Logique 1A

June 8, 2021

1 Introduction

En logique du premier ordre et, en particulier, en théorie de la démonstration, les objets que l'on étudie sont les *formules* et leur *démonstrations*. Les termes et les formules forment la grammaire d'une langue, simplifiée à l'extrême et calculée exactement pour dire ce que l'on veut sans ambiguïté et sans détour inutiles.

Les termes. Ils distinguent les objets dont on veut prouver les propriétés.
Les formules. Elles représentent les propriétés des objets que l'on étudie.
Les démonstrations. Elles permettent d'établir qu'une formule est *vraie*.

1.1 Les formules

1.1.1 Le langage

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

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

- les symboles de *constante*;
- les symboles de *fonction*. A 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 corréspond au nombre d'arguments et on parle de relation *unaire, binaire, n-aire.*

Exemple:

Le langage de l'analyse réelle continent les symboles:

- constantes: $0, 1, ..., e, ..., \pi$
- fonctions: $+, -, |.|, \sin, \ln, ...$
- relations: $=, \geq, \leq, ...$

1.1.2 Les termes

On se donne un ensemble (infini) V de variables. Les variables seront notées: $x, y, z, ...$ (éventuellement indexées: $x_1, ..$)

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

Définition: Soit L un langage et $(T_i)_{i\in\mathbb{N}}$ une famille d'ensembles.

L'ensemble T des termes est $T = \bigcup_{k\in\mathbb{N}} T_k$ où:

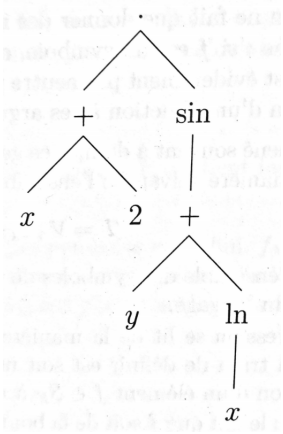
- $T_0 = \{t/t$ est une variable ou un symbole de constante$\}$
- $\forall k \in \mathbb{N}^*, T_{k+1} = T_k \cup \{f(t_1, ..., t_n)/t_i \in T_k$ et f le symbole de fonction d'arité $n\}$

On appellera *hauteur* d'un terme t le plus petit k tel que $t \in T_k$.
Un terme *clos* est un terme qui ne contient pas de variables.

Cette 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 pas neutre puisque son *sens* est l'application d'une fonction à ses arguments.

Exemple:
Sur le langage précédent, $\sin(\ln(x) + \cos(|y + e|))$ est un terme et $\ln(|\cos(e) - \sin(e)|)$ est un terme clos.

Remarque:
Il est souvent commode de voir un terme comme un arbre dont chaque noeud est étiqueté par un symbole de fonction et chaque feuille par une variable ou une constante. Par exemple, le terme $(x + 2).\sin(y + \ln(x))$ est représenté par l'arbre suivant:



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 $P(t_1, ..., t_n)$. On fait une preuve par induction sur la hauteur d'un terme.

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.

Ainsi, avant d'avancer, il faut rappeler ce qu'est l'induction.

2 L'Induction

2.1 Rappel sur les prédicats

Définition: Un prédicat est une fonction dont la valeur est vrai ($\top$) ou faux ($\bot$).

Une famille de prédicats est par exemple l'ensemble des prédicats sur ls entiers, c'est à dire les fonctions de la forme:

$$P : \mathbb{N} \to \{\top, \bot\}$$

Exemple:

Soit $P : \mathbb{N} \to \{\top, \bot\}$ le prédicat défini par:

$$P(n) = \begin{cases} \top & \text{si } \sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \\ \bot & \text{sinon} \end{cases}$$

2.1.1 L'induction mathématique

Dans l'exemple précédent, on peut démontrer que le prédicat $P(n)$ est vrai pour tout $n \geq 0$. Ainsi, les assertions $P(0), P(1), P(2), P(3), P(4), ...$ sont vraies. Mais comment peut-on prouver qu'une infinité d'assertions sont vraies ?

Première stratégie naïve:
1. On commence par prouver que $P(1)$ est vrai.
2. On montre ensuite que $P(1) \Rightarrow P(2)$ est vrai.
3. On utilise le modus ponens pour en déduire que $P(2)$ est vrai.
4. On montre que $P(2) \Rightarrow P(3)$ est vrai.
5. On utilise le modus ponens pour en déduire que $P(3)$ est vrai. ECT...

Le problème de l'infinité d'étapes pour montrer que $P(n)$ est vrai pour tout $n \geq 0$ n'est pas encore reglé, mais il va rapidement l'être puisqu'on remarque bien que les étapes 2 et 4 suivent une même méthode. Si on généralise, on peut les regrouper en une seule étape:
Montrer que $P(n-1) \Rightarrow P(n)$ est vrai.

On a:

$$\sum_{i=0}^{n} 2^i = 2^n + \sum_{i=0}^{n-1} 2^i$$

Si on suppose que $P(n-1)$ est vrai on a:

$$\sum_{i=0}^{n-1} 2^i = 2^n - 1$$

Donc on a $P(n)$ vrai:

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

Or dans l'assertion logique $A \Rightarrow B$, si on suppose que A est vrai et qu'on montre que B est vrai, alors l'implication est forcément vraie d'après le tableau de vérité de $\Rightarrow$ (voir la première et la dernière ligne):

A	B	$A \Rightarrow B$
$\top$	$\top$	$\top$
$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$

Ainsi, on a montré que $P(n-1) \Rightarrow P(n)$.

Pourtant, on a aucune indication sur la véracité ou la fausseté de $P(0), P(1), P(2), ...$ Pour en avoir une, il faut maintenant formaliser **l'induction mathématique**:

Soit $P : \mathbb{N} \to \{\top, \bot\}$ un prédicat et $k \geq 0$ une constante. Si les deux conditions suivantes sont vérifiés:

- $P(k)$ est vrai
- Pour tout $n > k, P(n-1) \Rightarrow P(n)$ est vrai

Alors $P(m)$ est vrai pour tout $m > k$.

Ainsi, en montrant que $P(0)$ est vrai, on peut démontrer que le prédicat $P(n)$ est vrai pour tout $n \geq 0$.

2.2 La définition inductive

Plus généralement, l'induction est un outil formel élégant utilisé pour définir des ensembles et des fonctions ainsi que démontrer des propriétés sur ces objets.

La définition **inductive** d'un ensemble $X \subset E$ consiste en la donnée:

- de certains éléments de X (les éléments de base)
- de règles de construction d'éléments de X à partir d'éléments déjà connus de X (étapes inductives)
- d'une règle implicite

Formellement:

Soit E un ensemble. Une partie X de E a une définition inductive si on a:

- Un sous-ensemble B de E
- Un ensemble K de règles Reg

et si X est le **plus petit ensemble vérifiant les assertions (B) et (I) suivantes**:

- (B): $B \subset E$
- (I) : $\forall Reg \in K$:

$$x_1, x_2, ..., x_{a(Reg)} \in X \Rightarrow Reg(x_1, x_2, ..., x_{a(Reg)}) \in X$$

Ou $a(Reg)$ est l'arité de Reg, c'est à dire le nombre d'arguments d'une règle Reg.

Exemple: définition inductive de $\mathbb{N}$.

On prend comme ensemble B le singleton $\{0\}$ et comme ensemble K, le singleton réduit à l'opération *sucesseur* définit comme la fonction suivante: $\forall n \in \mathbb{N}$, $Reg(n) = n + 1$.

La définition inductive de $\mathbb{N}$ est donc:

$$\begin{cases} 0 \in \mathbb{N} \\ n \in \mathbb{N} \Rightarrow Reg(n) = n + 1 \in \mathbb{N} \end{cases}$$

Pour simplifier on écrira:

$$\frac{}{0}\ (\text{B})$$

$$\frac{n \in \mathbb{N}}{n + 1}\ (\text{I})$$

Exemple: définition inductives des palindromes

Soit A un ensemble (fini). Un mot sur A est une suite finie d'éléments de A. On définit l'opération de concaténation C sur les mots de A comme étant $u.v$ (la suite d'éléments u suivie de la suite d'éléments v). On note ϵ la suite vide.

Soit V l'ensemble des palindromes de A. Donnons une définition inductive de V:

$$\frac{}{\epsilon \in V}\ (\text{B})$$

$$\frac{x \in V}{\forall y \in A,\ y.x.y \in V}\ (\text{I})$$

On peut maintenant définir le **principe d'induction** de façon généralisé (qui englobe donc l'induction mathématique que l'on a définit):

Soit X un ensemble définit inductivement par un ensemble B et un ensemble K. Soit $P : X \to \{\bot, \top\}$ un prédicat. Si on a:

- (B"): $\forall b \in B, P(b)$ (c'est à dire: $P(b)$ est vrai)
- (I"): $\forall Reg \in K, P(x_1), P(x_2), ..., P(x_{a(Reg)}) \Rightarrow P(Reg(x_1, x_2, ..., x_{a(Reg)}))$

Alors $\forall x \in X, P(x)$.

Ainsi, faire une démonstration par induction, c'est vérifier B" et I".

2.3 Exemple sur le langage de Dyck

Définition inductive du langage $D \subset \{(,)\}^*$ de Dyck des parenthésages bien formés:

$$\frac{}{\epsilon \in D}\,(\mathrm{B})$$

$$\frac{x \in D}{(x) \in D}\,(I_1)$$

$$\frac{x \in D \qquad y \in D}{xy \in D}\,(I_2)$$

Exercice: montrer que $(()())(())$ est un élément de D.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \in D}\,(B)}{()}\,(I_1) \quad \dfrac{\dfrac{}{\epsilon \in D}\,(B)}{()}\,(I_1)}{()()}\,(I_2)}{(()())}\,(I_1) \quad \dfrac{\dfrac{\dfrac{}{\epsilon \in D}\,(B)}{()}\,(I_1)}{(())}\,(I_1)}{(()())(())}\,(I_2)$$

Exercice: montrer que tout mot du langage de Dyck a autant de parenthèses ouvrantes que de parenthèses fermantes.

Le prédicat que nous voulons prouver (montrer qu'il est vrai pour tout mot du langage de Dyck) se définit donc de la façon suivante:

$$P(x) = \begin{cases} \top & \text{si le nombre de parenthèses ouvrantes de } x \text{ est égal} \\ & \text{au nombres de ses parenthèses fermantes.} \\ \bot & \text{sinon} \end{cases}$$

Ainsi, notre démonstration par induction doit montrer que:

$$\frac{}{P(\text{"}\epsilon\text{"}) = \top}\ (\text{B"})$$

$$\frac{x \in D,\ P(\text{"}x\text{"}) = \top}{P(\text{"}(x)\text{"}) = \top}\ (I_1'')$$

$$\frac{x \in D,\ P(\text{"}x\text{"}) = \top \qquad y \in D,\ P(\text{"}y\text{"}) = \top}{P(\text{"}xy\text{"}) = \top}\ (I_2'')$$

On pourrait justifier ces propositions par un argument de bon sens, mais il serait plus rigoureux de mathématiser le problème. Ainsi on définit $\sigma_o : D \to \mathbb{N}$ la fonction qui compte les parenthèses ouvertes et $\sigma_f : D \to \mathbb{N}$ qui compte les parenthèses fermées. P équivaut alors à $P(x) = (\sigma_o(x) - \sigma_f(x) = 0)$.

B" est vrai puisque $\sigma_o(\epsilon) = \sigma_f(\epsilon) = 0$.

I_1'' est vrai puisque si $P(\text{"}x\text{"}) = \top$ alors $\sigma_o(x) - \sigma_f(x) = 0$ et donc:
$\sigma_o(x) - \sigma_f(x) + \sigma_o(\text{"}()\text{"}) - \sigma_f(\text{"}()\text{"}) = \sigma_o(x) - \sigma_f(x) + 1 - 1 = 0$.

I_2'' est vrai puisque si $P(\text{"}x\text{"}) = \top$ et $P(\text{"}y\text{"}) = \top$ alors $\sigma_o(x) - \sigma_f(x) = \sigma_o(y) - \sigma_f(y) = 0$ donc $\sigma_o(x) - \sigma_f(x) + \sigma_o(y) - \sigma_f(y) = 0$.

Ainsi, $\forall x \in D, P(x)$, donc tout mot du langage de Dyck a autant de parenthèses ouvrantes que de parenthèses fermantes

3 Les Formules

Pour revenir aux termes, on peut maintenant définir leur *taille* inductivement:

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

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

Exemple: Si on veut la taille du terme $\sin(\ln(x))$, on peut construire une sorte d'arbre qui l'exprime:

$$\cfrac{\cfrac{\cfrac{1 + 1 + 0 = 2}{1 + 1 + \tau(x)}\ (B)}{1 + \tau(\ln(x))}\ (I)}{\tau(\sin(\ln(x)))}\ (I)$$

3.1 Types de formules

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

Définition: Soit L un langage. Les formules *atomiques* de L sont les formules de la forme : $R(t_1, ..., t_n)$ où R est un symbole de relation n-aire de L et $t_1, ..., t_n$ sont des termes de L. On note $Atom$ l'ensemble des formules atomiques. Si on note S_R l'ensemble des symboles de relation, on peut écrire:

$$Atom = S_R(T, ..., T)$$

Exemple:

Dans les langages appropriés:
La formule $= (x, y)$, que l'on écrit $x = y$ est atomique.
La formule $\geq (x, y)$, que l'on écrit $x \geq y$ est atomique.
La formule $\wedge(A, B)$, que l'on écrit $A \wedge B$ est atomique.
La formule $\forall x, \sin(x) = 3$ n'est pas atomique mais $\sin(x) = 3$ l'est.
ect...

En outre, on peut aussi définir les *sous-formules* d'une formule:

Définition: Une *sous-formule* d'une formule F est l'un de ses "composants", ie. une formule à partir de laquelle F est construite. Formellement on définit l'ensemble $SF(F)$ des sous-formules de F par:

- Si F est atomique, $SF(F) = \{F\}$
- Si $F = F_1 \oplus F_2$ avec $\oplus \in \{\wedge, \vee, \rightarrow\}$ alors $SF(F) = \{F\} \cup SF(F_1) \cup SF(F_2)$
- Si $F = \neg F_1$ ou Qx, F_1 où $Q \in \{\exists, \forall\}$ alors $SF(F) = \{F\} \cup SF(F_1)$

Tout comme les termes, il existe aussi une définition de la *taille* d'une formule:

Définition: 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.
- $\tau(F_1 \oplus F_2) = 1 + \tau(F_1) + \tau(F_2)$
- $\tau(\neg F_1) = \tau(Qx, F_1) = 1 + \tau(F_1)$

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. $\phi(F_1)$.

3.2 Variables libres et variables liées

La présence des quantificateurs $\exists$ et $\forall$ pose un problème concernant les noms donnés aux variables:

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 que l'on appelle *muettes*). 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.

Définition: Soit F une formule. L'ensemble $VL(F)$ des *variables libres* de F et l'ensemble $VM(F)$ des *variables muettes* de F sont définis par récurrence sur $\tau(F)$:

- Si $F = R(t_1, ..., t_n)$ est atomique: $VL(F)$ est l'ensemble des variables apparaissant dans les t_i et $VM(F) = \emptyset$

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

Exemple: $\forall y(x.y = y.x)$ 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.

On remarque bien que on ne peut toujours pas renommer y et x dans la formule $\forall y(x.y = y.x)$ et obtenir $\forall x(x.x = x.x)$: la variable x serait ***capturée***. La définition précédente est informelle et incomplète car on ne peut pas renommer les occurences liées sans précaution: il faut éviter de *capturer* des occurences libres. On trouvera donc une définition formelle plus tard.

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* $x_1, ..., x_n$ ie. si y est libre dans F, alors y est l'un des x_i mais les x_i n'apparaissent pas nécessairement dans F.

Définition:
1. Une formule *close* est une formule sans variables libres.

2. Soit F une formule dont les variables libres sont $x_1, ..., x_n$. La *clôture* (universelle) de F est la forme 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.

3.3 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.

Définition: 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 (ie. muettes) de F qui apparaissent libres dans t. Le renommage à donc pour but d'éviter la capture de variables.

3.4 Définition du calcul propositionnel

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

Définition: L'ensemble C_P des formules du *calcul propositionnel* est défini par la grammaire (ou V_P est l'ensemble des relations d'arité 0) :

$$C_P = V_P | \bot | \neg C_P | C_P \wedge C_P | C_P \vee C_P | C_P \rightarrow C_P$$

L'expression de la grammaire ci-dessus se lit de la manière suivante: un élement de l'ensemble C_P que l'on est en train de définir est soit un élement de V_P, soit $\bot$, soit un élement de $\neg C_P$, soit un élement de $C_P \wedge C_P$, soit un élement de $C_P \vee C_P$, soit un élement de $C_P \rightarrow C_P$.

Remarque: les relations d'arité 0 sont souvent appelés *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:

La formules ci-dessous est un exemple de formule du calcul propositionnel ayant comme variables propositionnelles $X, YetZ$:

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

4 Quelques applications

4.1 Exercice 1.1 du livre (page 56)

(1): $g(a, f(b))$

Ici, a et b sont des symboles de constante, g étant un symbole de fonction binaire et f un symbole de fonction unaire, (1) est bien un terme; il est en plus de taille 2 puisque on compte

deux symbole de fonction. Ce n'est pas une formule puisqu'il n'y a ni symbole de relation, ni connecteurs ou quantificateurs.

(2): $g(R(a), b)$

Ici, on a un "problème": L'ensemble des termes est défini inductivement par $T = \bigcup_{k\in\mathbb{N}} T_k$, où T_0 est l'ensemble des symboles de variable ou de constante. Ce dernier s'unit, inductivement, avec les symboles de fonction inclus dans les $(T_k)_{k\in\mathbb{N}^*}$. **Il n'y a donc jamais de symboles de relations dans un terme** ce qui fait que (2) n'en ai pas un. Étant donné qu'il n'est pas une formule non plus, on ne peut rien dire sur lui.

(3): $(\forall x, g(x, x) = b) \wedge (\exists x, f(x) = b)$

Pour les mêmes raisons que celles cités précédemment, (3) ne peut pas être un terme. Sans ce prononcer sur (3), on peut extraire ces deux formules atomiques:

- $g(x, x) = b$
- $f(x) = b$

Attention: $(\forall x, g(x, x) = b) \wedge (\exists x, f(x) = b)$ n'est pas une formule atomique elle-même: **pour que ce soit une formule atomique, il faut que la relation soit entre deux termes**.

Ainsi,

- $F_1 : \forall x, g(x, x) = b$
- $F_2 : \exists x, f(x) = b$

sont des formules.

Donc, (3) revient à $F_1 \oplus F_2$ avec $\oplus$ qui représente le connecteur $\wedge$. F **est donc une formule qui à comme sous-formules** F_1 **et** F_2.

Quant à se taille:

$$\begin{aligned}\tau(F) &= \tau(F_1 \wedge F_2)\\ &= 1 + \tau(F_1) + \tau(F_2)\\ &= 1 + (1 + \tau(g(x, x) = b) + 1 + \tau(f(x) = b))\\ &= 1 + 1 + 0 + 1 + 0 = 3\end{aligned}$$

4.2 Exercice 3 du TD1

Exercice 3

On définit inductivement l'ensemble $X \subset \{a, b\}^*$ de la façon suivante : $\epsilon \in X$; si $u \in X$ alors $a.u.b \in X$.
Montrer que $X = \{a^n b^n \mid n \in \mathbb{N}\}$.
Par convention $a^0 = \epsilon$

Montrons que $X \subset \{a^n b^n | n \in \mathbb{N}\}$

Nous allons faire une preuve par induction. Il faut donc montrer:

- B": $\epsilon \in \{a^n b^n | n \in \mathbb{N}\}$
- I": si $u \in X$ et $u \in \{a^n b^n | n \in \mathbb{N}\}$ alors $a.u.b \in \{a^n b^n | n \in \mathbb{N}\}$

B" est vrai puisque par convention: $a^0 b^0 = \epsilon\epsilon = \epsilon$.

I" est vrai puisque si $u \in \{a^n b^n | n \in \mathbb{N}\}$ alors $\exists n_0 \in \mathbb{N}$ tel que $u = a^{n_0} b^{n_0}$ et donc: $a.u.b = a^{n_0+1} b^{n_0+1} \in \{a^n b^n | n \in \mathbb{N}\}$.

Montrons que $\{a^n b^n | n \in \mathbb{N}\} \subset X$

Nous allons faire une preuve par récurrence. Soit $n \in \mathbb{N}$, on pose comme hypothèse de récurrence:

$$H_n : a^n b^n \in X$$

Initialisation: $a^0 b^0 = \epsilon \in X$.

Hérédité: Soit $k \in \mathbb{N}$ tel que (H_k). On a $a^k b^k \in X$ donc, par application de la règle I, on a $a^{k+1} b^{k+1} \in X$. (H_{k+1}) est prouvé.

Conclusion: Notre récurrence démontre que $\forall n \in \mathbb{N}, a^n b^n \in X$, donc $\{a^n b^n | n \in \mathbb{N}\} \subset X$.

Finalement, on a donc montré que:

$$X = \{a^n b^n | n \in \mathbb{N}\}$$

4.3 Exercice 6 du TD1

Exercice 6

On considère l'ensemble $X \subset \mathbb{N}^2$ définit inductivement par l'élément de base $(0, 0)$ et par les règles d'inférence suivantes :

$$\frac{(a, b)}{(a+1, b+1)} I_1 \qquad \frac{(a, b)}{(a+1, b)} I_2$$

1) Donner quelques éléments de X.
2) Pour chaque élément suivant dire s'il appartient à X ou non. Si oui , donnez l'arbre de construction, sinon justifiez.
   a) $(3, 3)$
   b) $(2, 5)$
   c) $(4, 2)$
3) Donner une définition non inductive des éléments de X.

2) Élément $(3, 3)$

$$\frac{(0, 0)}{(3, 3)} (I_1) * 3$$

2) Élément $(2,5)$

Pour démontrer que $(2,5)$ n'est pas dans X, on va prouver par induction, la proposition suivante (rigoureusement ce devrait être un prédicat mais on peut se permettre quelques approximations):

$$(P) : \forall (x,y) \in X, x \geq y$$

B" : $0 \geq 0$ est immédiatement vérifié.

I_1": si $a \geq b$ alors $a+1 \geq b+1$: I_1'' est vérifié.

I_2'': si $a \geq b$ alors $a+1 \geq b$: I_2'' est vérifié.

Par notre preuve par induction, si $(x,y) \in X$, alors $x \geq y$ donc $(2,5) \notin X$.

2) Élément $(4,2)$

$$\dfrac{\dfrac{(0,0)}{(2,0)}\,(I_2)*2}{(4,2)}\,(I_1)*2$$

3)

Montrons que $X = \{(x,y) \in \mathbb{N}^2 / x \leq y\}$. On a déjà montré que $X \subset \{(x,y) \in \mathbb{N}^2 / x \geq y\}$, il ne reste donc que l'autre inclusion à démontrer.

Montrons que $\{(x,y) \in \mathbb{N}^2 / x \geq y\} \subset X$

Soit $(a,b) \in \{(x,y) \in \mathbb{N}^2 / x \geq y\}$. Quelque soit les éléments a, b on peut leur donner un arbre de construction:

$$\dfrac{\dfrac{(0,0)}{(b,b)}\,(I_1)*b}{(a,b)}\,(I_2)*(a-b)$$

Donc tous les couples (a,b) tels que $a \geq b$ sont des théorèmes; ce qui montre que $\{(x,y) \in \mathbb{N}^2 / x \geq y\} \subset X$.

4.4 Exercice 1.5 du livre (page 57)

Sujet: Montrer que le nombre de sous-formules d'une formule F est inférieur ou égal à $2^{\tau(F)+1}-1$.

Re-définissons tout d'abord **inductivement**, l'ensemble des formules que nous noterons F_a.

$$\dfrac{F \text{ est atomique}}{F \in F_a}\,(B)$$

$$\frac{F \in F_a}{QF \in F_a}\,(I_1)$$

$$\frac{F \in F_a}{\neg F \in F_a}\,(I_2)$$

$$\frac{F_1 \in F_a \qquad F_2 \in F_a}{F_1 \oplus F_2 \in F_a}\,(I_3)$$

Faisons une preuve par induction de la proposition en jeu:

Notation: $ns(F) = Card(SF(F))$

B": si F est atomique on sait que $\tau(F) = 1$, donc $2^{\tau(F)+1} - 1 = 1$. On sait aussi que $SF(F) = \{F\}$ donc $ns(F) = 1$ ce qui donne bien $ns(F) \leq 2^{\tau(F)+1} - 1$.

I_1'': **si** $ns(F) \leq 2^{\tau(F)+1} - 1$ **alors** $ns(QF) \leq 2^{\tau(QF)+1} - 1$. Preuve:

Comme on sait que $\tau(QF) = 1 + \tau(F)$, on a:

$$ns(QF) = ns(F) + 1 \leq 2^{\tau(F)+1} \leq 2^{\tau(F)+2} - 1 = 2^{\tau(QF)+1} - 1$$

I_2'': Analogue à la preuve de I_1'' (on peut le voir en "remplacant" Q par $\neg$)

I_3'': Supposons que $ns(F_1) \leq 2^{\tau(F_1)+1} - 1$ et que $ns(F_2) \leq 2^{\tau(F_2)+1} - 1$.

On sait que $SF(F_1 \oplus F_2) = \{F_1 \oplus F_2\} \cup SF(F_1) \cup SF(F_2)$ donc:

$$\begin{aligned} ns(F_1 \oplus F_2) = 1 + ns(F_1) + ns(F_2) &\leq 1 + 2^{\tau(F_1)+1} - 1 + 2^{\tau(F_2)+1} - 1 \\ &= 2^{\tau(F_1 \oplus F_2) - \tau(F_2)} + 2^{\tau(F_1 \oplus F_2) - \tau(F_1)} - 1 \\ &\leq 2 * 2^{\tau(F_1 \oplus F_2) - \min(\tau(F_1), \tau(F_2))} - 1 \\ &\leq 2^{\tau(F_1 \oplus F_2)+1} - 1 \end{aligned}$$

car $\tau(F_1 \oplus F_2) = 1 + \tau(F_1) + \tau(F_2)$ et $\min(\tau(F_1), \tau(F_2)) \geq 0$.

5 Les Démonstrations

5.1 Les séquents

On démontre, en général, des formules en utilisant un ensemble d'hypothèses, et cet ensemble peut varier en 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: 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.

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

Remarques:

- "$\Gamma \vdash F$" représente à la fois le séquent et la phrase "$\Gamma \vdash F$ est prouvable". Il n'y aura, en général, pas d'ambiguïté.
- On écrira $\Gamma \nvdash F$ pour dire "$\Gamma \vdash F$ n'est pas prouvable".
- Il existe des systèmes de démonstration qui n'utilisent pas ce principe hypothèses/conclusions (par exemple le système d'Hilbert).

5.2 Rappel: démonstration formelle d'une formule

Rappelons comment construire un arbre de preuve pour démontrer une formule classique: $\neg A \leftrightarrow (A \to \bot)$.

Avant de commencer, définissions une règle d'introduction de $\leftrightarrow$:

$$\frac{\vdash A \to B \qquad \vdash B \to A}{\vdash A \leftrightarrow B} \leftrightarrow_i$$

Cette règle se déduit naturellement de l'introduction de la conjonction et du fait que: $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$

A partir de là, on peut écrire l'arbre de $\neg A \leftrightarrow (A \to \bot)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \to \bot, A \vdash A \to \bot \qquad A \to \bot, A \vdash A}{A \to \bot, A \vdash \bot} \to_e}{A \to \bot \vdash \neg A} \neg_i}{\vdash (A \to \bot) \to \neg A} \to_i \qquad \dfrac{\dfrac{\dfrac{\neg A, A \vdash \neg A \qquad \neg A, A \vdash A}{\neg A, A \vdash \bot} \neg_e}{\vdash \neg A \to (A \to \bot)} \to_i *2}{\vdash \neg A \leftrightarrow (A \to \bot)} \leftrightarrow_i$$

6 Complétude de la logique du premier ordre

6.1 Interprétations

Dans cette section, on définit la *vérité* d'une formule, ce qui nécessite la notion d'**interprétation** d'un langage.

Définition: Soit L un langage de la logique du premier ordre. On appelle *interprétation* du langage L, l'ensemble M des données suivantes:

- Un ensemble non vide $|M|$, appelé *domaine* ou *ensemble de base* de M.
- pour chaque symbole de constante c, un élément c_M de $|M|$.
- pour chaque symbole de fonction n-aire f, une fonction f_M, partout définie, de $|M|^n$ dans $|M|$
- pour chaque symbole de relation n-aire R (autre que $=$) un sous ensemble R_M de $|M|^n$.

Remarques:

- On utilise quelquefois les mots *structure* ou *modèle*, avec le même sens, à la place du mot *interprétation*
- Donner une interpétation consiste donc à dire dans quel ensemble on travaille et ce que représente les symboles utilisés. L'hypothèse $|M|$ non vide, outre le fait qu'une interprétation vide n'est pas très intéressante, est nécessaire pour avoir le **théorème de complétude**. En effet, si on veut qu'une formule démontrable soit vraie dans tout interpétation, il faut, à cause de la règle $\exists_i$, que $|M|$ soit non vide.
- Par convention: $|M|^0 = \emptyset$
- **Pour simplifier les notations, on confondra souvent une interprétation avec son ensemble de base**. C'est une pratique courante en mathématique: quand on parle de l'espace vectoriel $\mathbb{R}^n$ on ne précise pas ce qu'est l'addition et le produit par un scalaire. On dira donc souvent "soit M une interprétation" sans préciser d'avantage.

Définition: Soit M une interprétation du langage L.

1. Un *environnement* est une fonction de l'ensemble des variables dans $|M|$, l'ensemble de base de M.

2. Si e est un environnement et $a \in |M|$, on note $e[x := a]$ l'environnement e' tel que $e'(x) = a$ et $e'(y) = y$ pour y différent de x.

Définition: Soit M une interprétation du langage L. La *valeur* du terme t dans l'environnement e (on la note $\text{Val}_M(t, e)$) est définie, par récurrence sur la taille de t, de la manière suivante:

- $\text{Val}_M(c, e) = c_M$
- $\text{Val}_M(x, e) = e(x)$ si x est une variable
- $\text{Val}_M(f(t_1, ..., t_n), e) = f_M(\text{Val}_M(t_1, e), ..., \text{Val}_M(t_n, e))$

Exemple: Soit $L = \{0, 1, +, *, =\}$ le langage de l'arithmétique.

On peut prendre comme interprétation l'ensemble $\mathbb{N}$ en interprétant les symboles dans leur sens habituel.

On peut également prendre $\mathbb{Z}$ (ou $\mathbb{R}, \mathbb{Q}, \mathbb{C}$...) en interprétant les symboles avec leur sens habituel.

Mais on peut aussi prendre l'interpétation M définie par $|M| = \mathbb{N}$, $0_M = 1_M = 18$ et $n +_M m = 3.n + 5.m$. Ce qui donne par exemple: $0_M +_M 3 = 3 * 18 + 5 * 3 = 69$.

Lemme: $\text{Val}_M(t, e)$ ne dépend que de la valeur de e sur les variables de t

Notation: Pour simplifier les notations, on évitera de noter l'indice M ou le paramètre e quand le contexte est suffisamment clair. On pourra donc noter $\text{Val}(t, e)$ ou encore $\text{Val}(t)$.

Définition: Soit M une interprétation du langage L. La valeur d'une formule F de L dans l'environnement e est un élement de l'ensemble $\{0, 1\}$ noté $\text{Val}_M(F, e)$ (que l'on notera ici $\text{Val}(F)$) et défini, par récurrence sur la taille de F, de la manière suivante:

- $\text{Val}(\bot) = 0$
- $\text{Val}(R(t_1, ..., t_n)) = 1$ ssi $(\text{Val}(t_1), ..., \text{Val}(t_n)) \in R_M$
- $\text{Val}(\neg F_1) = 1$ ssi $\text{Val}(F_1) = 0$
- $\text{Val}(F_1 \wedge F_2) = 1$ ssi $\text{Val}(F_1) = 1$ et $\text{Val}(F_2) = 1$
- $\text{Val}(F_1 \vee F_2) = 1$ ssi $\text{Val}(F_1) = 1$ ou $\text{Val}(F_2) = 1$
- $\text{Val}(\forall x F_1) = 1$ ssi pour tout $a \in |M|$, $\text{Val}(F_1, e[x := a]) = 1$
- $\text{Val}(\exists x F_1) = 1$ ssi il existe $a \in |M|$, $\text{Val}(F_1, e[x := a]) = 1$

Remarques et notations

1. Les définitions précédentes peuvent paraître triviales, mais cette trivialité apparente vient seulement du fait que les connecteurs et les quantificateurs ont pour nous leur sens "courant". On notera qu'on a, par exemple, écrit $\wedge$ pour designer le connecteur qui apparaît dans une formule, mais on a utilisé le mot "ou" quand on définit son sens: c'est alors, intuitivement, le "ou" de notre méta-langage.

2. De même que formules et termes sont des objets de nature différentes, leur valeur est aussi de valeur différente: la valeur d'un terme est un élément du domaine $|M|$ alors que la valeur d'une formule est un booléen (0 représentant le faux et 1 le vrai).

3. Dans le cas d'un symbole R de relation d'arité 0 (une variable propositionnelle) on conviendra que $\mathrm{Val}_M(R) = 1$ si $R_M = \{\emptyset\}$ et $Val_M(R) = 0$ si $R_M = \emptyset$. En effet, $|M|^0 = \{\emptyset\}$ et on peut identifier ses sous ensembles à $\{0, 1\}$

4. On notera souvent $M, e \Vdash F$ (ou même $M \Vdash F$ si l'environnement est clair) au lieu de $\mathrm{Val}_M(F, e) = 1$. De la même façon, on notera $M, e \nVdash F$ (ou $M \nVdash F$) à la place de $\mathrm{Val}_M(F, e) = 0$. Ainsi, au lieu de dire "la valeur de F dans l'environnement e est 1 (resp. 0)" on dira plutôt "M satisfait (resp. ne satisfait pas) F dans l'environnement e". Si $M \Vdash F$, on dit aussi que M est un *modèle* de F.

5. Il résulte immédiatement de la définition que, pour tout environnement e, on a: $M, e \Vdash F$ ou $M, e \nVdash F$. Il est également facile de voir que $M, e \Vdash \neg F$ ssi $M, e \nVdash F$. Donc on ne peut pas avoir simultanément $M, e \Vdash F$ et $M, e \Vdash \neg F$.

Lemme: $\mathrm{Val}_M(F, e)$ ne dépend que de la valeur de e sur les variables **libres** de F.

Corollaire: Si F est une formule close, alors la valeur de F est indépendante de l'environnement. On note alors $M \Vdash F$.

Exemple:

Soit $L = \{e, f\}$ le langage où e est un symbole de constante et f est un symbole de fonction binaire. On définit les interprétations N et Z par :

- $|N| = \mathbb{N}$, $e_N = 0$ et $f_N(a, b) = a + b$
- $|Z| = \mathbb{Z}$, $e_Z = 0$ et $f_Z(a, b) = a + b$

Considérons les formules:

$$F : \forall x, y, z\{f(x, f(y, z)) = f(f(x, y), z)$$

$$G : \forall x\{f(e, x) = x \wedge f(x, e) = x\}$$

$$H : \forall x \exists y\{f(x, y) = e \wedge f(y, x) = e\}$$

On sait que $M \Vdash F$ ssi pour tout élément a, b, c de $|M|$ on a $f_M(a, f_M(b, c)) = f_M(f_M(a, b), c)$. F exprime donc le fait que f_M est une opération associative dans $|M|$. G exprime le fait que e_M est un élement neutre pour f_M dans $|M|$ et H exprime le fait que tout élement de $|M|$ a un inverse pour f_M.

Donc: $N \Vdash F$, $N \Vdash G$, $N \nVdash H$ et $Z \Vdash F$, $Z \Vdash G$, $Z \Vdash H$

Définition d'un théorème

On dit qu'une formule F est un *théorème* ssi pour toute interprétation M et tout environnement e, M satisfait F dans l'environnement e ($M, e \Vdash F$).

En d'autres mots, les théorèmes sont les formules *toujours* vraies, c'est à dire quelque soit le sens qu'on donne aux objets, fonctions et relations ($\approx$ quelque soit l'interprétation, aussi appelé **modèle** dans ce contexte, sachant qu'on néglige un peu l'environnement).

Exemple:

Soit $L = \{e, *\}$ où e est une constante et $*$ un symbole de fonction binaire. Soit A la formule suivante:

$$\forall x, y, z\{(x * y) * z = x * (y * z) \wedge e * x = x \wedge x * e = x \wedge \exists x'(x' * x = e \wedge x * x' = e)\}$$

Il est clair que chaque modèle de A est un groupe. Maintenant, soit la formule B:

$$\forall e'\{\forall x[e' * x = x \wedge x * e' = x] \rightarrow e = e'\}$$

B est la formule qui exprime l'unicité de l'élement neutre. **On a donc la formule $A \rightarrow B$ qui est un théorème.**

Définition: Soient L et L' deux langages.

- On dit que L' *enrichit* L (ou que L est une *réstriction* de L') si $L \subset L'$.
- On suppose $L \subset L'$. Soient M une interprétation de L et M' une interprétation de L'. On dit que M' est un *enrichissement* de M (ou que M est une restriction de M') ssi $|M| = |M'|$ et chaque symbole de constante, de fonction ou de relation de L a la même interprétation dans M et dans M'.

Propriété: Soient $L \subset L'$ deux langages et M (resp. M') une interprétation de L (resp. L'). Si M' est un enrichissement de M et e un environnement, alors:

- Si t est un terme de L, $\mathrm{Val}_M(t, e) = \mathrm{Val}_{M'}(t, e)$.
- Si F est une formule de L, alors $M, e \Vdash F$ ssi $M', e \Vdash F$.

Ainsi, la vérité d'une formule dans une interpétation ne dépend que de la restriction de cette interpétation au langage de la formule. **On peut donc parler d'une interpétation M d'une formule sans dire de quel langage elle est une interpétation.**

6.2 Introduction à la théorie des modèles

Dans un cours de mathématiques, on étudie les propriétés de certaines structures: groupes, annaux, corps, espaces vectoriels, ect.

Certaines notions sont omniprésentes, par exemple celles de morphisme et d'isomorphisme. Pour d'autres, le mot utilisé dépend du type de la structure: sous-groupe, sous-espace vectoriel ect.

La théorie des modèles est l'étude générale des structures mathématiques. Dans cette séction on étudie quelques-unes de ces propritétés. La proposition qui suit la première définition, sera utile pour simplifier certaines preuves:

Définition: Deux formules F et G sont *équivalentes* ssi la formule $F \leftrightarrow G$ est un théorème; ie. ssi pour toute interpétation M et tout environnement e: $M, e \Vdash F$, soit $\mathrm{Val}_M(F, e) = 1$.

Proposition: Toute formule est équivalente à une formule n'utilisant que les connecteurs $\neg$, $\vee$ et le quantificateur $\exists$.

Preuve: Cela résulte des équivalences suivantes pour les formules quelconques A et B:

- $A \wedge B$ avec $\neg(\neg A \vee \neg B)$.
- $A \to B$ avec $\neg A \vee B$.
- $\forall x A$ avec $\neg\exists x \neg A$.

Pour finir la preuve il faut donc prouver ces équivalences suivant la définition précédente de l'équivalence (cf. livre page 76)

Définition d'un morphisme et d'un isomorphisme

Soient M et N deux interpétations d'un langage L.

- Un L-morphisme de M dans N est une fonction $\phi : |M| \to |N|$ tele que:
  - Pour chaque symbole de constante c on a: $\phi(c_M) = c_N$
  - Pour chaque symbole de fonction n-aire f et pour $a_1, ...a_n \in |M|$ on a: $\phi(f_M(a_1, ..., a_n)) = f_N(\phi(a_1), ..., \phi(a_n))$.
  - Pour chaque symbole de relation n-aire de R (autre que $=$) et pour $a_1, ...a_n \in |M|$ on a: $(a_1, ..., a_n) \in R_M$ ssi $(\phi(a_1), ..., \phi(a_n)) \in R_N$.
- Un L-isomorphisme est un L-morphisme bijectif.
- M et N sont L-isomorphes s'il existe un L-isomorphisme de M dans N.

Remarque et exemple

- La notion de morphisme dépend du langage. Soit $L = \{0, +, -, \times\}$ et $L' = \{1\} \cup L$. Soient $\mathbb{Z}/3\mathbb{Z}$ et $\mathbb{Z}/12\mathbb{Z}$ les interprétations usuelles. La fonction $\phi : n \to 4n$ de $\mathbb{Z}/3\mathbb{Z}$ dans $\mathbb{Z}/12\mathbb{Z}$ est un L-morphisme puisque $\phi(0_L) = 0'_L = 0_L$, $\phi(0_M +_M 0_M) = \phi(0_M) +_N \phi(0_M) = 0_M +_M +0_M = 0_M = 4 * 0_M$ ect. Par contre, ϕ n'est pas L'-morphisme puisque $\phi(1_M) = 4 \neq 1_N$.

En guise d'exemple, on peut vérifier que si $L = \{c, f, S\}$ et M et N sont définies par:

- $|M| = \mathbb{R}, c_M = 0, f_M(a, b) = a + b$ et $S_M = \{(a, b)/a \leq b\}$
- $|N| =]0, \infty[, c_N = 1, f_N(a, b) = ab$ et $S_N = \{(a, b)/a \leq b\}$

Alors la fonction $\phi : x \to e^x$ est un isomorphisme de M dans N.
En effet:

- $\phi(c_M) = c_N$ est vérifié puisque $e^0 = 1$.
- $\phi(f_M(a, b)) = f_N(\phi(a), \phi(b))$ est vérifié puisque $e^{a+b} = e^a e^b$
- $a, b \in S_M$ ssi $\phi(a), \phi(b) \in S_N$ est aussi vérifié puisque $a \leq b$ ssi $e^a \leq e^b$ par croissance de la fonction exponentielle.

Proposition:

- La composée de deux morphismes est un morphisme.
- La composée de deux isomorphismes est un isomorphisme.
- L'inverse d'un isomorphisme est un isomorphisme.

Notation: Si ϕ est un isomorphisme de M dans N et e un environement dans M, on notera $\phi(e)$ l'environnement e' dans N défini par: $e'(x) = \phi(e(x))$ pour toute variable x.

Lemme: Soient M et N des interprétations d'un langage L et ϕ un morphisme de M dans N. Pour tout terme t et tout environnement e dans M on a:

$$\phi(\mathrm{Val}_M(t, e)) = \mathrm{Val}_N(t, \phi(e)).$$

Preuve:

- Si t est une constante, cela résulte de la définition d'un morphisme.
- Si t est une variable x: $\phi(\mathrm{Val}_M(t, e)) = \phi(e(x)) = \phi(e)(x) = \mathrm{Val}_N(t, \phi(e))$.
- Si $t = f(t_1, ..., t_n)$, On a: $\mathrm{Val}_M(t, e) = f_M(\mathrm{Val}_M(t_1, e), ..., \mathrm{Val}_M(t_n, e))$; on veut donc montrer que: $\phi(f_M(\mathrm{Val}_M(t_1, e), ..., \mathrm{Val}_M(t_n, e))) = f_N(\mathrm{Val}_N(t_1, \phi(e)), ..., \mathrm{Val}_N(t_n, \phi(e)))$ Ce qui par définition du morphisme, revient à: $f_M(\phi(\mathrm{Val}_M(t_1, e)), ..., \phi(\mathrm{Val}_M(t_n, e))) = f_N(\mathrm{Val}_N(t_1, \phi(e)), ..., \mathrm{Val}_N(t_n, \phi(e)))$ En appliquant les résultats précédents à $\phi(\mathrm{Val}_M(t_i, e))$, pour chaque $i \in [1, n]$ on obtient bien la formule précédente.

Lemme: Soient M et N des interprétations d'un langage L et ϕ un morphisme **injectif** de M dans N. Soit e un environnement dans M et F une formule atomique. Alors $M, e \Vdash F$ ssi $N, \phi(e) \Vdash F$.

Preuve: On rappel qu'une formule atomique est une formule de la forme $R(t_1, ..., t_n)$ ou R est un symbole de relation n-aire de L et $t_1, ..., t_n$ sont des termes de L.

On peut donc dire que F est de la forme $F = R(t_1, ..., t_n)$, avec R différent de $=$.

Etant donné que ϕ est un morphisme on sait que pour $t_1, ..., t_n \in |M|$, $(t_1, ..., t_n) \in R_M$ ssi $(\phi(t_1), ..., \phi(t_n)) \in R_N$.

Or, $M, e \Vdash F$ ssi $(\mathrm{Val}_M(t_1, e), ..., \mathrm{Val}_M(t_n, e)) \in R_M$, ssi $(\mathrm{Val}_N(t_1, \phi(e)), ..., \mathrm{Val}_N(t_n, \phi(e))) \in R_N$ ssi $N, \phi(e) \Vdash F$ par application de la définition des morphismes (sur les rélations) rappelée ci-dessus et du lemme précédent (qui donne $(\phi(\mathrm{Val}_M(t_1, e)), ..., \phi(\mathrm{Val}_M(t_n, e))) \in R_M$ ssi $(\mathrm{Val}_N(t_1, \phi(e)), ..., \mathrm{Val}_N(t_n, \phi(e))) \in R_N$).

Il reste encore deux cas particuliers à traiter: pour $F = \bot$, le résultat est évident. Enfin, dans le cas où F est une équation $t_1 = t_2$, on applique le lemme précédent pour dire que $\phi(\mathrm{Val}_M(t_1, e)) = \phi(\mathrm{Val}_M(t_2, e))$ ssi $\mathrm{Val}_N(t_1, \phi(e)) = \mathrm{Val}_N(t_2, \phi(e))$; c'est à dire: $\phi(e(t_1)) = \phi(e(t_2))$ ssi $\phi(e(t_1)) = \phi(e(t_2))$. JSP

Théorème 1.

Soient M et N des interpétations d'un langage L, ϕ un isomorphisme de M dans N et F une formule. Soit e un environnement dans M. Alors $M, e \Vdash$ ssi $N, \phi(e) \Vdash F$.

Corollaire:
Deux interpétations isomorphes satisfont les mêmes formules closes.

Ce corollaire est souvent utilisé pour montrer que deux interpétations (de même cardinal) ne sont pas isomorphes. C'est ainsi, par exemple qu'on montre que les groupes $\mathbb{Z}/4\mathbb{Z}$ et $\mathbb{Z}/2\mathbb{Z}\times\mathbb{Z}/2\mathbb{Z}$ ne sont pas isomorphes. En effet, le premier satisfait la formules $\exists x\{x * x \neq e\}$ contrairement au second.

Le corollaire donne une condition nécessaire pour que deux interpétations soient isomorphes. Mais il faut noter que cette condition n'est pas suffisante.

Définition: Soient L un langage, M et N deux interpétations de L. On dit que M est une *extension* de N (ou N est une *sous-interpétation* de M) ssi les conditions suivantes sont satisfaites:

- $|N| \in |M|$
- Pour tout symbole de constante c de L on a: $c_N = c_M$.
- Pour tout symbole de fonction n-aire f de L on a: $f_N = f_M \mid |N|^n$. (fonction f_M dont l'ensemble de départ est $|N|^n$?) On remarquera que, puisque f_N est une fonction de $|N|^n$ dans $|N|$, cela implique que $f_M(|N|^n)$ est inclus dans $|N|$.
- Pour tout symbole de relation n-aire R de L on a: $R_N = R_M \cap |N|^n$.

Propriété: Soient M et N deux interprétations d'un langage L. M est isomorphe à une sous-interpétation M' de N ssi il existe un morphisme injectif de M dans N.

Remarque: Ce résultat est beaucoup utilisé. Par exemple quand on construit axiomatiquement l'ensemble $\mathbb{Z}$ à partir de $\mathbb{N}$: on définit $\mathbb{Z}$ comme le quotient de $\mathbb{N} \times \mathbb{N}$ par une relation d'équivalence. On définit ensuite une fonction de $\mathbb{N}$ dans $\mathbb{Z}$ et on vérifie que c'est un morphisme injectif. On convient alors de dire que $\mathbb{N}$ est un sous-ensemble de $\mathbb{Z}$.

6.3 Théorème de complétude

Définition: Une *théorie* est un ensemble (fini ou infini) de formules closes. Les éléments d'une théorie sont souvent appelés les *axiomes* de cette théorie.

Définition: Soit T une théorie.

- Une interpétation M *satisfait* T (on dit aussi que M est un *modèle* de T et on note $M \Vdash T$) si M satisfait toutes les formules de T.
- T est *contradictoire* ssi il n'existe pas de modèle de T.
- Une formule close A est *valide* dans T (on le note $T \Vdash A$) ssi $M \Vdash A$ pour tout modèle M de T.

Définition: Soit T une théorie.

- Soit A une formule. On note $T \vdash A$ s'il existe un sous-ensemble *fini* T' de T tel que $T' \vdash A$.
- On dit que T est *consistante* ssi $T \nvdash \bot$.
- On dit que T est *complète* ssi T est consistante et pour toute formule close F: $T \vdash F$ ou $T \vdash \neg F$

Propriété: Soit T une théorie complète.

- Soient A et B des formules closes. $T \vdash A \vee B$ ssi $T \vdash A$ ou $T \vdash B$.
- Soit A une formule close. $T \vdash \neg A$ ssi $T \nvdash A$.

Théorème de complétude

Soient T une théorie et F une formule close.

$$T \vdash F \text{ ssi } T \Vdash F.$$

Remarques:

Il faut noter que la pluspart des théories utilisées en mathématique ne sont pas complètes. Par exemple, on peut montrer que ni la théorie G_1 des groupes ni celle G_2 des groupes commutatifs, ni celle des anneaux A n'est complète.

Soit U la formule: $\forall x, y, \{xy = yx\}$. On a $G_1 \nvdash U$ et $G_1 \nvdash \neg U$ puisqu'on sait bien qu'on peut aussi bien avoir des groupes commutatifs comme des groupes qui ne le sont pas dans G_1. Le théorème de complétude nous donne donc $G_1 \nVdash U$, donc G_1 n'est pas complet.

Sur le théorème de complétude: Ce théorème montre l'équivalence entre la notion de prouvabilité (le côté syntaxique) et la notion de vérité (le côté sémantique). $T \vdash F$ corréspond à la prouvabilité (Le séquent $T \vdash F$ est *prouvable* s'il peut être obtenu par une application finie de régles issues des axiomes.) La notion de vérité équivalente à cette prouvabilité, signifie que la théorie T satisfait F quelque soit le sens qu'on donne aux objets, fonctions et relations.

Théorème sur la consistance

Une théorie T est consistante ssi elle est non contradictoire.

Corollaire: Soit T une théorie non contradictoire. Si tous les modèles de T sont isomorphes, alors T est complète.

Théorème de compacité

Soit T une théorie. T est contradictoire ssi il existe un sous-ensemble fini de T qui est contradictoire. Autrement dit: T est satisfiable ssi tout sous-ensemble fini de T est satisfiable.

6.4 Formes Canoniques

On montre ici que toute formule peut être mise, à équivalence près, sous certaines formes canoniques.

Proposition: Soient A et B des formules. Si x n'est pas libre dans B, les formules $\forall x A \vee B \leftrightarrow \forall x(A \vee B)$ et $\exists x A \vee B \leftrightarrow \exists x(A \vee B)$ sont des théorèmes.

Corollaire: Soient A et B des formules telles que les variables $x_1, ..., x_n$ (resp. $y_1, ..., y_m$ ne sont pas libres dans B (resp. A). Pour tout choix de quantificateurs $Q_1, ..., Q_n; P_1, ..., P_m$, la formule ci-dessous est un théorème:

$$Q_1 x_1 ... Q_n x_n A \vee P_1 y_1 ... P_m y_m B \leftrightarrow Q_1 x_1 ... Q_n x_n P_1 y_1 ... P_m y_m (A \vee B)$$

6.4.1 Formules conjonctives et disjonctives

Définition:

- On dit que F est sous formes *conjonctive* si F est une conjonction de disjonctions de formules atomiques ou de négations de formules atomiques. On écrira une formule conjonctive sous la forme $\wedge_i \vee_j F_{i,j}$: cela signifie donc que les formules $F_{i,j}$ sont des formules atomiques ou des négations de formules atimiques.
- On dit que F est sous formes *disjonctive* si F est une disjonction de conjonctions de formules atomiques ou de négations de formules atomiques. De même on écrira une formule disjonctive sous la forme: $\vee_i \wedge_j F_{i,j}$.

Exemple: $(A \wedge \neg B) \vee C \vee (D \wedge \neg A)$ est sous forme disjonctive. $(D \vee A \vee C) \wedge (D \vee \neg B)$ est sous formes conjonctive.

Théorème

Toute formule sans quantificateurs est équivalente à une formule sous forme conjonctive et à une formule sous forme disjonctive.

6.4.2 Formules prénexes

Définition: Une formule F est sous forme *prénexe* si elle est de la forme $Q_1 x_1 ... Q_n x_n G$ où $Q_i = \forall$ ou $\exists$ et G est sans quantificateurs. On dira alors que $Q_1 x_1 ... Q_n x_n$ est le préfixe de F. Le préfixe peut être vide, ie. une formule sans quantificateur est sous forme prénexe.

Théorème

Toute formule est équivalente à une formule sous forme prénexe.

Définition: On dit qu'une formule F est sous forme *prénexe conjonctive* (resp. *pénexe disjonctive*) ssi $F = Q_1 x_1 ... Q_n x_n G$ ou G est une fomule sans quantificateur et sous forme conjonctive (resp. disjonctive)

Proposition: Toute formule est équivalente à une formule sous forme prénexe conjonctive et à une formule sous forme prénexe disjonctive.

Définition: Une formule est *universelle* (resp. *existentielle*) si elle est prénexe et son préfixe ne contient que des $\forall$ (resp. $\exists$)

Proposition: Soit N une sous-interpétation de M et e un environnement de N.

- Si t est un terme, alors $Val_N(t, e) = Val_M(t, e)$.
- Si F est une formule sans quantificateur, alors $M, e \Vdash F$ ssi $N, e \Vdash F$.
- Si F est universelle et $M, e \Vdash F$ alors $N, e \Vdash F$.
- Si F est existentielle et $N, e \Vdash F$ alors $M, e \Vdash F$.

6.5 Skolémisation

On donne, dans cette section, les résultats mathématiques qui sont à la base de plusieurs algorithmes de démonstration automatique.

Définition: Soit $F = Q_1 x_1 ... Q_n x_n G$ une formule prénexe ($Q_i = \forall$ ou $\exists$ et G est sans quantificateur). Soient $i_1 < ... < i_m$ les indices tels que $Q_i = \exists$.

- On associe à L le langage $L_S(F) = L \cap \{f_1, ..., f_m\}$ où $f_1, .., f_m$ sont des nouveaux symboles de fonction (appelés *fonctions de Skolem associées à* F). L'arité de f_r est $i_r - r$ c'est à dire le nombre de $\forall$ situés à gauche de Q_{i_r} dans le préfixe de F.
- Pour $1 \leq r \leq m$, on désigne par t_r le terme de $L_S(F)$ obtenu en appliqant f_r aux $i_r - r$ variables quantifiées universellement à gauche de Q_{i_r} dans le préfixe de F.
- Une formule F_S (appelée *forme de Skolem* de F) est obtenue à partir de F en enlevant les $\exists$ du préfixe de F et en remplaçant dans la formule G chaque occurence de la variable x_{i_r} par le terme t_r.

Exemple: Soit $L = \{f, R\}$ où f est un symbole de fonction unaire et R est un symbole de relation binaire. Soit $F : \exists x_1 \forall y_1 \forall y_2 \exists x_2 \{R(x_1, f(y_1)) \wedge R(f(y_2), x_1) \rightarrow R(x_1, x_2)\}$

Ici on a $i_1 = 1$ et $i_2 = 4$. $L_S(F) = L \cap \{e, g\}$. Ici, e et g (les fonctions de Skolem associées à F) ont pour arité réspective $1 - 1 = 0$ et $4 - 2 = 2$. e est donc un terme constant (fonction d'arrité 0). g est le symbole de fonction qui s'appliquera à y_1 et y_2 dans la forme de Skolem de F. Ainsi, la formule F_S est donc:

$$F_S : \forall y_1 \forall y_2 \{R(e, f(y_1)) \wedge R(f(y_2), e) \rightarrow R(e, g(y_1, y_2))\}$$

Lemme: Soit F une formule prénexe de L. Alors la formule $F_S \to F$ du langage $L_S(F)$ est un théorème et, par conséquent, $\vdash F_S \to F$.

Lemme: Soient L un langage, F une formule prénexe et M une interprétation de L tels que $M \Vdash F$. Il existe un enrichissement de M en une interprétation N de $L_S(F)$ tel que $N \Vdash F_S$.

Corollaire: Soit F une formule close prénexe et F_S la forme de Skolem de F. F admet un modèle ssi F_S admet un modèle.

C'est le théorème ci-dessous qui est la base théorique de la pluspart des algorithmes de démonstration automatique:

Théorème

Une formule close prénexe F est démontrable ssi la forme de Skolem de $\neg F$ est contradictoire.

Exemple: Soit $F : \exists x \forall y \{R(x) \to R(y)\}$. On a donc $\neg F : \forall x \exists y \{R(x) \wedge \neg R(y)\}$. On a $i_1 = 2$; donc on remplace y par $f(x)$ pour avoir la forme de Skolem de $\neg F$:

$$\neg F_S : \forall x \{R(x) \wedge \neg R(f(x))\}$$

Soit M une interprétation et $a \in |M|$. Si $M \Vdash \neg F_S$ alors $M \Vdash R(f(a))$ et $M \Vdash \neg R(f(a))$ (pas sûr). On a donc une contradiction: F est démontrable, d'après le théorème précédent.

7 Quelques applications sur la théorie des modèles

7.1 Exercice 1

Enoncé: Soit $L = \{0, S, +, \times\}$ où 0 est une constante et S (resp. $+$, $\times$) est un symbole de fonction unaire (resp. binaire). Soit $\mathbb{N}$ l'interpétation standard de L.

Donner la formule, écrite sur L qui exprime: **p est strictement plus petit que q**.

$$F_1 : \exists x, x \neq 0 \wedge p + x = q$$

Donner la formule, écrite sur L qui exprime: **p divise q**

$$F_2 : \exists k, k \neq 0 \wedge p \times k = q$$

Donner la formule, écrite sur L qui exprime: **p est premier**

$$F_3 : \neg(\exists a \exists b, a \times b = p \wedge \neg(a = p \vee b = p))$$

Donc:

$$F_3 : \forall a \forall b, a \times b \neq p \vee a = p \vee b = p$$

7.2 Exercice 2

Exercice 2.2 Soit R un symbole de relation binaire. Pour chacune des formules et des interprétations ci-dessous, dire si la formule est satisfaite ou non dans l'interprétation :

(1) $\forall x, y, z\ \{\neg R(x,x) \wedge [R(x,y) \rightarrow \neg R(y,x)] \wedge [R(x,y) \wedge R(y,z) \rightarrow R(x,z)]\}$
(2) $\exists x \forall y\ R(x,y)$
(3) $\exists x \forall y\ R(y,x)$
(4) $\forall x \exists y\ \{R(x,y) \wedge \forall z\ [R(x,z) \rightarrow (z = y \vee R(y,z))]\}$
(5) $\forall x, y\ \{R(x,y) \rightarrow \exists z\ [R(x,z) \wedge R(z,y)]\}$

(a) $\mathbb{N}$ où R est interprété par $<$.
(b) $\mathbb{Q}$ où R est interprété par $<$.
(c) $P(\mathbb{N})$ où R est interprété par $\subsetneq$.

7.2.1 Cas (1), modèle (a)

Avec le modèle $|M| = \mathbb{N}$, où R est interpété par $<$ on a:
$\forall x, y, z\{\neg(x < x) \wedge (x < y \rightarrow y < x) \wedge (x < y \wedge y < z \rightarrow x < z)\}$

Simplifiable en:

$$\forall x, y, z\{(x < y \rightarrow y < x) \wedge (x < y \wedge y < z \rightarrow x < z)\}$$

Comme il est évident que la première condition dans le $\wedge$ est fausse; cette formule n'est pas satisfiable selon ce modèle.

7.3 Exercice 3

Le but de cet exercice est de montrer qu'en logique du premier ordre on peut se passer des symboles de fonctions: il suffit, intuitivement, de remplacer une fonction par son graphe.

Soient L un langage, f un symbole de fonction d'arité n et R un symbole de relation d'arité $n+1$. On suppose que ni f ni R ne sont dans L. Soient $L_1 = L \cap \{f\}$ et $L_2 = L \cap \{R\}$. Montrer qu'on peut transformer toute formule F écrite sur L_1 en une formule F' écrite sur L_2 telle que pour toute interprétation M de L_1, il existe une interprétation M' de L_2 qui coincide avec M sur L et telle que, pour tout environnement e: $M, e \Vdash F$ ssi $M', e \Vdash F'$.

Pour faire cette transformation, on veut remplacer les formules atomiques par des formules équivalentes qui utilisent des relations.

Exemple: Soit $S(f(u, f(v, w)), h(f(r, s)))$. Cette formule est équivalente à:

$$\exists x_1, x_2, x_3\{R(v, w, x_1) \wedge R(r, s, x_2) \wedge R(u, x_1, x_3) \wedge S(x_3, h(x_2))\}.$$

Preuve: Soit une fonction, définit par récurrence sur la taille, qui à chaque terme t de la formule F écrite sur L_1, associe un terme t', un ensemble $Var(t)$ de variables et un ensemble $A(t)$ de formules atomiques sur L_2. Cette fonction se construit comme suit:

- Si t est une variable ou une constante: $t' = t$ et $Var(t) = A(t) = \emptyset$
- Si $t = g(u_1, ..., u_k)$ où g est une fonction quelconque autre que f, alors: $t' = g(u'_1, ..., u'_k), Var(t) = \bigcup_{1 \leq i \leq k} Var(u_i)$ et $A(t) = \bigcup_{1 \leq i \leq k} A(u_i)$
- Si $t = f(u_1, ..., u_n)$, alors $t' = y$ où y est une nouvelle variable. $Var(t) = \{y\} \cup \bigcup_{1 \leq i \leq n} Var(u_i)$ et $A(t) = \{R(u'_1, ..., u'_n)\} \cup \bigcup_{1 \leq i \leq n} A(u_i)$

Soit $S(t_1, ..., t_k)$ une formule atomique sur L_1. On peut donc transformer cette formule en une formule écrite sur L_2 en la renplaçant par $\exists y_1, ..., y_p\{S(t'_1, ..., t'_k) \wedge \{F/F \in \bigcup_{1 \leq i \leq k} A(t_i)\}$ où $\{y_1, ..., y_p\} = \bigcup_{1 \leq i \leq k} Var(t_i)$.

On prouve le résultat par récurrence sur le nombre d'occurences de f dans la formule atomique.