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.

■ **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 :

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

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

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

Par contre, ce dernier est valide :

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

Ⓡ En mathématiques, on note généralement un séquent sous la forme d'un théorème avec l'im-

plication matérielle :

$$x \in \mathbb{R}, x^2 - 10x + 21 = 0 \Longrightarrow x = 3 \vee x = 7 \tag{5}$$

■ **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 :

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

■ **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 $a \wedge b$.

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

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 :

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash a \wedge b}\,\text{ax}}{\Gamma \vdash b}\wedge_e \quad \dfrac{\dfrac{}{\Gamma \vdash a \wedge b}\,\text{ax}}{\Gamma \vdash a}\wedge_e}{\Gamma \vdash b \wedge a}\wedge_i$$

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 $a \to b$ en se passant de l'hypothèse a.

$$\frac{\Gamma, a \vdash b}{\Gamma \vdash a \to b} \to_i$$

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 :

$$\frac{\Gamma \vdash a \to b \qquad \Gamma \vdash a}{\Gamma \vdash b} \to_e$$

R Dans l'antiquité, cette règle de l'élimination de l'implication $\to_e$ était nommé *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 $\vdash p \to p$.** On donne ci-dessous la preuve que l'implication matérielle est réflexive.

$$\dfrac{\dfrac{}{p \vdash p}\,\text{ax}}{\vdash p \to p}\to_i$$

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

$$\dfrac{\dfrac{\dfrac{}{p \wedge q, p \vdash p \wedge q}\,\text{ax}}{p \wedge q, p \vdash q}\wedge_e}{p \wedge q \vdash p \to q}\to_i$$

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.

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

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

$$\dfrac{\dfrac{}{p \vee q \vdash p \vee q}\,\text{ax} \qquad \dfrac{\dfrac{}{p \vee q, p \vdash p}\,\text{ax}}{p \vee q, p \vdash q \vee p} \vee_i \qquad \dfrac{\dfrac{}{p \vee q, q \vdash q}\,\text{ax}}{p \vee q, q \vdash q \vee p} \vee_i}{p \vee q \vdash q \vee p} \vee_e$$

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.

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

Symétriquement, l'élimination d'une négation conduit à une contradiction.

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

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

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash a} \bot_e$$

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 peut prouver une formule et son contraire. Elle utilise un syllogisme disjonctif pour introduire $\bot$ et conclue par un raisonnement par l'absurde.

Formule	Introduction	Élimination
$\top$	$\dfrac{}{\Gamma \vdash \top}\ \top_i$	
$\bot$		$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash a}\ \bot_e$ (Principe d'explosion)
$a \in \Gamma$ (Axiome)	$\dfrac{}{\Gamma \vdash a}\ \text{ax}$	
Conjonction	$\dfrac{\Gamma \vdash a \qquad \Gamma \vdash b}{\Gamma \vdash a \wedge b}\ \wedge_i$	$\dfrac{\Gamma \vdash a \wedge b}{\Gamma \vdash a}\ \wedge_e$
Disjonction	$\dfrac{\Gamma \vdash a}{\Gamma \vdash a \vee b}\ \vee_i$ et / ou $\dfrac{\Gamma \vdash b}{\Gamma \vdash a \vee b}\ \vee_i$	$\dfrac{\Gamma \vdash a \vee b \qquad \Gamma, a \vdash c \qquad \Gamma, b \vdash c}{\Gamma, a \vee b \vdash c}\ \vee_e$
Implication	$\dfrac{\Gamma, a \vdash b}{\Gamma \vdash a \to b}\ \to_i$	$\dfrac{\Gamma \vdash a \to b \qquad \Gamma \vdash a}{\Gamma \vdash b}\ \to_e$ (Modus ponendo ponens)
Négation	$\dfrac{\Gamma, a \vdash \bot}{\Gamma \vdash \neg a}\ \neg_i$	$\dfrac{\Gamma \vdash \neg a \qquad \Gamma \vdash a}{\Gamma \vdash \bot}\ \neg_e$ (Introduction de la contradiction $\bot_i$)

TABLE 1 – Ensemble des règles de la déduction naturelle

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.

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 :

$$\Gamma \vDash a \Longleftrightarrow \Gamma \vdash a \tag{6}$$

Démonstration. On procède en montrant les implications dans les deux directions.

($\Longleftarrow$) Dans ce sens, la démonstration s'appuie sur la définition des règles d'inférence : pour chaque règle, on montre que si $\Gamma \vdash a$ alors on a $\Gamma \vDash a$.

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

($\bot_e$) Supposons que $\Gamma \vdash \bot$. $\bot$ 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 $\bot$ est l'ensemble vide, alors il n'existe pas de modèle de Γ qui ne soit pas un modèle de $\bot$. Donc, $\Gamma \vDash \bot$.

($a \in \Gamma$) **Axiome** Soit $a \in \Gamma$. Un modèle de Γ est donc un modèle de a. Donc, $\Gamma \vDash a$.

($\wedge_i$) On s'appuie sur le cas précédent et on suppose donc que $\Gamma \vDash a_1$ et $\Gamma \vDash a_2$. Tout modèle de Γ est donc à la fois un modèle de a_1 **et** un modèle de a_2. Ce qui signifie que $\Gamma \vDash a_1 \wedge a_2$.

($\wedge_e$) On s'appuie sur le cas précédent et on suppose donc que $\Gamma \vDash a_1 \wedge a_2$, 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 s'appuie sur le cas précédent (axiome) et on suppose donc 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 $\Gamma \vDash a_1 \vee a_2$.

($\vee_e$) On suppose donc que $\Gamma \vDash a_1 \vee a_2$, Γ, $\Gamma, a_1 \vDash c$ 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 $\Gamma, a_1 \vee a_2 \vDash c$.

($\rightarrow_i$) D'après l'hypothèse et les cas précédents, on a $\Gamma, a_1 \vDash a_2$ et on cherche à montrer que $\Gamma \vDash a_1 \rightarrow a_2$. 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 quod libet.

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 $\Gamma \vDash a_1 \rightarrow a_2$.

($\rightarrow_e$) D'après l'hypothèse et les cas précédents, on a $\Gamma \vDash a_1 \rightarrow a_2$ et $\Gamma \vDash a_1$. 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, $\Gamma \vDash a_2$.

($\neg_i$) D'après l'hypothèse et les cas précédents, on a $\Gamma, a \vDash \bot$. Soit un modèle de Γ et de a. D'après l'hypothèse, de ce modèle, on ne peut qu'engendrer que le faux. 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 $\Gamma \vDash \neg a$ et $\Gamma \vDash a$. Tout modèle de Γ est à la fois un modèle de a et de $\neg a$. Si l'on part du principe du tiers exclus, cela est impossible. L'ensemble des modèles vérifiant ces conditions est vide. On a donc $\Gamma \vDash \bot$.

($\Longrightarrow$) Cette démonstration est l'objet du théorème de complétude de Gödel[**godel_vollstandigkeit_1930**]. On l'admet dans ce cours.

■

E Raisonnements utiles en logique classique

a Raisonnement par l'absurde

Le raisonnement par l'absurde s'énonce simplement :

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

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 :

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash a}\ \bot_e$$

La preuve s'appuie sur le raisonnement par l'absurde. On introduit habilement la négation de c dans les hypothèses du séquent en procédant par affaiblissement [a] que l'on note aff.

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \bot}\ \text{ax}}{\Gamma, \neg a \vdash \bot}\ \text{aff}}{\Gamma \vdash a}\ \text{raa}$$

a. qui peut le plus peut le moins

b Élimination de la double négation

La double négation en logique classique peut être éliminée puisqu'on admet le principe du tiers exclus.

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

■ **Exemple 7 — Preuve du raisonnement par l'absurde par la double négation.** Il est possible de prouver le principe d'explosion en utilisant l'élimination de la double négation :

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \neg c \vdash \bot}\ \text{ax}}{\Gamma \vdash \neg\neg c}\ \neg_i}{\Gamma \vdash c}\ \neg\neg_e$$

c Tiers exclus

Le principe du tiers exclus s'énonce simplement :

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

■ **Exemple 8 — Preuve du raisonnement par l'absurde par le tiers exclus.** Il est possible de prouver le raisonnement par l'absurde en utilisant le tiers exclus, le principe d'explosion (élimination de $\bot$) et l'élimination de la disjonction :

$$\dfrac{\dfrac{}{\Gamma \vdash a \vee \neg a}\ \text{te} \qquad \dfrac{}{\Gamma, a \vdash a}\ \text{ax} \qquad \dfrac{\dfrac{}{\Gamma, \neg a \vdash \bot}\ \text{ax}}{\Gamma, \neg a \vdash a}\ \bot_e}{\Gamma \vdash a}\ \vee_e$$

F Exemples de preuves

a Syllogisme hypothétique

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

$$\dfrac{\dfrac{\dfrac{}{p \to q, q \to r, p \vdash q \to r}\ \text{ax} \qquad \dfrac{\dfrac{}{p \to q, q \to r, p \vdash p \to q}\ \text{ax} \qquad \dfrac{}{p \to q, q \to r, p \vdash p}\ \text{ax}}{p \to q, q \to r, p \vdash q}\ \to_e}{p \to q, q \to r, p \vdash r}\ \to_e}{p \to q, q \to r \vdash p \to r}\ \to_i$$

b Modus tollendo tollens

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

$$\dfrac{\dfrac{\dfrac{}{p \to q, \neg q, p \vdash p \to q}\ \text{ax} \qquad \dfrac{}{p \to q, \neg q, p \vdash p}\ \text{ax}}{p \to q, \neg q, p \vdash q}\ \to_e \qquad \dfrac{}{p \to q, \neg q, p \vdash \neg q}\ \text{ax}}{p \to q, \neg q, p \vdash \bot}\ \neg_e}{p \to q, \neg q \vdash \neg p}\ \neg_i$$

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.

1. c'est-à-dire avec le quantificateur universel $\forall$ qui naîtra bien plus tard : *Tous les hommes sont mortels* ou bien *Aucun homme n'est mortel.*
2. c'est-à-dire avec le quantificateur existentiel $\exists$: *Il existe au moins un homme mortel* ou bien *Aucun homme n'est mortel.*

■ **Exemple 9 — 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 10 — 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 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 : $(\forall x F(x)) \wedge G(x)$

■ **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 :

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

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 :

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

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

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

H Correspondance Curry-Howard

--→ HORS PROGRAMME

■ **Définition 9 — Expression bien typée.** Dans une expression bien typée, les types des fonctions et des opérateurs utilisés coincident avec le type des paramètres des fonctions.

Par exemple, en OCaml, `2 + 3` est bien typée car l'opérateur `+` sait opérer sur deux entiers. Par contre, `2.0 + 3.0` n'est pas bien typée.

■ **Définition 10 — Jugement de typage.** Si, pour un environnement de variables Γ donné, l'expression e est bien typée et a le type τ, alors on note

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

■ **Exemple 12 — Jugements de typage en OCaml.** Voici quelques exemples simples de jugement de typage en OCaml.

- $\vdash$ `21 : int`
- $\vdash$ `true : bool`
- $\vdash$ `(+)2 3 : int`
- $\vdash$ `(+): int -> int -> int`

Si on dispose des jugements :

- $\vdash$ `f : int -> int -> int`
- $\vdash$ `a : int`
- $\vdash$ `b : int`

alors on peut appliquer `a` et `b` à `f` et écrire $\vdash$ `f a b : int`.

De la même manière que pour les formules logiques, un contexte de variables,c'est-à-dire une environnement définissant un typage des variables, doit donc être précisé :

`a : int, b : int, f : int -> int -> int` $\vdash$ `f a b : int`

Ainsi, pourvu que le type défini des expressions utilisées soit respecté, alors on peut déduire le type d'une autre expression.

La définition d'un programme bien typé peut se faire de manière inductive et la dérivation de typage se fait de manière similaire au calcul des séquents. On peut donc vérifier formellement le typage d'un programme.

Les règles de l'axiome, de l'élimination de l'implication ou de l'introduction de l'implication logique possèdent leur correspondant dans la vérification de types.

$$\frac{}{\Gamma \vdash e : \tau}\ \text{ax}$$

$$\dfrac{\dfrac{}{\Gamma, \texttt{a : s} \vdash \texttt{e : t}}\,\text{ax}}{\Gamma \vdash \texttt{fun a -> e : s -> t}}\,\text{fun}_i$$

$$\dfrac{\dfrac{}{\Gamma \vdash \texttt{f : s -> t}}\,\text{ax} \qquad \dfrac{}{\Gamma \vdash \texttt{e : s}}\,\text{ax}}{\Gamma \vdash \texttt{f e : t}}\,\text{fun}_e$$

En observant ces dérivations, on est frappé par le fait qu'on pourrait établir une correspondance entre :

- les types et les formules logiques,
- les programmes et les preuves.

C'est la correspondance de Curry-Howard qui est au cœur des logiciels d'assistant de preuve comme Coq qui permettent de vérifier une démonstration et de démonstration automatique. Les applications sont mathématiques, électroniques (conception des circuits) et informatique (vérification d'assertions relatives à des programmes).

R La correspondance Curry-Howard permet de mettre en lumière le lien étroit entre les formules mathématiques et les types, les preuves et les programmes. 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 ?