes logiques ou des hypothèses, dans tous les cas, des règles sans conditions.

■ **Définition 3 — Règle d'inférence ou règle de déduction.** Une règle d'inférence en déduction naturelle est un ensemble de séquents, les hypothèses (H) ou prémisses, suivi d'un autre séquent conclusion (C). On la représente généralement sous la forme de Gentzen :

$$\frac{\Gamma \vdash H_1 \quad \Gamma \vdash H_2 \quad \dots \quad \Gamma \vdash H_n}{\Gamma \vdash C}$$

■ **Définition 4 — Axiome.** Un axiome est une règle d'inférence pour laquelle l'ensemble des hypothèses est vide.

$$\frac{}{\Gamma \vdash a} \text{ax}$$

■ **Définition 5 — Arbres de preuve ou arbres de dérivation (définition inductive).** L'ensemble des arbres de preuve \mathcal{A} d'un séquent s par déduction naturelle est soit :

(une feuille) l'application d'un **axiome** dont la conclusion est s ,

(un nœud) l'application d'une règle d'inférence (R) dont la conclusion est s et dont les prémisses dérivent d'éléments de \mathcal{A} par des règles d'inférence ^a.

a. Ces prémisses sont les conclusions d'arbres de preuve

R Une règle d'inférence forme un constructeur de l'ensemble inductif des arbres de preuve.

La déduction naturelle comporte une dizaine de règles d'inférences qui permettent de construire un arbre de preuve. On distingue les règles qui introduisent une conséquence de plusieurs séquents des règles qui éliminent des séquents en réduisant les conséquences.

B Règles d'introduction et d'élimination

a Introduction et élimination de la conjonction

Lorsqu'on connaît une preuve de la formule a et une preuve de la formule b , alors on peut construire une preuve de la formule $a \wedge b$.

$$\frac{\Gamma \vdash a \quad \Gamma \vdash b}{\Gamma \vdash a \wedge b} \wedge_i$$

On dit qu'on a **introduit** la conjonction et on note cette règle \wedge_i . De même, si on connaît une preuve de $a \wedge b$, alors on peut construire une preuve de a ou de b en **éliminant** la conjonction.

$$\frac{\Gamma \vdash a \wedge b}{\Gamma \vdash a} \wedge_e$$

■ **Exemple 2 — L'opérateur \wedge est commutatif.** On peut montrer que la conjonction est commutative :

$$\frac{\frac{\frac{\Gamma \vdash a \wedge b}{\Gamma \vdash b} \wedge_e \quad \frac{\frac{\Gamma \vdash a \wedge b}{\Gamma \vdash a} \wedge_e}{\Gamma \vdash a \wedge b} \wedge_i}{\Gamma \vdash b \wedge a}}{\Gamma \vdash a \wedge b}$$

b Introduction et élimination de l'implication

Pour introduire l'implication, on suppose que b peut être déduit de a , alors il est possible de déduire l'implication $a \rightarrow b$ en se passant de l'hypothèse a .

$$\frac{\Gamma, a \vdash b}{\Gamma \vdash a \rightarrow b} \rightarrow_i$$

Pour déduire une formule d'une implication, on suppose qu'on peut justifier a et l'implication. On dispose alors d'une preuve de b :

$$\frac{\Gamma \vdash a \rightarrow b \quad \Gamma \vdash a}{\Gamma \vdash b} \rightarrow_e$$

(R)

Dans l'antiquité, cette règle de l'élimination de l'implication \rightarrow_e était nommée *modus ponens*. On la désigne aussi parfois sous le nom de *détachement*. L'implication et le fait de poser a permettent de poser (ou détacher) b .

■ **Exemple 3 — Preuve de $\vdash p \rightarrow p$.** On donne ci-dessous la preuve que l'implication matérielle est réflexive.

$$\frac{\overline{p \vdash p} \text{ ax}}{\vdash p \rightarrow p} \rightarrow_i$$

■ **Exemple 4 — Preuve de $p \wedge q \vdash p \rightarrow q$.** Pour construire cet arbre de preuve, on utilise l'introduction de l'implication et l'élimination de la conjonction.

$$\frac{\frac{\overline{p \wedge q, p \vdash p \wedge q} \text{ ax}}{\frac{\overline{p \wedge q, p \vdash q} \wedge_e}{\overline{p \wedge q \vdash p \rightarrow q} \rightarrow_i}}}{p \wedge q \vdash p \rightarrow q}$$

c Introduction et élimination de la disjonction

De la même manière, on introduit et on élimine la disjonction. Lorsqu'on connaît une preuve de la formule a et une preuve de la formule b , alors on peut construire une preuve de la formule $a \vee b$. On peut écrire soit

$$\frac{\Gamma \vdash a}{\Gamma \vdash a \vee b} \vee_i$$

soit

$$\frac{\Gamma \vdash b}{\Gamma \vdash a \vee b} \vee_i$$

puisque la disjonction n'exige nullement que les deux soient vraies pour être vraie.

La déduction d'une disjonction est possible s'il existe une formule commune que l'on peut déduire des deux formules de la disjonction.

$$\frac{\Gamma \vdash a \vee b \quad \Gamma, a \vdash c \quad \Gamma, b \vdash c}{\Gamma \vdash c} \vee_e$$

■ **Exemple 5 — Preuve que la disjonction est commutative.** On construit la preuve du séquent $p \vee q \vdash q \vee p$.

$$\frac{}{p \vee q \vdash p \vee q} \text{ax} \quad \frac{\frac{p \vee q, p \vdash p}{p \vee q, p \vdash q \vee p} \text{ax} \quad \frac{p \vee q, q \vdash q}{p \vee q, q \vdash q \vee p} \text{ax}}{p \vee q \vdash q \vee p} \vee_e$$

d Introduction et élimination de la négation

Le fait que la négation d'une formule soit vraie lorsque cette formule est fausse nous permet de justifier la négation en montrant que la formule conduit à la contradiction.

$$\frac{\Gamma, a \vdash \perp}{\Gamma \vdash \neg a} \neg_i$$

Symétriquement, l'élimination de la négation conduit à une contradiction.

$$\frac{\Gamma \vdash \neg a \quad \Gamma \vdash a}{\Gamma \vdash \perp} \neg_e$$

R La contradiction peut donc être engendrée par une proposition et son contraire. C'est le seul moyen d'introduire \perp dans un séquent. On pourrait donc noter l'élimination de la négation \perp_i . Il est également possible d'éliminer la contradiction en utilisant le **principe d'explosion**.

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash a} \perp_e$$

Ce principe peut-être démontré si on accepte le raisonnement par l'absurde^a. Pour la logique intuitionniste, c'est donc un axiome.

^a Les savants de l'antiquité avaient trouvé ce principe. Mais cela a été prouvé au XII^e siècle par Guillaume de Soisson : la démonstration s'appuie sur l'hypothèse qu'on ne peut prouver une formule et son contraire. Elle utilise un syllogisme disjonctif pour introduire \perp et conclut par un raisonnement par l'absurde.

C Synthèse des règles de la déduction naturelle

Le tableau 1 rassemble les règles de construction de la déduction naturelle. Elles permettent de construire des arbres de preuves. Il s'agit de comprendre ces règles avant de les apprendre en les utilisant sur des démonstrations simples.

Formule	Introduction	Élimination
\top	$\frac{}{\Gamma \vdash \top} \top_i$	
\perp		$\frac{\Gamma \vdash \perp}{\Gamma \vdash a} \perp_e$ (Principe d'explosion)
$a \in \Gamma$ (Axiome)	$\frac{}{\Gamma \vdash a} ax$	
Conjonction	$\frac{\Gamma \vdash a \quad \Gamma \vdash b}{\Gamma \vdash a \wedge b} \wedge_i$	$\frac{\Gamma \vdash a \wedge b}{\Gamma \vdash a} \wedge_e$
Disjonction	$\frac{\Gamma \vdash a}{\Gamma \vdash a \vee b} \vee_i$ et / ou $\frac{\Gamma \vdash b}{\Gamma \vdash a \vee b} \vee_i$	$\frac{\Gamma \vdash a \vee b \quad \Gamma, a \vdash c \quad \Gamma, b \vdash c}{\Gamma \vdash c} \vee_e$
Implication	$\frac{\Gamma, a \vdash b}{\Gamma \vdash a \rightarrow b} \rightarrow_i$	$\frac{\Gamma \vdash a \rightarrow b \quad \Gamma \vdash a}{\Gamma \vdash b} \rightarrow_e$ (Modus ponendo ponens)
Négation	$\frac{\Gamma, a \vdash \perp}{\Gamma \vdash \neg a} \neg_i$	$\frac{\Gamma \vdash \neg a \quad \Gamma \vdash a}{\Gamma \vdash \perp} \neg_e$ (Introduction de la contradiction \perp_i)

TABLE 1 – Ensemble des règles de la déduction naturelle

D Correction de la déduction naturelle

La sémantique des formules logiques et la déduction naturelle constituent deux points de vue sur ce que pourrait être la *vérité* en logique des propositions. La sémantique s'appuie sur des valuations tandis que la déduction cherche à construire le raisonnement qui prouve la formule. En fait, si une proposition est prouvable sous une certaine hypothèse, alors cette proposition est une conséquence sémantique de cette hypothèse et réciproquement.

Théorème 1 — Équivalence entre prouvabilité et conséquence sémantique. En logique des propositions, toute conséquence sémantique est prouvable et toute formule logique prouvable est une conséquence sémantique.

Formulé autrement, si tout modèle de Γ est un modèle de $a \in \mathcal{F}$ alors on peut déduire a de Γ et réciproquement.

Plus formellement :

$$\Gamma \models a \iff \Gamma \vdash a \tag{5}$$

Démonstration. On procède par double implication.

(\Leftarrow) Dans ce sens, on procède par induction structurelle sur l'arbre de preuve (cf. définition 5). La

démonstration s'appuie sur la définition des règles d'inférence : pour chaque règle, on montre que si $\Gamma \vdash a$ alors on a $\Gamma \models a$.

Cas de base, Axiome On suppose que l'on peut déduire a de Γ et que a est un axiome, aucune hypothèse n'est nécessaire pour prouver a . Un modèle de Γ est donc un modèle de a . Donc, $\Gamma \models a$.

(\top_i) Si $\Gamma \vdash \top$, comme \top est une constante et la constante associée au vrai, tout modèle de Γ la satisfait. On a donc $\Gamma \models \top$.

(\perp_e) Supposons que $\Gamma \vdash \perp$. \perp est une constante mais aucun modèle de Γ ne la satisfait, par définition. D'après la définition de la conséquence sémantique, a est une conséquence sémantique de Γ si tout modèle de Γ est un modèle de a . Comme l'ensemble des modèles de \perp est l'ensemble vide, alors il n'existe pas de modèle de Γ qui ne soit pas un modèle de \perp . Donc, $\Gamma \models \perp$.

(\wedge_i) On suppose que $\Gamma \vdash a_1$ et que $\Gamma \vdash a_2$. On s'appuie sur le cas de l'axiome et on en déduit donc que $\Gamma \models a_1$ et $\Gamma \models a_2$. Tout modèle de Γ est donc à la fois un modèle de a_1 et un modèle de a_2 . Ce qui signifie que $\Gamma \models a_1 \wedge a_2$.

(\wedge_e) On suppose que $\Gamma \vdash a_1 \wedge a_2$, Γ et donc Γ est un modèle de a_1 et un modèle de a_2 . Qui peut le plus peut le moins, Γ est donc un modèle de a_1 .

(\vee_i) On suppose que $\Gamma \vdash a_1$. Tout modèle de Γ est donc un modèle de a_1 . Même s'il n'est pas un modèle de a_2 , d'après la définition de la disjonction, ce modèle est un modèle de $a_1 \vee a_2$. Ce qui signifie que $\Gamma \models a_1 \vee a_2$.

(\vee_e) On suppose que $\Gamma \vdash a_1 \vee a_2$, $\Gamma, a_1 \vdash c$ et $\Gamma, a_2 \vdash c$. Soit un modèle de Γ et a_1 , alors ce modèle est un modèle de c . On procède de même avec un modèle de Γ et a_2 . Donc, un modèle de $\Gamma, a_1 \vee a_2$ est un modèle de c , d'après la définition de la disjonction. Ce qui signifie que $\Gamma, a_1 \vee a_2 \models c$.

(\rightarrow_i) D'après l'hypothèse et les cas précédents, on a $\Gamma, a_1 \models a_2$ et on cherche à montrer que $\Gamma \models a_1 \rightarrow a_2$. Plusieurs cas sont possibles :

- Si a_1 est vraie, alors pour tout modèle de Γ , a_2 est vraie.
- Si a_1 est fausse, $a_1 \rightarrow a_2$ est toujours vraie, car ex falso quodlibet.

Donc l'implication $a_1 \rightarrow a_2$ est vraie (d'après sa table de vérité) pour tout modèle de Γ . Ce qui signifie que $\Gamma \models a_1 \rightarrow a_2$.

(\rightarrow_e) D'après l'hypothèse et les cas précédents, on a $\Gamma \models a_1 \rightarrow a_2$ et $\Gamma \models a_1$. Prenons un modèle de Γ . D'après notre hypothèse, ce modèle est à la fois un modèle de a_1 et un modèle de l'implication $a_1 \rightarrow a_2$. D'après la table de vérité de l'implication, comme celle-ci est vraie et que a_1 est vraie, on a nécessairement a_2 vraie. Donc, $\Gamma \models a_2$.

(\neg_i) D'après l'hypothèse et les cas précédents, on a $\Gamma, a \models \perp$. Soit un modèle de Γ et de a . D'après l'hypothèse, de ce modèle, on ne peut qu'engendrer que la contradiction. Or, c'est un modèle de a . Donc on ne peut pas en déduire a . Par contre, on peut en déduire $\neg a$. Donc de ce modèle de Γ , on peut déduire $\neg a$.

(\neg_e) D'après l'hypothèse et les cas précédents, on a $\Gamma \models \neg a$ et $\Gamma \models a$. Tout modèle de Γ est à la fois un modèle de a et de $\neg a$. Si l'on part du principe du tiers exclu, cela est impossible. L'ensemble des modèles vérifiant ces conditions est vide. On a donc $\Gamma \models \perp$.

(\implies) Cette démonstration est l'objet du théorème de complétude de Gödel[gödel_vollständigkeit_1930]. On l'admet dans ce cours. ■

R On peut donc conclure que, dans le cadre de la déduction naturelle, :

- tout ce qui est prouvable est vrai (correction),
- tout ce qui est vrai est prouvable (complétude).

R Si des stratégies de preuve existent en déduction naturelle pour chercher un arbre de preuve automatiquement, elles ne sont en général pas très simples à exécuter et font souvent appel au retour sur trace, ce qui nuit à leur efficacité. D'autres systèmes ont été développés depuis (calcul des séquents, calcul des constructions) qui dépassent ces limitations.

E Raisonnements utiles en logique classique

a Raisonnement par l'absurde

Le raisonnement par l'absurde s'énonce simplement :

$$\frac{\Gamma, \neg a \vdash \perp}{\Gamma \vdash a} \text{ raa}$$

R En latin, le raisonnement se dit *reduction ad absurdum*, d'où l'acronyme raa.

■ **Exemple 6 — Preuve du principe d'explosion.** Il est possible de prouver le principe d'explosion c'est-à-dire :

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash a} \perp_e$$

La preuve s'appuie sur le raisonnement par l'absurde qui est admis en logique classique. On introduit habilement la négation de a dans les hypothèses du séquent en procédant par affaiblissement que l'on note aff.

$$\frac{\frac{\overline{\Gamma \vdash \perp}}{\Gamma \vdash \perp} \text{ ax}}{\frac{\Gamma, \neg a \vdash \perp}{\Gamma \vdash a}} \text{ aff}$$

b Tiers exclu

Le principe du tiers exclu s'énonce simplement :

$$\frac{}{\Gamma \vdash a \vee \neg a} \text{ te}$$

R Le tiers exclu ainsi formulé, $\Gamma \vdash a \vee \neg a$, signifie que l'une des deux propositions est vraie. Cependant, cela n'implique pas que a et $\neg a$ ne puissent pas être vraies en même temps. Ce qui empêche a et $\neg a$ d'être vraies simultanément, cest le **principe de non-contradiction** :

$$\Gamma \vdash \neg(a \wedge \neg a). \tag{6}$$

Ce principe peut être démontré en déduction naturelle, comme ci-dessous. Cette preuve est valable en logique classique et intuitionniste.

$$\frac{\frac{\frac{a \wedge \neg a \vdash a \wedge \neg a}{a \wedge \neg a \vdash a} \text{ax}}{\frac{a \wedge \neg a \vdash a \wedge \neg a}{a \wedge \neg a \vdash \neg a} \wedge_e} \text{ax}}{\frac{a \wedge \neg a \vdash \perp}{\vdash \neg(a \wedge \neg a)} \neg_i}$$

R On peut démontrer le principe du tiers exclu si on admet le raisonnement par l'absurde et l'élimination de la double négation :

$$\frac{\frac{\frac{\Gamma, a \vee \neg a \vdash \perp}{\Gamma \vdash \neg(a \vee \neg a) \equiv \neg a \wedge a} \text{ax}}{\frac{\Gamma \vdash \neg(a \vee \neg a) \equiv \neg a \wedge a}{\Gamma \vdash a} \wedge_e} \text{raa}}{\frac{\frac{\Gamma, a \vee \neg a \vdash \perp}{\Gamma \vdash \neg(a \vee \neg a) \equiv \neg a \wedge a} \text{ax}}{\frac{\Gamma \vdash \neg a}{\Gamma \vdash a \vee \neg a} \wedge_e} \text{raa}} \vee_i$$

c Élimination de la double négation

La double négation en logique classique peut être éliminée puisqu'on admet le principe du raisonnement par l'absurde.

$$\frac{\Gamma \vdash \neg\neg a}{\Gamma \vdash a} \neg\neg_e$$

On peut le démontrer si l'on admet le raisonnement par l'aburde :

$$\frac{\frac{\frac{\neg\neg a, \neg a \vdash \neg\neg a}{\neg\neg a, \neg a \vdash \perp} \text{ax}}{\neg\neg a, \neg a \vdash a} \neg\neg_e}{\neg\neg a \vdash a} \text{raa}$$

F Exemples de preuves

a Syllogisme hypothétique

Le syllogisme hypothétique s'exprime sous la forme du séquent $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$.

$$\frac{\frac{\frac{p \rightarrow q, q \rightarrow r, p \vdash q \rightarrow r}{p \rightarrow q, q \rightarrow r, p \vdash p \rightarrow q} \text{ax}}{\frac{p \rightarrow q, q \rightarrow r, p \vdash p \rightarrow q}{p \rightarrow q, q \rightarrow r, p \vdash q} \rightarrow_e} \text{ax}}{\frac{p \rightarrow q, q \rightarrow r, p \vdash r}{p \rightarrow q, q \rightarrow r \vdash p \rightarrow r} \rightarrow_i} \rightarrow_e$$

b Modus tollendo tollens

Du latin *en niant, je nie*, cette figure s'exprime sous la forme du séquent $p \rightarrow q, \neg q \vdash \neg p$.

$$\frac{\frac{p \rightarrow q, \neg q, p \vdash p \rightarrow q}{p \rightarrow q, \neg q, p \vdash q} \text{ax} \quad \frac{p \rightarrow q, \neg q, p \vdash p}{p \rightarrow q, \neg q, p \vdash \perp} \text{ax}}{\frac{p \rightarrow q, \neg q, p \vdash \perp}{p \rightarrow q, \neg q \vdash \neg p}} \neg_i$$

G Vers la logique du premier ordre → HORS PROGRAMME

a Syllogismes

■ **Définition 6 — Mnémonique.** Au féminin, une mnémonique est un ensemble des procédés qui facilitent les opérations de mémorisation. Dans le cadre de la logique, il s'agit donc d'astuces pour mémoriser des formules logiques. Dans le cadre de l'informatique, on peut utiliser ce mot au masculin; il désigne alors une instruction en langage d'assemblage (de type chaîne de caractères) correspondant à une instruction du langage machine (de type entier codé en binaire), par exemple : ADD R1, R2.

Au moyen âge, les philosophes et les logiciens ont développé des mnémoniques pour identifier et mémoriser facilement certaines figures de la logique et de la rhétorique. Ils choisissaient des mots dont les voyelles représentaient des affirmations (A) ou des réfutations (E) **universelles**¹, des affirmations(I) ou des réfutations (O) **particulières**². Les syllogismes du moyen-âge et de l'antiquité exprime donc des prédicats de la logique d'ordre 1.

■ **Exemple 7 — barbara.** Le syllogisme barbara est un syllogisme de type AAA. Il représente une figure du type TOUT M EST P, OR TOUT S EST M, DONC TOUT S EST P.

■ **Exemple 8 — celarent.** Le syllogisme celarent est un syllogisme de type EAE. Il représente une figure du type AUCUN M N'EST P, OR TOUT Q EST M, DONC AUCUN Q N'EST P

La logique du premier ordre introduit la notion de prédicat, de fonction, de variable liée ou libre ainsi que deux quantificateurs. Lorsqu'une formule logique F dépend par une certaine variable propositionnelle x , on note $F(x)$.

■ **Définition 7 — Variable liée.** Dans une formule logique du premier ordre, une variable est liée à un quantificateur si le nom par lequel on la désigne ne modifie pas la formule. C'est pourquoi elle est aussi désigné par le terme variable muette.

Par exemple, $\forall x.F(x) \wedge y$ possède la même signification que $\forall t.F(t) \wedge y$. x et t sont des variables liées.

Par contre, $\forall x.F(x) \wedge y$ et $\forall x.F(x) \wedge z$ ne possèdent pas la même signification : l'une est une propriété sur y et l'autre sur z . y et z sont des variables libres.

R Une même variable peut apparaître liée et libre dans une même formule comme c'est le cas pour celle-ci : $(\forall x F(x)) \wedge G(x)$

1. c'est-à-dire avec le quantificateur universel \forall qui naîtra bien plus tard : *Tous les hommes sont mortels*.

2. c'est-à-dire avec le quantificateur existentiel \exists : *Il existe au moins un homme mortel*

■ **Définition 8 — De la liberté dans les formules.** Une variable est libre dans une formule si elle possède au moins une occurrence libre dans cette formule.

Une variable est liée dans une formule si toutes les occurrences de la variable dans la formule sont liées.

b Règles du quantificateur existentiel

Soit une instance d'une formule $F(x)$. Si au moins une des valeurs possibles de x fait que la formule $F(x)$ est vraie, alors on introduit le quantificateur existentiel et on note : $\exists x.F(x)$.

Le quantificateur peut être introduit en déduction naturelle par la règle suivante, t étant une valeur pour laquelle F est satisfaite :

$$\frac{\Gamma \vdash F[x \leftarrow t]}{\Gamma \vdash \exists x.F(x)} \exists_i$$

De même, si la variable x n'est libre dans aucune formule, on peut éliminer le quantificateur existentiel par la règle :

$$\frac{\Gamma \vdash \exists x.F(x) \quad \Gamma, F \vdash \phi \quad x \text{ n'est une variable libre ni de } \Gamma \text{ ni de } \phi}{\Gamma \vdash \phi} \exists_e$$

c Règles du quantificateur universel

Le quantificateur universel traduit l'idée que F peut être déduite indépendamment de x . Il est introduit en déduction naturelle par la règle suivante :

$$\frac{\Gamma \vdash F \quad x \text{ n'est pas une variable libre de } \Gamma}{\Gamma \vdash \forall x.F(x)} \forall_i$$

De même, on peut éliminer le quantificateur universel par la règle en rompant la généralisation :

$$\frac{\Gamma \vdash \forall x.F(x)}{\Gamma \vdash F[x \leftarrow t]} \forall_e$$

■ **Exemple 9 — Preuve simple en logique du premier ordre.** On cherche à montrer que $\forall x.F(x) \vdash \exists x.F(x)$.

$$\frac{\overline{\forall x.F(x) \vdash \forall x.F(x)} \text{ ax}}{\overline{\forall x.F(x) \vdash F[x \leftarrow t]} \exists_i} \forall_e$$

H Correspondance de Curry–Howard

→ HORS PROGRAMME

a Typage et jugements de typage

■ **Définition 9 — Expression bien typée.** Une expression est dite *bien typée* lorsque les fonctions et opérateurs qui la composent sont appliqués à des paramètres de types compatibles avec leur signature.

■ **Exemple 10** En OCaml, l'expression `2 + 3` est bien typée, car l'opérateur `(+)` a pour type `int -> int -> int`. En revanche, `2.0 + 3.0` n'est pas bien typée : l'opérateur adapté est `(+.)`.

■ **Définition 10 — Jugement de typage.** Soit Γ un environnement de typage associant des types aux variables libres. On note

$$\Gamma \vdash e : \tau$$

pour exprimer que, dans le contexte Γ , l'expression e est bien typée et a pour type τ .

■ **Exemple 11** Voici quelques jugements de typage valides en OCaml :

- $\vdash 21 : \text{int}$
- $\vdash \text{true} : \text{bool}$
- $\vdash (+) : \text{int} \rightarrow \text{int} \rightarrow \text{int}$
- $\vdash (+) 2 3 : \text{int}$

Si l'on dispose du contexte :

$$\Gamma = \{f : \text{int} \rightarrow \text{int} \rightarrow \text{int}, a : \text{int}, b : \text{int}\},$$

alors on peut dériver :

$$\Gamma \vdash f\ a\ b : \text{int}.$$

b Règles de typage et analogie logique

Le typage d'un programme est défini de manière inductive à l'aide de règles d'inférence, analogues aux règles de dérivation en logique. On retrouve notamment des règles correspondant à l'axiome, à l'introduction et à l'élimination de l'implication.

$$\begin{array}{c} \frac{}{\Gamma, x : \tau \vdash x : \tau} (\text{ax}) \\[10pt] \frac{\Gamma, x : s \vdash e : t}{\Gamma \vdash \lambda x. e : s \rightarrow t} (\rightarrow \text{I}) \\[10pt] \frac{\Gamma \vdash f : s \rightarrow t \quad \Gamma \vdash e : s}{\Gamma \vdash fe : t} (\rightarrow \text{E}) \end{array}$$

c Correspondance de Curry-Howard

L'observation de ces règles met en évidence une correspondance structurelle entre le typage des programmes et la déduction logique :

Logique intuitionniste	Programmation typée
Formule	Type
Preuve	Programme
Implication $A \Rightarrow B$	Type fonction $A \rightarrow B$
Élimination / introduction	Application / abstraction

Cette correspondance, dite *correspondance de Curry-Howard*, établit un lien profond entre démonstrations et programmes : démontrer une proposition revient à construire un programme du type correspondant.

Elle est au fondement des assistants de preuve tels que COQ ou du langage LEAN, dans lesquels les preuves sont des termes bien typés d'un langage fonctionnel riche.

R La correspondance de Curry–Howard éclaire le caractère fondamentalement mathématique de la notion de programme, sans pour autant identifier totalement logique et programmation : elle s'applique principalement à des logiques constructives et à des langages fortement typés, et doit être comprise comme une correspondance structurelle plutôt que comme une équivalence absolue. Cependant, c'est un argument fort **contre** la brevetabilité du logiciel, car breveter un logiciel, c'est breveter une formule mathématique. Doit-on breveter les formules mathématiques ? Où est-ce un bien commun ?