Le calcul des prédicats

Sophie Pinchinat
IRISA, Université de Rennes 1
sophie.pinchinat@irisa.fr

François Schwarzentruber
ENS Rennes
francois.schwarzentruber@irisa.fr

Pierre Le Barbenchon
ENS Rennes
pierre.le-barbenchon@ens-rennes.fr

Contents

Chapter 1

Introduction

1.1 Lien avec le calcul propositionnel

Le calcul des prédicats hérite des propriétés établies pour celui des propositions.

- Le langage est défini inductivement.
- Les énoncés de cette logique sont *interprétés*, comme ceux du calcul propositionnel. Une *interprétation* en calcul des prédicats joue le rôle d'une valuation dans le calcul propositionnel. Étant donnée une interprétation, on peut évaluer les formules du calcul des prédicats et le résultat de cette évaluation est soit "vrai" soit "faux".

Remarque 1 (Différence algorithmique importante) *On peut montrer qu'il n'existe pas d'algorithme permettant de répondre à la question de la validité d'une formule du calcul des prédicat. On dit que* le calcul des prédicat est indécidable.

1.2 Notions préliminaires

Rappels sur les fonctions Étant donnés un ensemble E et un entier $k \in \mathbb{N}$, une fonction *d'arité* k (ou *k-aire*) sur E est une fonction de E^k dans E.

Exemple 1
1. *Soit la fonction d'arité* 1 *succ* : $\mathbb{N} \to \mathbb{N}$ *définie par* $succ(n) = n + 1$ *et la fonction* 0 *d'arité zéro (pas d'argument, donc une constante).*
2. *Soit la fonction binaire (d'arité* 2*)* $+_3 : (\mathbb{Z}/3\mathbb{Z})^2 \to \mathbb{Z}/3\mathbb{Z}$ *qui fait la somme de deux éléments modulo* 3*.*

Remarque 2 *Une fonction d'arité* 0 *sur E est une constante $c \in E$.*

($E^k = E \times E \times \ldots E$, k fois)

Rappels sur les relations Étant donnés un ensemble E et un entier $k \in \mathbb{N}$, une relation *d'arité k* (ou *k-aire*) sur E est un sous-ensemble de E^k.

Exemple 2
1. $E = \{1, 2, 3\}$ *et* $\mathcal{R}$ *est la relation* binaire *(d'arité* 2*) définie par* $\mathcal{R} = \{(1, 1), (2, 2), (3, 3)\} \subseteq E^2$*.*
2. $E = \mathbb{N}$ *et* $\mathcal{S}$ *est la relation binaire* $\mathcal{S} = \{(n, n + 1) \mid n \in \mathbb{N}\} \subseteq E^2$*.*

3. $E = \{1, 2, 3\}$ *et* $\mathcal{R}$ *est la relation unaire (d'arité* 1*) définie par* $\mathcal{R} = \{1, 2\}$*. Notez que l'on a bien* $\mathcal{R} \subseteq E$

Remarque 3 *Si* $\mathcal{R}$ *est une relation d'arité* n*, on note* $\mathcal{R}(a_1, ..., a_n)$ *pour* $(a_1, ..., a_n) \in \mathcal{R}$*.*

Exercice 1 *L'ensemble* E^0 *est un ensemble à UN élément. Par conséquent, les seules relations d'arité* 0 *sur* E*, c-à-d. les sous-ensembles de* E^0 *sont soit* $\emptyset$ *soit* E^0 *lui-même.*

La syntaxe du calcul des prédicat repose sur les objets suivants.

1. Syntaxe des *termes*, qui s'interprètent comme des objets

2. Syntaxe des formules *atomiques*, qui sont les énoncés de base.

3. Syntaxe des formules plus complexes.

Chapter 2

Termes

Les termes ont pour vocation de dénoter des éléments d'un domaine.

2.1 Syntaxe des termes

On se fixe un ensemble infini de variables $X = \{x, y, x_1, x_2, \ldots\}$, et un ensemble de $\mathcal{F} = \{c, f, g, \ldots\}$ de *symboles de fonctions* avec leur *arité* $ar : \mathcal{F} \to \mathbb{N}$. La fonction ar indique le nombre d'arguments de chaque symbole. On convient de noter $\mathcal{F}_n$ les éléments de $\mathcal{F}$ d'arité n.

Un symbole de fonction d'arité nulle est appelé une *constante*.

Exemple 3

- $\mathcal{F} = \{0, s, +, \times\}$ *avec* $ar(0) =$, $ar(s) = 1$, *et* $ar(+) = ar(\times) = 2$;
- $\mathcal{F} = \{\emptyset, \cap, \cup\}$, *avec* $ar(\emptyset) = 0$, *et* $ar(\cap) = ar(\cup) = 2$;
- $\mathcal{F} = \{\epsilon, a, b, \bullet\}$, *avec* $ar(\epsilon) = ar(a) = ar(b) = 0$, *et* $ar(\bullet) = 2$.

On convient de noter $\mathcal{F} = \{c(0), f(1), \ldots\}$ pour indiquer directement l'arité des symboles de fonctions, c-à-d. que $c, f \in \mathcal{F}$, et que $\mathrm{ar}(c) = 0$, $\mathrm{ar}(f) = 1$.

Un terme est une expression syntaxique formée à partir de X et de symboles de fonction d'arité 0 en utilisant les symboles de $\mathcal{F}$ de sorte qu'un symbole f soit appliqué à un nombre de termes égal à $\mathrm{ar}(f)$. Plus formellement,

Définition 1 (Ensemble des termes sur $\mathcal{F}$ et X) *Étant donnés une signature $\mathcal{F}$ et un ensemble X (de variables), on note $\mathcal{T}(\mathcal{F}, X)$ l'ensemble des termes sur $\mathcal{F}$ et X, il est défini par induction.*

- $x \in \mathcal{T}(\mathcal{F}, X)$ *pour toute variable* $x \in X$
- $c \in \mathcal{T}(\mathcal{F}, X)$ *pour tout symbole d'arité* 0 *(*constante*)*
- *si* $ar(f) = n$ *et* $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, X)$, *alors* $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, X)$.

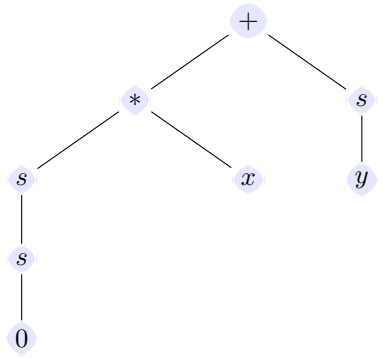
Un terme est clos *s'il ne contient aucune variable. On note $\mathcal{T}(\mathcal{F})$ l'ensemble des termes clos.*

Exemple 4 *Une signature pour les entiers naturels peut être donnée par :*

- *la constante* 0,
- *le symbole s d'arité* 1 *(qui représente la fonction "successeur"),*
- *les symboles $+$ et $\times$ d'arité* 2.

C'est à dire la signature $\mathcal{F} = \{0(0), s(1), +(2), *(2)\}$.

Soient $x, y \in X$. *L'expression* $+(*(s(s(0)), x), s(y))$ *est un terme, que l'on peut représenter par l'arbre suivant :*



Le terme $+(*(s(s(0)), x), s(y))$ *n'est pas clos car il contient des variables, à savoir* x *et* y.

Parfois, nous noterons ABUSIVEMENT $(s(s(0)) * x) + s(y))$

Pour pouvoir donner un signification à un terme, il faut interpréter les symboles de fonction comme des vraies fonctions, donc en particulier se donner un domaine (notion de $\mathcal{F}$-algèbre), et une valeur des variables dans le terme sur ce domaine (notion d'affectation).

2.2 $\mathcal{F}$-algèbre, ou algèbre universelle

Le concept d'algèbre universelle offre un cadre mathématique pour étudier les propriétés de toutes le structures algébriques, telles que les monoïdes, les groupes, les anneaux, les corps, les treillis, etc. Une $\mathcal{F}$-algèbre est une structure mathématique dans laquelle on pourra interpréter les termes de $\mathcal{T}(\mathcal{F}, X)$.

Définition 2 ($\mathcal{F}$-algèbre) *Une* $\mathcal{F}$-algèbre *est une structure* $\mathcal{A} = (D_\mathcal{A}, \{f_\mathcal{A}\}_{f \in \mathcal{F}})$ *où* $D_\mathcal{A}$ *est un ensemble non vide, appelé* domaine*, et pour chaque symbole de fonction* $f \in \mathcal{F}_n$, $f_\mathcal{A} : D_\mathcal{A}^n \to D_\mathcal{A}$.

Notez bien que f est un symbole (appelé symbole de fonction). Le symbole f peut apparaître dans un fichier texte, c'est un caractère ou plusieurs caractères. Au contraire, $f_\mathcal{A}$ est une fonction qui à un tuple de n éléments du domaine $D_\mathcal{A}$ associe un élément du domaine $D_\mathcal{A}$.

Exemple 5

- *Soit* $\mathcal{F} = \{0(0), s(1), +(2)\}$. *Alors* $(\mathbb{N}, succ, +)$ *est une* $\mathcal{F}$*-algèbre et* $(\mathbb{Q}^+, 1, \div 2, \div)$ *est aussi une* $\mathcal{F}$*-algèbre. où* $\mathbb{Q}^+$ *est l' ensemble des rationnels strictement positifs.*
- $\mathcal{T}(\mathcal{F})$ *et* $\mathcal{T}(\mathcal{F}, X)$ *sont des* $\mathcal{F}$*-algèbres dans lesquels les symboles de fonctions jouent le rôle de collage des termes.*

  Plus formellement, pour $f \in \mathcal{F}_n$, *on a ;*

$$f_{\mathcal{T}(\mathcal{F})} : \begin{array}{lcl} \mathcal{T}(\mathcal{F})^n & \to & \mathcal{T}(\mathcal{F}) \\ (t_1, \ldots, t_n) & \mapsto & f(t_1, \ldots, t_n) \end{array}$$

  et de même pour $\mathcal{T}(\mathcal{F}, X)$.
- *Soit* $\mathcal{F} = \{0(0), 1(0), +(2), \times(2)\}$. *Alors on peut proposer des* $\mathcal{F}$*-algèbres tels des anneaux.*

Exercice 2 *Proposer une signature adaptée aux autres structures algébriques des mathématiques.*

Les $\mathcal{F}$-algèbres peuvent être reliées entre elles par des applications particulières.

Définition 3 *Si $\mathcal{A}$ et $\mathcal{B}$ sont deux $\mathcal{F}$-algèbre, un* homomorphisme *(ou simplement* morphisme*) est une application $h : \mathcal{A} \to \mathcal{B}$ telle que, pour tout symbole $f \in \mathcal{F}_n$ et pour tous éléments $a_1, \ldots, a_n$, on a $h(f_{\mathcal{A}}(a_1, \ldots, a_n)) = (f_{\mathcal{B}}(h(a_1), \ldots, h(a_n))$.*

La $\mathcal{F}$-algèbre joue un rôle particulier dans le sens suivant :

Théorème 1 *Soit $\mathcal{A}$ une $\mathcal{F}$-algèbre, et soit $\iota : X \to \mathcal{T}(\mathcal{F}, X)$ l'injection canonique.*

Pour toute affection λ de X dans $\mathcal{A}$, il existe un homomorphime $\hat{\lambda} : \mathcal{T}(\mathcal{F}, X) \to \mathcal{A}$ qui étend λ. Autrement dit tel que

$$\text{pour tout } x \in X, \hat{\lambda}(x) = \lambda(x).$$

Définition 4 (Affectation) *Une* $\mathcal{A}$-affectation *(de X) est une application $\lambda : X \to D_{\mathcal{A}}$.*

Soit $X = \{x_1, \ldots, x_k\}$, et $a_1, \ldots, a_k$ des éléments d'une $\mathcal{F}$-algèbre $\mathcal{A}$. On notera $\{x \mapsto 1a_1, \ldots, x \mapsto ka_k\}$ la $\mathcal{A}$-affectation λ telle que $\lambda(x_i) = a_i$.

Pour le cas particulier où la $\mathcal{F}$-algèbre est $\mathcal{T}(\mathcal{F}, X)$, les $\mathcal{T}(\mathcal{F}, X)$-affectations seront notées σ (au lieu de λ). On appelle *domaine* de la $\mathcal{T}(\mathcal{F}, X)$-affectation σ l'ensemble de variables $\{x \in X \mid \sigma(x) \neq x\}$; il est noté $dom(\sigma)$.

Si $t_1, \ldots, t_k$ sont des termes de $\mathcal{T}(\mathcal{F}, X)$, on note $\{x \mapsto 1t_1, \ldots, x \mapsto kt_k\}$ l'affection σ telle que $\sigma(x_i) = t_i$ pour tout i, et $\sigma(y) = y$ pour tout $y \in X \setminus \{x_1, \ldots, x_k\}$.

2.3 Sémantique des termes

Soient $\mathcal{F}$ un ensemble de symboles de fonctions, X un ensemble de variables.

La valeur d'un terme $t \in \mathcal{T}(\mathcal{F}, X)$ dans la $\mathcal{F}$-algèbre $\mathcal{A}$ pour l'affectation λ, notée $[\![t]\!]_{\mathcal{A},\lambda}$, se définit par induction sur t.

Définition 5 (Valeur d'un terme) *Soit une $\mathcal{F}$-algèbre $\mathcal{A} = (D, \{f\}_{f \in \mathcal{F}})$, et $\lambda \in Affect_{\mathcal{A}}$ une affectation des variables de X dans $D_{\mathcal{A}}$.*

Pour $t \in \mathcal{T}(\mathcal{F}, X)$, on définit $[\![t]\!]_{\mathcal{A},\lambda} \in D_{\mathcal{A}}$ par :

1. *$[\![x]\!]_{\mathcal{A},\lambda} = \lambda(x)$, la valeur de x pour λ.*

2. *$[\![f(t_1, t_2, \ldots, t_n)]\!]_{\mathcal{A},\lambda} = f^{\mathcal{A}}([\![t_1]\!]_{\mathcal{A},\lambda}, [\![t_2]\!]_{\mathcal{A},\lambda}, \ldots, [\![t_n]\!]_{\mathcal{A},\lambda})$, c-à-d. l'image par $f^{\mathcal{A}}$ du tuple d'éléments $([\![t_1]\!]_{\mathcal{A},\lambda}, [\![t_2]\!]_{\mathcal{A},\lambda}, \ldots, [\![t_n]\!]_{\mathcal{A},\lambda})$ de $D_{\mathcal{A}}$.*

Remarque 4 *Le cas d'une constante c est un cas particulier du Point 2. ci-dessus appliqué à un symbole de fonction d'arité 0, puisque cela donne :*

$$[\![c]\!]_{\mathcal{A},\lambda} = c_{\mathcal{A}} \in D_{\mathcal{A}}$$

Exercice 3 *On considère la $\{0(0), s(1), +(2), \times(2)\}$-structure $(\mathbb{N}, 0, s, +, \times)$.*

Évaluer le terme $+(s(s(s(0))), s(s(0)))$ dans cette structure.

D'après le Théorème 1, on établit que :

Lemme 2 *Pour tout $t \in \mathcal{T}(\mathcal{F}, X)$, on a $[\![t]\!]_{\mathcal{T}(\mathcal{F},X),\sigma} = \hat{\sigma}(t)$, et on notera plus simplement $t\sigma$.*

Preuve □

Définition 6 *Une* substitution *est une $\mathcal{T}(\mathcal{F}, X)$-affectation σ tel que $dom(\sigma)$ est fini.*

Les affectation σ de $\mathcal{T}(\mathcal{F}, X)$ dans $\mathcal{T}(\mathcal{F})$ de domaine fini sont des substitutions closes.

Exemple 6 $[\![+(x,x)]\!]_{\mathbb{N},\{x\mapsto 1\}} = 2$ *et* $[\![+(x,x)]\!]_{\mathcal{T}(\mathcal{F},X),\{x\mapsto s(y)\}} = +(s(y),s(y))$.

Lemme 3 (Lemme de substitution pour les termes) *Soit u un terme, x une variable, et t un autre terme. On a :*

$$[\![u\{x\mapsto t\}]\!]_{\mathcal{A},\lambda} \text{ ssi } [\![u]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}}.$$

Preuve Soit $\mathcal{P}r(u)$ la propriété donné dans le lemme. Nous allons démontrer que $\mathcal{P}r(u)$ est vraie pour tout terme u, par induction structurelle sur u.

Le cas de base porte sur une variable. Soit y un symbole de variable différent de x. On a :

$$[\![y\{x\mapsto t\}]\!]_{\mathcal{A},\lambda} = [\![y]\!]_{\mathcal{A},\lambda} = \lambda(y) = (\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda})(y) = [\![y]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}}$$

d'où $\mathcal{P}r(y)$.

Pour la variable x on a :

$$[\![x\{x\mapsto t\}]\!]_{\mathcal{A},\lambda} = [\![t]\!]_{\mathcal{A},\lambda} = (\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda})(x) = [\![x]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}}$$

d'où $\mathcal{P}r(x)$.

Considérons maintenant n termes $t_1,\ldots,t_n$ pour lesquels on a $\mathcal{P}r(t_1),\ldots,\mathcal{P}r(t_n)$ vraies. Montrons que $\mathcal{P}r(f(t_1,t_2,\ldots,t_n))$. On a :

$$\begin{aligned}
[\![f(t_1,\ldots,t_n)\{x\mapsto t\}]\!]_{\mathcal{A},\lambda} &= [\![f(t_1\{x\mapsto t\},\ldots,t_n\{x\mapsto t\})]\!]_{\mathcal{A},\lambda} \\
&= f^{\mathcal{A}}([\![t_1\{x\mapsto t\}]\!]_{\mathcal{A},\lambda},\ldots,[\![t_n\{x\mapsto t\}]\!]_{\mathcal{A},\lambda}) \\
&= f^{\mathcal{A}}([\![t_1]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}]},\ldots,[\![t_n]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}]}) \\
&= [\![f(t_1,\ldots,t_n)]\!]_{\mathcal{A},\lambda[x\mapsto [\![t]\!]_{\mathcal{A},\lambda}]}
\end{aligned}$$

d'où $\mathcal{P}r(f(t_1,t_2,\ldots,t_n))$.

□

2.4 Classes équationnelles de $\mathcal{F}$-algèbres

Une *identité* (ou encore *équation*) sur $\mathcal{F}$ a la forme $s=t$, où s et t de $\mathcal{T}(\mathcal{F},X)$.

Exemple 7 *Soit $\mathcal{F}=\{e(0),\boldsymbol{i}(1),\star(2)\}$. On peut considérer l'ensemble d'identités*

$$\begin{cases} e\star x = x \\ x\star e = x \\ x\star \boldsymbol{i}(x) = e \\ \boldsymbol{i}(x)\star x = e \\ x\star(y\star z) = (x\star y)\star z \end{cases}$$

Une identité $s=t$ est vraie dans une $\mathcal{F}$-algèbre $\mathcal{A}$ si

$$[\![s]\!]_{\mathcal{A},\lambda} = [\![t]\!]_{\mathcal{A},\lambda}, \text{ pour toute } \mathcal{A}\text{-affectation } \lambda.$$

On dit alors que $\mathcal{A}$ *satisfait* $s=t$, et on le notera $\mathcal{A}\models s=t$.

On peut aussi s'intéresser au problème de savoir si étant données $\mathcal{F}$-algèbre $\mathcal{A}$ et une équation $s=t$, où s et t de $\mathcal{T}(\mathcal{F},X)$, il existe une $\mathcal{A}$-affectation λ telle que $[\![s]\!]_{\mathcal{A},\lambda} = [\![t]\!]_{\mathcal{A},\lambda}$. (voir TD).

Une classe $\mathcal{K}$ de $\mathcal{F}$-algèbres est *équationnelle* s'il existe un ensemble $\mathbf{E}$ sur $\mathcal{F}$ telle que

$$\mathcal{K} = \{\mathcal{A} \mid \mathcal{A}\models s=t, \text{ pour toute } s=t\in\mathbf{E}\}$$

Chapter 3

Le calcul des prédicats

3.1 Un exemple

On commence par des formules où les termes sont réduits à des variables.

On pourra écrire des formules telles que :

1. $\varphi_G : \forall x \forall y \forall z (P(x,y) \wedge P(y,z)) \rightarrow G(x,z)$
2. $\varphi_P : \forall x \exists y P(y,x)$
3. $\varphi_C : \forall x \exists y G(y,x)$
4. $\varphi_D : \forall x \forall z P(z, f(x)) \rightarrow G(z,x)$
5. $\varphi_F : (\varphi_G \wedge \varphi_P) \rightarrow \varphi_C$

On peut évaluer ces formules si :

- on ne sait dans quel ensemble les variables x, y, z prennent leur valeur, ou
- on ne connaît pas non plus la fonction f, ou
- on ne connaît pas les relations binaires P et G.

Il faut fixer une *interprétation* pour pouvoir évaluer les formules.

Interprétation 1 Les objets/individus sont les êtres humains, avec tout ce qui en découle.

$P(x,y)$ signifie que "x est le père de y", $G(x,y)$ signifie que "x est un grand-père de y", et f retourne la mère d'un individu.

- $\varphi_G = \forall x \forall y \forall z (P(x,y) \wedge P(y,z)) \rightarrow G(x,z)$ signifie "pour tous êtres humains x, y, z, si x est le père de y et y est le père de z, alors x est un grand-père de z".
- $\varphi_P = \forall x \exists y P(y,x)$ signifie "pour tout individu x il existe un individu y tel que y est le père de x" c-à-d. "tout individu a un père".

Les objets/individus sont les êtres humains, avec tout ce qui en découle.

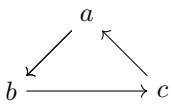
$P(x,y)$ signifie que "x est le père de y", $G(x,y)$ signifie que "x est un grand-père de y", et f retourne la mère d'un individu.

- $\varphi_C = \forall x \exists y G(y,x)$ signifie "pour tout individu x il existe un individu y tel que y est le grand père de x", c-à-d. "tout individu a un grand-père".
- $\varphi_D = \forall x \forall z P(z, f(x)) \to G(z,x)$ signifie "si z est le père de la mère de x, alors z est un grand-père de x".

Les quatre formules sont vraies dans l'Interprétation 1.
L'énoncé $\varphi_F = ((\varphi_G \wedge \varphi_P) \to \varphi_C)$ est donc aussi vrai.

Remarque 5 *Les deux formules φ_P et φ_G sont loin de modéliser toutes les propriétés des relations P et G, respectivement : par exemple, on n'a pas énoncé que "le père de chaque individu est unique", ni qu'"un individu x peut être le grand-père d'un autre z sans qu'il existe un individu dont x soit le père et qui soit le père de z".*

Interprétation 2 On se place dans ce graphe, et les objets sont les trois sommets a, b, c du graphe :



- P décrit la relation prédécesseur, donc les couples (a,b), (b,c) et (c,a), et G décrit la relation successeur, donc les couples (b,a), (c,b) et (a,c).
- La fonction f s'interpréte comme $a \mapsto a, b \mapsto b, c \mapsto a$.
- $\varphi_P = \forall x \exists y P(y,x)$ signifie "tout point a un prédécesseur immédiat"

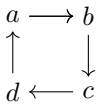
  Elle est vraie dans ce graphe.
- $\varphi_G = \forall x \forall y \forall z (P(x,y) \wedge P(y,z)) \to G(x,z)$ signifie "pour tous points x, y, z, si x précède immédiatement y et si y précède immédiatement z, alors x suit immédiatement z"

  Elle est vraie dans ce graphe.
- $\varphi_C = \forall x \exists y G(y,x)$ "tout point a un successeur immédiat"

  Elle est vraie dans ce graphe.
- $\varphi_F = ((\varphi_P \wedge \varphi_G) \to \varphi_C)$ est donc vraie dans ce graphe.
- $\varphi_D = \forall x \forall z P(z, f(x)) \to G(z,x)$ "pour tout z et pour tout x, si z précède immédiatement $f(x)$, alors z suit immédiatement x".

  Elle est fausse dans ce graphe.

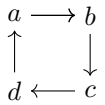
Interprétation 3 On se place dans ce graphe, et les objets sont les quatre sommets a, b, c, d du graphe :



- $P(x,y)$ "x précède immédiatement y sur le graphe".
- $G(x,y)$ "x suit immédiatement y sur le graphe".
- $\varphi_P = \forall x \exists y P(y,x)$ " tout point a un prédécesseur immédiat". Elle est vraie.
- $\varphi_G = \forall x \forall y \forall z (P(x,y) \wedge P(y,z)) \to G(x,z)$ "pour tous points x, y, z, si x précède immédiatement y et si y précède immédiatement z, alors x suit immédiatement z". Elle est fausse.
- $\varphi_C = \forall x \exists y G(y,x)$ "tout point a un successeur immédiat". Elle est vraie.

$\varphi_F = ((\varphi_P \wedge \varphi_G) \to \varphi_C)$ est donc vraie dans cette interprétation.

Interprétation 4 On se place dans ce graphe, et les objets sont les quatre sommets a, b, c, d du graphe :



- $P(x, y)$ "x précède immédiatement y" et $G(x, y)$ "pour aller de x à y on rencontre exactement un point z différent de x et de y".
- $\varphi_P = \forall x \exists y P(y, x)$ "tout point a un prédécesseur immédiat". Elle est vraie.
- $\varphi_G = \forall x \forall y \forall z (P(x, y) \wedge P(y, z)) \rightarrow G(x, z)$ "pour tous points x, y, z, si x précède immédiatement y et si y précède immédiatement z, alors x suit immédiatement z". Elle est vraie.
- $\varphi_C = \forall x \exists y G(y, x)$ "tout point a un successeur immédiat". Elle est vraie.

$\varphi_F = ((\varphi_P \wedge \varphi_G) \rightarrow \varphi_C)$ est donc vraie dans cette interprétation.

Interprétation 5 Les objets sont les éléments de $\mathbb{N}$, les entiers naturels.

- $P(x, y)$ "$x = y + 1$". Ex. $P(5, 4)$ est vraie, mais $P(4, 5)$ est fausse.
- $G(x, y)$ "$x = y + 2$". Ex. $G(6, 4)$ est vraie, mais $G(4, 6)$ est fausse.
- $\varphi_P = \forall x \exists y P(y, x)$ "pour tout $x \in \mathbb{N}$, il existe $y \in \mathbb{N}$ tel que $y = x + 1$". Elle est vraie.
- $\varphi_G = \forall x \forall y \forall z (P(x, y) \wedge P(y, z)) \rightarrow G(x, z)$ "pour tous $x, y, z \in \mathbb{N}$, si $x = y+1$ et $z = y+1$ alors $z = x+2$". Elle est vraie.
- $\varphi_C = \forall x \exists y G(y, x)$ "pour tout $x \in \mathbb{N}$, il existe $y \in \mathbb{N}$ tel que $y = x + 2$". Elle est vraie.

$\varphi_F = ((\varphi_P \wedge \varphi_G) \rightarrow \varphi_C)$ est donc vraie dans cette interprétation.

Interprétation 6 Les objets sont les éléments de $\mathbb{N}$, les entiers naturels.

- $P(x, y)$ "$y = x + 1$". Ex. $P(5, 4)$ est vraie, mais $P(4, 5)$ est fausse.
- $G(x, y)$ "$y = x + 2$". Ex. $G(6, 4)$ est vraie, mais $G(4, 6)$ est fausse.
- $\varphi_P = \forall x \exists y P(y, x)$ "pour tout $x \in \mathbb{N}$, il existe un entier y tel que $x = y + 1$". Elle est fausse.
- $\varphi_G = \forall x \forall y \forall z (P(x, y) \wedge P(y, z)) \rightarrow G(x, z)$ "pour tous entiers x, y, z, si $x = y + 1$ et $z = y + 1$ alors $z = x + 2$". Elle est vraie.
- $\varphi_C = \forall x \exists y G(y, x)$ "pour tout $x \in \mathbb{N}$, il existe $z \in setn$ tel que $x = z + 2$". Elle est fausse.

$\varphi_F = ((\varphi_P \wedge \varphi_G) \rightarrow \varphi_C)$ est donc vraie dans cette interprétation.

3.2 Vue d'ensemble de la logique du premier ordre

La *logique du premier ordre* permet d'énoncés des propriétés dans lesquels les objets sont mis en relation. En ce sens, il est plus riche/expressif que le calcul propositionnel.

La notion d'objet est capturée dans les $\mathcal{F}$-algèbres. On se donne donc pour commencer :

- des symboles de *variables* $x, y, z, \ldots$ qui prennent leur valeurs dans l'ensemble des *objets* (ou entités) du discours.

- des symboles de *fonctions* $(c, f(x), g(x, y), \ldots)$ permettant d'opérer sur les objets.

Pour obtenir des énoncés (formules) on se donne aussi des symboles de *relations* $(P(x), Q(x, y), R(x, y, z), \ldots)$ qui permettront de relier les objets entre eux.

Pour former les phrases du langage du premier ordre on utilise :

- des formules *atomiques* : celles basées sur les relations, ex. $Q(c, f(x))$, qui s'évaluent à `vrai` ou `faux` (selon les *interprétations*).
- des combinaison de formules atomiques à l'aide des connecteurs classiques de la logique du calcul propositionnel,
- des *quantificateurs*, de la forme $\forall x$ et $\exists x$ (il y en a autant que de variables x), dont la sémantique sera sans surprise celle dans les mathématiques usuelles.

Chapter 4

Syntaxe de la logique du premier ordre

Pour toute la suite, on se fixe un ensemble $X = \{x, y, z, \ldots\}$ de variables.

Définition 7 *Une* signature *est une paire* $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ *où :*

- $\mathcal{F}$ *est un ensemble de symboles de fonctions avec leur arité;*
- $\mathcal{P}$ *est un ensemble de symboles de relations avec leur arité.*

De même que pour les symboles de fonctions, on notera $\mathcal{P}_n$ *l'ensemble des synboles de relations d'arité* n*.*

Exemple 8 (Pour les ensembles) *On considère la signature* $\mathcal{F} = \{\emptyset(0), \cap(2), \cup(2), \setminus(2)\}$ *et* $\mathcal{P} = \{=(2), \subseteq (2)\}$.

Exemple 9 $\mathcal{S} = \{\emptyset : \mathcal{F}_0\ , \cap : \mathcal{F}_2\ , \setminus : \mathcal{F}_2\ , =\ : \mathcal{P}_2\ , \subseteq\ : \mathcal{P}_2\ \}$.

Exercice 4

- *Quel(s) signatures(s) proposeriez-vous pour les entiers naturels ?*
- *Pour les listes d'entiers naturels ?*
- *Sur quelle signature (minimale) sont construites les formules de l'introduction ?*

Il ne faut pas confondre "fonctions" (dans $\mathcal{F}$) et "relations" (dans $\mathcal{P}$) :

- les symboles de fonction, comme + ou 0, sont utilisés pour décrire des termes et dénoteront les éléments d'un domaine, et
- les symboles de relation, comme $\geq$, qui sont utilisés pour décrire des propriétés entre les éléments.

Exemple 10 *Si* $1, 2, 4 \in \mathcal{F}_0$, $+ \in \mathcal{F}_2$ *et* $\geq \in \mathcal{P}_2$ *:*

- $+(1, 2)$ *est un terme (interprété comme un élément du domaine),*
- $\geq (+(1, 2), 4)$ *est une formule (interprétée comme un énoncé).*

4.1 Les formules atomiques/prédicats

Étant donnée une signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$, on construit les formules atomiques en appliquant un symbole de relation à des termes.

Définition 8 (Formule atomique) *Une* formule atomique sur $\mathcal{S}$ *est de la forme* $R(t_1, \ldots, t_n)$ *où* $t_1, t_2, \ldots, t_n \in \mathcal{T}(\mathcal{F}, X)$ *sont des termes, et* $R \in \mathcal{P}_n$*, c-à-d. un symbole de prédicat d'arité* n*.*

Remarque 6 *Les éléments de* $\mathcal{P}_0$ *sont des prédicats d'arité 0, c'est-à-dire qu'il ne prenne pas d'arguments. En un sens, ils correspondent à des variables propositionnelles.*

Exemple 11 *Avec* $\mathcal{F} = \{\emptyset(0), \cap(2), \cup(2), \setminus(2)\}$ *et* $\mathcal{P} = \{=(2), \subseteq(2)\}$*, on peut écrire la formule atomique*

$$\subseteq (\cap(x, \setminus(y, x), \emptyset)$$

que l'on écrit plus naturellement si nous parlons d'ensemble :

$$x \cap (y \setminus x) \subseteq \emptyset.$$

Remarque 7 *On convient que le symbole de relation* $\bot$ (d'arité 0) *représente l'énoncé toujours faux, et* $=$ (d'arité 2) *représente l'égalité.*

4.2 L'ensemble des formules

On se fixe une signature du premier ordre $\mathcal{S} = (\mathcal{F}, \mathcal{P})$, et un ensemble X de variables.

Définition 9 (Formule du premier ordre sur $\mathcal{S}$) *On note* $\mathcal{L}^1_{\mathcal{S}}$*, ou tout simplement* $\mathcal{L}^1$ *(quand* $\mathcal{S}$ *est clair), l'ensemble des formules du premier ordre (sur* $\mathcal{S}$*) défini par induction.*

1. *toute formule atomique appartient à* $\mathcal{L}^1$
2. *si* $\varphi \in \mathcal{L}^1$*, alors* $\neg\varphi \in \mathcal{L}^1$
3. *si* $\varphi, \psi \in \mathcal{L}^1$*, alors* $\varphi \wedge \psi, \varphi \vee \varphi, \varphi \to \psi \in \mathcal{L}^1$
4. *si* $\varphi \in \mathcal{L}^1$ *et* $x \in X$*, alors* $\forall x\varphi, \exists x\varphi \in \mathcal{L}^1$*.*

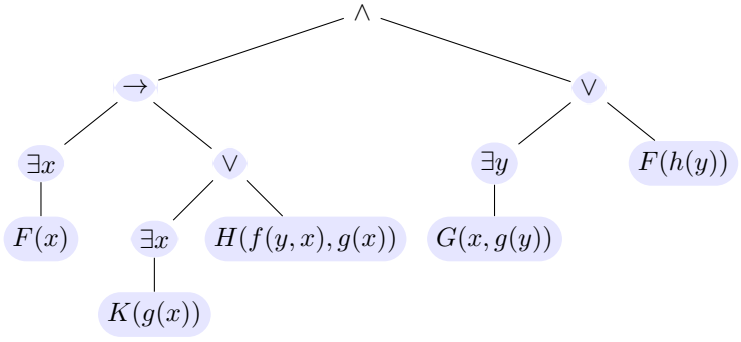
On notera $Var(\varphi)$ *l'ensemble des variables apparaissant dans la formule* φ*.*

Exercice 5 *Donnez des exemples de formules sur la signature* $\mathcal{S} = \{\{\emptyset(0), \cap(2), \cup(2), \setminus(2)\}, \{=(2), \subseteq(2)\}\}$ *pour manipuler des ensembles.*

Comme pour le calcul propositionnel, on peut voir une formule comme un arbre.

Exemple 12 *Soit la formule*

$$[(\exists x F(x)) \to (\exists x K(g(x)) \vee H(f(y,x), g(x)))] \wedge [(\exists y G(x, g(y))) \vee F(h(y))]$$



Remarque 8 *Les feuilles de l'arbre sont des formules atomiques (de façon similaire au cas des propositions pour le calcul propositionnel).*

4.3 Variables libres et liées

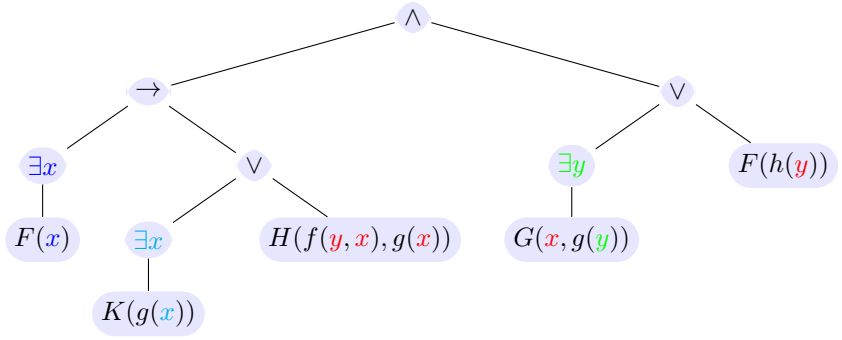
Définition 10 (Occurrence de variable libre, liée) *On dit qu'une occurrence d'une variable x est liée lorsqu'elle appartient à une sous-formule précédée d'un quantificateur $\forall x$ ou $\exists x$. Sinon, on dit qu'elle est libre.*

Remarque 9 *La distinction entre occurrence de variable libre et occurrence de variable liée est importante : une occurrence de variable liée ne possède pas d'identité propre. Elle peut par exemple être remplacée par n'importe quel nom de variable qui n'apparaît pas dans la formule.*

Exemple 13 *On verra que la formule $\exists x(x < y)$ est "équivalente" à la formule $\exists z(z < y)$ car on a simplement renommé la variable liée x en z, mais qu'elle n'est pas équivalent à $\exists x(x < z)$ et ni moins à $\exists y(y < y)$.*

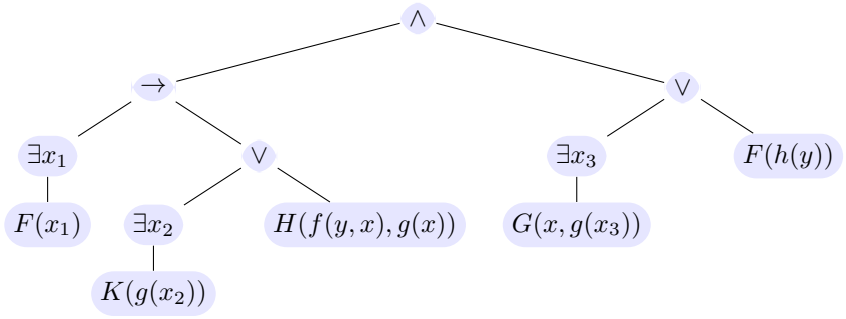
Remarque 10 *On pourrait faire une analogie entre variable liée dans une formule et variable locale à une méthode en programmation.*

Exemple 14 *Considérons la formule décrite par l'arbre syntaxique suivant :*



Les occurences x et y sont libres dans la formule, les autres sont liées.

On reste "équivalent" (dans un sens qui sera précisé plus tard) si on renomme les occurrences liées des variables liées (justfication avec la sémantique).



Définition 11 (Variable libre, liée) *L'ensemble des* variables libres *de φ, noté $\texttt{vlibres}(\varphi)$, est l'ensemble des variables de $Var(\varphi)$ qui ont une occurrence libre dans φ. Il est défini par induction sur φ :*

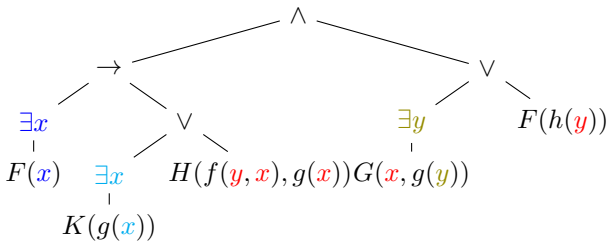
- *si φ est une formule atomique : $\texttt{vlibres}(\varphi) = Var(\varphi)$;*

- $\texttt{vlibres}(\neg\varphi) = \texttt{vlibres}(\varphi)$;
- $\texttt{vlibres}(\varphi \wedge \psi) = \texttt{vlibres}(\varphi \vee \psi) = \texttt{vlibres}(\varphi \to \psi) = \texttt{vlibres}(\varphi) \cup \texttt{vlibres}(\psi)$;
- $\texttt{vlibres}(\forall x\varphi) = \texttt{vlibres}(\exists x\varphi) = \texttt{vlibres}(\varphi) \setminus \{x\}$

On notera $\texttt{vliees}(\varphi)$ *l'ensemble* $Var(\varphi) \setminus \texttt{vlibres}(\varphi)$ *et on appelera* variables liées *ses éléments.*

Définition 12 (Formule close) *Une formule* φ *est* close *si elle ne possède aucune variable libre,* i.e., $\texttt{vlibres}(\varphi) = \emptyset$.

Exemple 15 *Soit la formule* φ *:*



On a alors $\texttt{vlibres}(\varphi) = \{x, y\}$.

Chapter 5

Sémantique de la logique du premier ordre

Maintenant que nous sommes familiarisés avec la syntaxe de la logique des prédicats, nous donnons une signification aux symboles de manières à interpréter les formules.

Soit $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ une signature de la logique du premier.

On évalue les formules de $\mathcal{L}^1_{\mathcal{S}}$ sur des $\mathcal{S}$-*structures*, c'est à dire des $\mathcal{F}$-algèbres enrichies d'une interprétation pour chaque symbole de relation.

5.1 Structures

Définition 13 (Structures) *Une* $\mathcal{S}$-structure*, ou plus simplement* structure *si la signature* $\mathcal{S}$ *est claire, est la donnée*

1. *d'une* $\mathcal{F}$*-algèbre* $\mathcal{A} = (D_{\mathcal{A}}, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$*, et*
2. *d'un sous-ensemble* $P^{\mathcal{A}} \subseteq (D_{\mathcal{A}})^n$*, pour chaque symbole de relation* $P \in \mathcal{P}$*.*

On conservera la notation $\mathcal{A}$ *pour les structures.*

Exemple 16 *Soit la signature* $\mathcal{S} = (\{o(0), f(1)\}, \{P(2)\})$*.*

1. *Soit* $\mathcal{A}$ *la structure/interprétation définie par :*
   - $D_{\mathcal{A}} = \mathbb{N}$
   - $o^{\mathcal{A}} = 0$
   - $f^{\mathcal{A}}(n) = n + 1$
   - $P^{\mathcal{A}} = \{(n, n') \mid n \leq n'\}$
2. *Soit* $\mathcal{A}'$ *la structure/interprétation définie par :*
   - $D_{\mathcal{A}'} = \{Paul, Anne, Ida\}$
   - $o^{\mathcal{A}'} = Paul$
   - $f^{\mathcal{A}'}(Paul) = Anne$*,* $f^{\mathcal{A}'}(Anne) = Paul$*,* $f^{\mathcal{A}'}(Ida) = Anne$*, par exemple pour "héritier"*
   - $P^{\mathcal{A}'} = \{(Anne, Paul), (Paul, Ida), (Anne, Ida)\}$*, par exemple pour "est plus âgé que"*

Quelques structures importantes Sur la signature $\{\{0, s, +, \times\}, \{=, \leq\}\}$.

Domaine	fonc.	rel.	Nom de la structure
$\mathbb{N}$	$0, s, +, \times$	$=, \leq$	Arithmétique vraie
$\mathbb{N}$	$0, s, +$	$=, \leq$	Arithmétique de Presburger
$\boldsymbol{R}$	$0, s, +, \times$	$=, \leq$	Théorie des réels
$\boldsymbol{R}$	$0, s, +$	$=, \leq$	Théorie additive des réels

5.2 Sémantique de la logique du premier ordre

On se fixe une signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ et un ensemble de variables X, une $\mathcal{S}$-structure $\mathcal{A} = (D, \{f^{\mathcal{A}}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}}\}_{P\in\mathcal{P}})$, et une affectation $\lambda \in \text{Affect}_{\mathcal{A}}$.

On définit l'expression $\mathcal{A}, \lambda \models \varphi$ qui exprime que la formule φ est vraie/fausse dans $\mathcal{A}$ pour l'affectation λ.

Intuitivement, pour établir que $\mathcal{A}, \lambda \models \exists x \forall P(f(x), c)$, il faut :

1. trouver un élément $d \in D_{\mathcal{A}}$ à affecter à x (donc on considérera l'affectation $\lambda[x \mapsto d]$),
2. vérifier que $\mathcal{A}, \lambda[x \mapsto d] \models P(f(x), y)$, c-à-d. $(f^{\mathcal{A}}(d), c^{\mathcal{A}}) \in P_{\mathcal{A}}$.

Vérité d'une formule atomique On a fixé une signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ et un ensemble de variables X, une $\mathcal{S}$-structure $\mathcal{A}$, et $\lambda \in \text{Affect}_{\mathcal{A}}$ une affectation pour les variables de X.

Définition 14 (Vérité d'une formule atomique dans $\mathcal{A}$ pour λ) *On définit $\mathcal{A}, \lambda \models R(t_1, \ldots, t_n)$ par :*

$$(\llbracket t_1 \rrbracket_{\mathcal{A},\lambda}, \llbracket t_2 \rrbracket_{\mathcal{A},\lambda}, \ldots, \llbracket t_n \rrbracket_{\mathcal{A},\lambda}) \in P^{\mathcal{A}}.$$

Exemple 17 *Considérons la formule atomique $P(f(x), y)$ que l'on évalue dans la structure $\mathcal{N} = (\mathbb{N}, f_{\mathcal{N}}(n) := n + 1, P_{\mathcal{N}} := \geq)$, et pour la affectation λ définie par $[x \mapsto 42, y \mapsto 3, z \mapsto \ldots]$.*

Alors, on a

$$\mathcal{N}, [x \mapsto 42, y \mapsto 3, z \mapsto \ldots] \models P(f(x), y)$$

dès lors que $(f_{\mathcal{N}}(\lambda(x)), \lambda(y)) \in \geq$, ce qui est le cas puisque $42 + 1 \geq 3$.

Exercice 6 *Proposer $\mathcal{A}$ et λ pour lesquels $\mathcal{A}, \lambda \not\models P(f(x), y)$.*

Vérité d'une formule composite Maintenant que les valeurs des formules atomiques sont définis, on peut définir les valeurs de formules plus complexes.

Définition 15 (Vérité d'une formule dans $\mathcal{A}$ pour λ) *On définit $\mathcal{A}, \lambda \models \varphi$ par induction sur φ.*

- $\mathcal{A}, \lambda \models \neg\varphi$ *ssi* $\mathcal{A}, \lambda \not\models \varphi$
- $\mathcal{A}, \lambda \models \varphi \wedge \psi$ *ssi* $\mathcal{A}, \lambda \models \varphi$ *et* $\mathcal{A}, \lambda \models \psi$
- $\mathcal{A}, \lambda \models \varphi \vee \psi$ *ssi* $\mathcal{A}, \lambda \models \varphi$ *ou* $\mathcal{A}, \lambda \models \psi$
- $\mathcal{A}, \lambda \not\models \varphi \to \psi$ *ssi* $\mathcal{A}, \lambda \models \varphi$ *et* $\mathcal{A}, \lambda \not\models \psi$
- $\mathcal{A}, \lambda \models \exists x\varphi$ *ssi il existe* $d \in D_{\mathcal{A}}$ *tel que* $\mathcal{A}, \lambda[x \mapsto d] \models \varphi$
- $\mathcal{A}, \lambda \models \forall x\varphi$ *ssi pour tout* $d \in D_{\mathcal{A}}$, $\mathcal{A}, \lambda[x \mapsto d] \models \varphi$

Remarque 11 *On a vu que les faits $\mathcal{A}, \lambda \models \exists x\varphi$ et $\mathcal{A}, \lambda \models \forall x\varphi$ sont indépendants de la valeur de $\lambda(x)$. Plus généralement, ils ne dépendent que des affectations des variables libres de la formule, et pas de ses variables liées.*

Lorsque φ est close, on notera plus simplement $\mathcal{A} \models \varphi$.

5.3 Autre angle de vue : les jeux d'évaluation

Chapter 6

Notions centrales qui découlent de la sémantique

En réalité, la notion pertinente est celle de formule close, les formules non closes ne servent qu'à donner la sémantique.

Comme pour le calcul propositionnel, on définit les notions de :

- modèles d'une formule close
- satisfaisabilité et validité
- équivalence
- conséquence logique

6.1 Satisfaisabilité et validité d'une formule

Définition 16 (Satisfaisabilité d'une formule) *Une formule close $\varphi \in \mathcal{L}^1$ est* insatisfaisable *si elle n'a pas de modèle, sinon elle est* satisfaisable.

Définition 17 (Validité d'une formule) *Une formule close $\varphi \in \mathcal{L}^1$ sur $\mathcal{S}$ est* valide, *noté $\models \varphi$, si pour toute $\mathcal{S}$-structure $\mathcal{A}$, on a $\mathcal{A} \models \varphi$.*

On définit le problème de décision suivant qui consiste à décider si une formule est valide :

VALIDE

Entrée : $\varphi \in \mathcal{L}^1$
Sortie : $\models \varphi$?

Théorème 4 *Le problème* VALIDE *est indécidable, même si on se restreint à une signature avec un seul symbole de constante, deux symboles de fonctions d'arité et un symbole de prédicat d'arité 2.*

Pour montrer le théorème 4, on réduit le problème POST, qui est indécidable, au problème VALIDE. On considère l'alphabet $\{a, b\}$ (quitte à recoder un alphabet fini Σ quelconque sur ces deux lettres).

POST

Entrée : un ensemble fini de dominos $\{\begin{array}{|c|}\hline u_i \\ \hline v_i \\ \hline \end{array}\}_{1\leq i\leq N}$
où les u_i et v_i sont des mots sur Σ.
Sortie : "oui" s'il existe une séquence finie de dominos

$$\begin{array}{|c|}\hline u_{i_1} \\ \hline v_{i_1} \\ \hline \end{array}\begin{array}{|c|}\hline u_{i_2} \\ \hline v_{i_2} \\ \hline \end{array}\cdots\begin{array}{|c|}\hline u_{i_p} \\ \hline v_{i_p} \\ \hline \end{array}$$

avec $u_{i_1}u_{i_2}\dots u_{i_p} = v_{i_1}v_{i_2}\dots v_{i_p}$ (les mots haut et bas sont égaux);
"non" sinon.

Sans perte de généralité, on pourra supposer aussi que pour chaque domino, l'un des mots u_i ou v_i est non vide, sinon le problème a la solution triviale $\begin{array}{|c|}\hline \epsilon \\ \hline \epsilon \\ \hline \end{array}$.

Soit $I = \{\begin{array}{|c|}\hline u_i \\ \hline v_i \\ \hline \end{array}\}_{1\leq i\leq N}$ une instance de POST. On la transforme en une formule φ_I sur la signature $\mathcal{S} = \{\{\epsilon(0), a(1), b(1)\}, \{SD(2)\}\}$ qui permet de parler des mots finis :

- le symbole de constante ϵ intuitivement dénote le mot vide ;
- les symboles de fonctions a et b intuitivement ajoutent la lettre a ou b en début de mot. D'ailleurs, pour chaque mot $u = \ell_1\ell_2\dots\ell_k \in \Sigma^*$, on note le terme $\ell_1(\ell_2(\dots(\ell_k(x)\dots))$ de façon plus abrégé : $\ell_1\ell_2\dots\ell_k(x)$. Ainsi, par exemple $aaba(x)$ est l'écriture succincte du terme $a(a(b(a(x))))$. Réciproquement, à tout terme clos t on peut associer un mot de Σ^*. Par exemple, $a(a(b(a(\epsilon))))$ on associe le mot $aaba$.
- le prédicat $SD(t, t')$ signifie intuitivement qu'il existe une succession de dominos avec le mot dénoté par t en haut et le mot dénoté par t' en bas. D'ailleurs, on note $\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}$ une succession de dominos de I telle que le mot du haut soit u et celui du bas v.

On construit alors la formule suivante :

$$\varphi_I := SD(\epsilon,\epsilon) \wedge \bigwedge_{i=1}^{N} \Big(\forall x\forall y\big(SD(x,y) \to SD(u_i(x), v_i(y))\big)\Big) \to \exists x\Big(SD(a(x),a(x)) \vee SD(b(x),b(x))\Big)$$

Intuitivement, la formule atomique $SD(t, t')$ signifie qu'il existe une succession de dominos $\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}$ où t dénote u et t' dénote v. Ainsi, la formule φ_I dit que si le prédicat SD vérifie bien comme l'existence d'une telle succession (succession vide, et on obtient une succession en accrochant un domino $\begin{array}{|c|}\hline u_i \\ \hline v_i \\ \hline \end{array}$), alors on trouve un mot non vide (dénoté par $a(x)$ ou alors $b(x)$) et une succession de dominos avec écrit ce mot en haut et en bas.

Premièrement, on se convainc que la formule φ_I est calculable algorithmique depuis une instance I. Deuxièmement, on démontre la correction de notre réduction. C'est l'objet de la proposition suivante.

Proposition 5 *$I \in$ POST ssi $\varphi_I \in$ VALIDE.*

Preuve $\boxed{\Longleftarrow}$ Supposons que φ_I est valide. On considère le modèle de Herbrand $\mathcal{H}$ défini par

$$SD^{\mathcal{H}} = \{(u, v) \mid \begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array} \text{ existe}\}.$$

On se permet de confondre un terme clos et un mot, e.g. le terme clos $a(b(\epsilon))$ et ab. Maintenant, nous allons à montrer que $\mathcal{H}$ satisfait la prémisse de φ_I. Cela achèvera la démonstration puisque si la conclusion est φ_I est vraie, cela implique l'existence d'une suite correcte de dominos.

D'une part, si l'on ne met aucun domino, par définition, on a $\boxed{\begin{matrix}\epsilon\\ \epsilon\end{matrix}}$.On a donc $\mathcal{H} \models SD(\epsilon, \epsilon)$.

Enfin pour tout mot $u, v \in \Sigma^*$, on a si $\mathcal{H}\begin{bmatrix} x := u \\ y := v \end{bmatrix} \models SD(x, y)$ cela veut dire que $\boxed{\begin{matrix}u\\ v\end{matrix}}$ existe. Donc en concaténant $\boxed{\begin{matrix}u\\ v\end{matrix}}$ et $\boxed{\begin{matrix}u_i\\ v_i\end{matrix}}$, il existe bien $\boxed{\begin{matrix}uu_i\\ vv_i\end{matrix}}$ pour tout $i \in \{1, \dots, N\}$, et donc $\mathcal{H}\begin{bmatrix} x := u \\ y := v \end{bmatrix} \models SD(u_i(x), v_i(y))$. Donc pour tout $i \in \{1, \dots, N\}$, $\mathcal{H} \models \Big(\forall x \forall y \big(SD(x, y) \to SD(u_i(x), v_i(y))\big)\Big)$ Ainsi $\mathcal{H} \models \varphi$.

Comme φ_I est valide, elle est vraie dans $\mathcal{H}$. La prémisse de φ_I était vraie dans $\mathcal{H}$, la conclusion de φ_I est aussi vraie dans $\mathcal{H}$. Par exemple, Sans perte de généralité, on peut dire que $\mathcal{H} \models \exists x\ SD(a(x), a(x))$ (le cas pour b serait similaire). Donc il existe un mot $\alpha \in \Sigma^*$ tel que $\mathcal{H}\big[x := \alpha\big] \models SD(a(x), a(x))$. Donc $\boxed{\begin{matrix}\alpha a\\ \alpha a\end{matrix}}$ existe. Ainsi par définition, il existe une succession de dominos telle que le mot du haut et celui du bas soit le même. L'instance I est bien une instance positive du problème POST.

$\boxed{\Longrightarrow}$ On suppose que I est une instance positive de POST donc il existe $m \geq 1$ tel qu'il existe $i_1, \dots, i_m \in \{1, \dots, N\}$ avec $u_{i_1}u_{i_2}\dots u_{i_m} = v_{i_1}v_{i_2}\dots v_{i_m}$. Soit $\mathcal{A}$ un modèle. Supposons que $\mathcal{A}$ satisfait la prémisse de φ_I, montrons que la conclusion de φ_I est vraie dans $\mathcal{A}$. Pour cela, nous allons montrer par récurrence sur k que $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_k}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_k}(\epsilon) \end{bmatrix} \models SD(x, y)$.

Initialisation : pour $k = 0$, on a $\mathcal{A} \models SD(\epsilon, \epsilon)$ car la prémisse le dit.

Hérédité : Supposons que $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_k}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_k}(\epsilon) \end{bmatrix} \models SD(x, y)$. Montrons que $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_{k+1}}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_{k+1}}(\epsilon) \end{bmatrix} \models SD(x, y)$. On sait que $\mathcal{A} \models \forall x \forall y \big(SD(x, y) \to SD(u_i(x), v_i(y))\big)$ pour tout $i \in \{1, \dots, N\}$, en particulier, pour $i = i_{k+1}$, on a $\mathcal{A} \models \forall x \forall y \big(SD(x, y) \to SD(u_{i_{k+1}}(x), v_{i_{k+1}}(y))\big)$. Donc $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_k}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_k}(\epsilon) \end{bmatrix} \models \big(SD(x, y) \to SD!(u_{i_{k+1}}(x), v_{i_{k+1}}(y))\big)$.
Ainsi $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_{k+1}}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_{k+1}}(\epsilon) \end{bmatrix} \models SD(x, y)$.

Conclusion : On a, pour tout $k \in \{1, \dots, m\}$,

$$\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_k}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_k}(\epsilon) \end{bmatrix} \models SD(x, y).$$

Sans perte de généralité, on suppose que $u_{i_1}u_{i_2}\dots u_{i_m} = v_{i_1}v_{i_2}\dots v_{i_m}$ finit par un a. Autrement dit s'écrit $\tilde{u}a$. Comme on a $\mathcal{A}, \begin{bmatrix} x := u_{i_1}u_{i_2}\dots u_{i_m}(\epsilon) \\ y := v_{i_1}v_{i_2}\dots v_{i_m}(\epsilon) \end{bmatrix} \models SD(x, y)$, c'est-à-dire $\mathcal{A}, \begin{bmatrix} x := \tilde{u}a(\epsilon) \\ y := \tilde{u}a(\epsilon) \end{bmatrix} \models p(x, y)$, on a aussi $\mathcal{A}, \big[x := \tilde{u}(\epsilon)\big] \models SD(a(x), a(x))$. Donc $\mathcal{A}$ satisfait la conclusion de φ_I. Cela pour tout modèle $\mathcal{A}$, donc φ_I est valide. □

6.2 Équivalences classiques en logique du premier ordre

Définition 18 (Équivalence de formules) *Deux formules closes φ et ψ de $\mathcal{L}^1$ sont dites* équivalentes *(noté $\varphi \equiv \psi$) si pour toute $\mathcal{S}$-structure $\mathcal{A}$, on a $\mathcal{A} \models \varphi$ ssi $\mathcal{A} \models \psi$.*

- Loi de conversion des quantificateurs

$$\begin{array}{rcl} \neg\forall x\varphi & \equiv & \exists x\neg\varphi \\ \neg\exists x\varphi & \equiv & \forall x\neg\varphi \end{array}$$

- Loi de distribution des quantificateurs

$$\begin{array}{rcl} \forall x(\varphi\wedge\psi) & \equiv & (\forall x\varphi\wedge\forall x\psi) \\ \exists x(\varphi\vee\psi) & \equiv & (\exists x\varphi\vee\exists x\psi) \end{array}$$

- Lois de permutation des quantificateurs de même sorte

$$\begin{array}{rcl} \forall x\forall y\varphi & \equiv & \forall y\forall x\varphi \\ \exists x\exists y\varphi & \equiv & \exists y\exists x\varphi \end{array}$$

- Lois de réalphabétisation (renommage) des variables. On peut toujours renommer une variable liée au sein d'une formule. Cependant, le nouveau nom ne doit pas être un nom déjà utilisé pour une variable libre de la formule, sinon la sémantique n'est pas préservée.

  Exercice 7 *Dans* $\exists x(\forall xF(x,y)\to(G(x)\vee q))$*, on peut par exemple renommer l'occurrence* x *en* x' *et l'occurrence* x *en* x''*. Alors*

$$\exists x(\forall xF(x,y)\to(G(x)\vee q))\equiv\exists x'(\forall x''F(x'',y)\to(G(x')\vee q))$$

  Montrer qu'en renommant l'occurrence x *en* y *la sémantique n'est pas préservée.*

- Lois de passage : si $x\notin\texttt{vlibres}(\psi)$,

$$\begin{array}{rcl|rcl} \forall x(\varphi\wedge\psi) & \equiv & (\forall x\varphi)\wedge\psi & \forall x(\varphi\to\psi) & \equiv & (\exists x\varphi)\to\psi \\ \exists x(\varphi\wedge\psi) & \equiv & (\exists x\varphi)\wedge\psi & \exists x(\varphi\to\psi) & \equiv & (\forall x\varphi)\to\psi \\ \forall x(\varphi\vee\psi) & \equiv & (\forall x\varphi)\vee\psi & \forall x(\psi\to\varphi) & \equiv & \psi\to(\forall x\varphi) \\ \exists x(\varphi\vee\psi) & \equiv & (\exists x\varphi)\vee\psi & \exists x(\psi\to\varphi) & \equiv & \psi\to(\exists x\varphi) \end{array}$$

Exercice 8 *Utiliser la sémantique de* $\equiv$ *et celle des formules pour démontrer ces équivalences.*

6.3 Conséquence logique

Définition 19 *Une formule close* φ *est* conséquence logique *d'un ensemble de formules closes* Γ *si tout modèle de* Γ *est un modèle de* φ*. On écrit alors* $\Gamma\models\varphi$*.*

Une théorie *est l'ensemble des conséquences logiques d'un ensemble de formules closes.*

Exemple 18 *La* théorie des groupes *est l'ensemble des conséquences logiques de l'ensemble de formules suivantes, appelées* axiomes de la théorie des groupes *:*

Associativité	$\forall x\forall y\forall z[(x*y)*z=x*(y*z)]$	(Ass)
Élément neutre	$\forall x[(x*e=x)\wedge(e*x=x)]$	(EN)
Inverse	$\forall x[(x*i(x)=e)\wedge(i(x)*x=e)]$	(Inv)

Par exemple,

$$\{Ass,EN,Inv\}\models\forall x\forall y[(x*y=e)\to(y=i(x))]$$

Dans un groupe l'inverse à droite de tout élément est unique.

6.4 Ordre des quantificateurs dans une formule

On peut montrer que l'ordre dans lequel sont écrits certains quantificateurs est important : par exemple, les formules $\forall y \exists x P(x,y)$ et $\exists x \forall y P(x,y)$ ne sont pas équivalentes.

Autrement dit, on peut exhiber structure qui satisfait l'une et pas l'autre. On choisit la structure des entiers ordonnées $\mathcal{N} = (\mathbb{N}, \geq)$, dans laquelle P s'interprète comme $\geq$.

$\mathcal{N} \models \forall y \exists x P(x,y)$ ssi
pour tout $n \in \mathbb{N}$, $\mathcal{N}, [y \mapsto n] \models \exists x P(x,y)$
ssi pour tout $n \in \mathbb{N}$, il existe $m \in \mathbb{N}$, $\mathcal{N}, [y \mapsto n, x \mapsto m] \models P(x,y)$
ssi pour tout $n \in \mathbb{N}$, il existe $m \in \mathbb{N}$, $(m,n) \in P_{\mathcal{N}}$
c'est à dire pour tout $n \in \mathbb{N}$, il existe $m \in \mathbb{N}$ tel que $m \geq n$ (qui est vrai)

$\mathcal{N} \models \exists x \forall y P(x,y)$
ssi il existe $m \in \mathbb{N}$ t.q, $\mathcal{N}, [x \mapsto m] \models \forall y P(x,y)$
ssi il existe $m \in \mathbb{N}$, pour tout $n \in \mathbb{N}$, $\mathcal{N}, [x \mapsto m, y \mapsto n] \models P(x,y)$
ssi il existe $m \in \mathbb{N}$, pour tout $n \in \mathbb{N}$, $(m,n) \in P_{\mathcal{N}}$
ssi il existe $m \in \mathbb{N}$ tel que pour tout $n \in \mathbb{N}$, $m \geq n$

Ce dernier énoncé est évidemment faux car il n'existe pas d'entier plus grand que tous les autres.

On peut par contre établir que $\exists x \forall y P(x,y) \models \forall y \exists x P(x,y)$, ou de façon équivalente que $\models \exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$.

Nous aurons besoin de la notion de substitution dans les formules.

6.5 Substitution de variables dans les formules

Soit σ une substitution $\{x_1 \mapsto t_1, \ldots x_n \mapsto t_n\}$ (où les $x_i \in X$ et les $t_i \in \mathcal{T}(\mathcal{F}, X)$). On rappelle que $dom(\sigma) = \{x_1, \ldots, x_n\}$, et on note $Im(\sigma)$ l'ensemble des variables qui apparaissent dans les termes $t_1, \ldots, t_n$.

Définition 20 *Soit $\varphi \in \mathcal{L}^1$, on note $\varphi\sigma$ la formule φ dans laquelle toutes les occurrences libres de $x_1 \ldots, x_n$ ont été remplacées respectivement par $t_1, \ldots, t_n$.*

- $P(t'_1, \ldots, t'_n)\sigma = P(t'_1\sigma, \ldots, t'_n\sigma)$
- $(\neg\varphi)\sigma = \neg\varphi\sigma$
- $(\varphi \bowtie \psi)\sigma = \varphi\sigma \bowtie \psi\sigma$, *où* $\bowtie \in \{\vee, \wedge, \rightarrow\}$
- $(\exists x \varphi)\sigma = \begin{cases} \exists x(\varphi\rho) & \text{si } x \in Dom(\sigma) \text{ où } \rho = \sigma_{|Dom(\sigma)\setminus\{x\}} \\ \exists x(\varphi\sigma) & \text{si } x \notin Dom(\sigma) \text{ et } x \notin Im(\sigma) \\ \exists y(\varphi\{x \mapsto y\}\sigma) & \text{si } x \notin dom(\sigma) \text{ et } x \in Im(\sigma) \text{ et } y \notin \texttt{vlibres}(\varphi) \end{cases}$

Exemple 19 *On considère la substitution $\sigma = \{y \mapsto f(z)\}$. On a :*

- $(\exists x, p(x,y))\sigma := \exists x p(x, f(z))$;
- $(\exists y, p(x,y))\sigma := \exists y, p(x,y)$;
- $(\exists z, p(z,y))\sigma := \exists z', p(z', f(z))$.

Remarque 12 *Dans la Définition 20, il faut comprendre l'égalité comme la congruence induite par le renommage de variables liées, car par exemple pour $(\exists x \varphi)\sigma$ il faut choisir $y \notin \texttt{vlibres}(\varphi)$. Toutefois, cette définition est acceptable telle quelle car on peut établir que*

Proposition 6 *Si $y \notin \texttt{vlibres}(\varphi)$, alors $\exists x\varphi \equiv \exists y\varphi\{x \mapsto y\}$.*
Preuve □

Exercice 9 *Trouver d'autres bons exemples qui illustrent cette définition.*

Exercice 10 *Faire le lien avec la loi de réalphabétisation, en démontrant*

Lemme 7 (Lemme de substitution)

$$\mathcal{A}, \lambda \models \varphi\{x \mapsto t\} \text{ ssi } \mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models \varphi$$

Remarque 13 *Attention, il y a bien écrit "$\mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models \varphi$" et non pas "$\mathcal{A}, \lambda[x \mapsto t] \models \varphi$. En effet, il faut interpréter le terme t, ce dernier n'était pas un élément du domaine mais un objet syntaxique.*

Preuve Soit $\mathcal{P}r(\varphi)$ la propriété donné dans le lemme. Nous allons démontrer que $\mathcal{P}r(\varphi)$ est vraie pour toute formule φ, par induction structurelle sur φ.

Le cas de base porte sur une formule atomique $P(t_1, \ldots, t_n)$. En utilisant le lemme 3 (lemme de substitution sur les termes), on a :

$$\begin{aligned}
\mathcal{A}, \lambda \models P(t_1, \ldots, t_n)\{x \mapsto t\} \text{ ssi } & \mathcal{A}, \lambda \models P(t_1\{x \mapsto t\}, \ldots, t_n\{x \mapsto t\}) \\
& \text{ssi } ([\![t_1\{x \mapsto t\}]\!]_{\mathcal{A},\lambda}, \ldots, [\![t_n\{x \mapsto t\}]\!]_{\mathcal{A},\lambda}) \in P_{\mathcal{A}} \\
& \text{ssi } ([\![t_1]\!]_{\mathcal{A},\lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}]}, \ldots, [\![t_n]\!]_{\mathcal{A},\lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}]}) \in P_{\mathcal{A}} \\
& \text{ssi } \mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models P(t_1, \ldots, t_n)
\end{aligned}$$

d'où $\mathcal{P}r(P(t_1, \ldots, t_n))$.
Les cas de la négation, conjonction, disjonction et implication sont laissés en exercice.
Soit une formule φ telle que $\mathcal{P}r(\varphi)$. Montrons que $\mathcal{P}r(\exists y\varphi)$. Distinguons les trois cas de la définition 20.

- Pour le cas où y est le symbole x, (la substitution ρ est triviale) on a :

$$\begin{aligned}
\mathcal{A}, \lambda \models (\exists x\varphi)\{x \mapsto t\} \text{ ssi } & \mathcal{A}, \lambda \models \exists x\varphi \\
& \text{ssi } \mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models \exists x\varphi
\end{aligned}$$

- Si y différent de x et y n'apparait pas dans t,

$$\begin{aligned}
\mathcal{A}, \lambda \models (\exists y\varphi)\{x \mapsto t\} \text{ ssi } & \mathcal{A}, \lambda \models \exists y\varphi\{x \mapsto t\} \\
& \text{ssi il existe } d \in D_{\mathcal{A}} \text{ t.q. } \mathcal{A}, \lambda[y \mapsto d] \models \varphi\{x \mapsto t\} \\
& \text{ssi il existe } d \in D_{\mathcal{A}} \text{ t.q. } \mathcal{A}, \lambda[y \mapsto d][x \mapsto [\![t]\!]_{\mathcal{A},\lambda[y \mapsto d]}] \models \varphi \\
& \text{ssi il existe } d \in D_{\mathcal{A}} \text{ t.q. } \mathcal{A}, \lambda[y \mapsto d][x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models \varphi \\
& \text{ssi } \mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models \exists y\varphi
\end{aligned}$$

- Si y différent de x mais y apparaît dans le terme t, il faut renommer. La preuve du cas précédent ne fonctionne plus, et dès la première ligne.

□

Exercice 11 *Finir la démonstration du lemme de substitution.*

Chapter 7

Formes normales

Soit une formule φ du premier ordre.

1. *forme prénexe* : φ est équivalente à une formule qui n'a que des quantificateurs accumulés au début.

$$\varphi \equiv \mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \dots \mathcal{Q}_n x_n \psi(x_1, \dots, x_n)$$

2. *forme de Skolem* : elle est obtenue à partir d'une forme prénexe $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \dots \mathcal{Q}_n x_n \psi(x_1, \dots, x_n)$, en ne gardant que les quantifications universelles. Les variables quantifiées existentiellement sont vues comme des fonctionnelles des variables quantifiées universellement plus avant dans la forme prénexe. Elle est de la forme $\forall y_1 \forall y_2 \dots \forall y_m \psi'(y_1, \dots, y_m)$.

   On peut montrer que $\forall y_1 \forall y_2 \dots \forall y_m \psi'(y_1, \dots, y_m)$ est satisfaisable ssi $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \dots \mathcal{Q}_n x_n \psi(x_1, \dots, x_n)$ est satisfaisable.

   En général, elles ne sont pas équivalentes.

3. *forme clausale* : elle est obtenue à partir d'une forme de Skolem $\forall y_1 \forall y_2 \dots \forall x_m \psi'(y_1, \dots, y_m)$ en ne gardant que $\psi'(y_1, \dots, y_m)$ que l'on met en forme normale confonctive pour obtenir $\psi''(y_1, \dots, y_m)$. On peut alors appliquer une méthode de résolution, dans le même esprit que celle du calcul propositionnel.

7.1 Formes prénexes

Définition 21 (Forme prénexe) *Une formule est* en forme prénexe *lorsqu'elle est de la forme*

$$\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \dots \mathcal{Q}_n x_n \psi,$$

où chaque $\mathcal{Q}_i$ est le symbole $\exists$ ou $\forall$ et ψ ne contient aucun quantificateur. La partie $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \dots \mathcal{Q}_n$ est appelée préfixe *et ψ est la* matrice.

Exemple 20 *Les formules $\exists x \forall y P(x, y)$ et $\forall x \exists y P(x, y)$ sont en forme prénexe.*

Exemple 21 *Par contre, la formule $\exists x((P(x, y) \to \forall y P(y, x))$ n'est pas en forme prénexe.*

Proposition 8 *Toute formule admet une forme prénexe équivalente.*

Preuve On considère l'algorithme suivant :

1. Renommer les variables de façon à ce qu'aucune variable n'ait à la fois une occurrence libre et une occurrence liée, ni d'occurrence liée à des quantificateurs différents.

2. Appliquer tant que possible les lois de conversion des quantificateurs et les lois de passage.

Ces transformations successives préservent l'équivalence des formules. □

Exercice 12 *Finir l'argument de la preuve de la Proposition 8.*

Exemple 22 $\exists x \forall y P(x,y) \to \forall x \exists y P(y,x) \equiv \exists y' \forall x' \exists x \forall y P(x,y) \to P(y',x')$.

Rappel des lois de passage : si $x \notin \texttt{vlibres}(\psi)$,

$$\begin{array}{lcllcl} \forall x(\varphi \wedge \psi) & \equiv & (\forall x\varphi) \wedge \psi & \forall x(\varphi \to \psi) & \equiv & (\exists x\varphi) \to \psi \\ \exists x(\varphi \wedge \psi) & \equiv & (\exists x\varphi) \wedge \psi & \exists x(\varphi \to \psi) & \equiv & (\forall x\varphi) \to \psi \\ \forall x(\varphi \vee \psi) & \equiv & (\forall x\varphi) \vee \psi & \forall x(\psi \to \varphi) & \equiv & \psi \to (\forall x\varphi) \\ \exists x(\varphi \vee \psi) & \equiv & (\exists x\varphi) \vee \psi & \exists x(\psi \to \varphi) & \equiv & \psi \to (\exists x\varphi) \end{array}$$

7.2 Formes de Skolem

Dans cette section on se fixe une signature $\mathcal{S}$.

Soit $\varphi = \exists x\psi$ close. On rappelle que $\mathcal{A} \models \varphi$ ssi il existe $d \in D_\mathcal{A}$ tel que $\mathcal{A}, [x \mapsto d] \models \psi$.

Considérons maintenant la signature $\mathcal{S}' = \mathcal{S} \cup \{c(0)\}$ où c est un nouveau symbole de constante, *i.e.,* $c \notin \mathcal{F}$. Soit $\mathcal{S}'$-structure $\mathcal{A}'$ qui est comme $\mathcal{A}$ avec en plus $c_{\mathcal{A}'} = d$. Clairement $\mathcal{A}' \models \psi\{x \mapsto c\}$.

Ainsi, $\exists x\psi$ a un $(\mathcal{F}, \mathcal{P})$-modèle ssi $\psi\{x \mapsto c\}$ a un $(\mathcal{F} \cup \{c\}, \mathcal{P})$-modèle, où $c \notin \mathcal{F}$.

Soit maintenant la formule $\varphi = \forall y \exists x \psi$ sur une signature $\mathcal{S}$. Une $\mathcal{S}$-structure $\mathcal{A}$ est un modele de φ ssi pour chaque $d \in D_\mathcal{A}$ il existe $d' \in D_\mathcal{A}$ tel que $\mathcal{A}, [y \mapsto d, x \mapsto d'] \models \psi$.

On considère une fonction $m : D_\mathcal{A} \to D_\mathcal{A}$ qui à chaque $d \in D_\mathcal{A}$ associe l'un des éléments $d' \in D_\mathcal{A}$, de sorte que $\mathcal{A}, [y \mapsto d, x \mapsto d'] \models \psi$.

Alors en enrichissant la signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ en $\mathcal{S}' = (\mathcal{F} \cup \{f(1)\}, \mathcal{P})$ avec où $f(1) \notin \mathcal{F}$ et en prenant la structure étendue $\mathcal{A}'$ où $f^{\mathcal{A}'} = m$, on a

$$\mathcal{A}' \models \psi\{x \mapsto f(y)\}$$

Donc φ admet un modèle ssi $\forall y \psi\{x \mapsto f(y)\}$ admet un modèle.

On généralise ce principe pour mettre une formule φ *en forme de Skolem* et obtenir une formule $Sk(\varphi)$ sans plus aucun quantificateur existentiel.

On suppose se donne pour chaque variable $x \in X$ et chaque formule $\varphi \in \mathcal{L}^1$ un nouveau symbole de fonction $f_{x,\varphi} \notin \mathcal{F}$, et on définit la nouvelle signature

$$\mathcal{S}_{sk} := (\mathcal{F}_{sk}, \mathcal{P})$$

où $\mathcal{F}_{sk} := \mathcal{F} \cup \{f_{x,\varphi} \mid x \in X, \varphi \in \mathcal{L}^1\}$.

Définition 22 *Les symboles de fonctions* $f_{x,\varphi}$ *s'appellent des symboles de fonctions de Skolem.*

Définition 23 (forme de Skolem) *Soit* φ *une formule prénexe. La forme de Skolem de* φ*, notée* $Sk(\varphi)$*, est la formule définie par induction sur* φ*:*

- *si* φ *est sans quantificateur,* $Sk(\varphi) = \varphi$

- $Sk(\forall x\varphi) = \forall x Sk(\varphi)$
- $Sk(\exists x\varphi) = Sk(\varphi)\{x \mapsto f_{x,\varphi}(\vec{y}_{x,\varphi})\}$ *où* $\vec{y}_{x,\varphi}$ *est une énumération des variables libres de la formule* $\exists x\varphi$*, et où* $f_{x,\varphi}$ *est un symbole de fonction frais.*

Exemple 23 *La définition de la skolémisation à* $\exists x\forall y\forall x'\exists y' p(x,y) \wedge p(y',x')$ *:*

$$\begin{array}{l} Sk(\exists x\forall y\forall x'\exists y' p(x,y) \wedge p(y',x')) \\ = Sk(\forall y\forall x'\exists y' p(x,y) \wedge p(y',x'))\{x \mapsto c\} \\ = \forall y\forall x' Sk(\exists y' p(x,y) \wedge p(y',x'))\{x \mapsto c\} \\ = \forall y\forall x' Sk(p(x,y) \wedge p(y',x'))\{y' \mapsto f(x,y,x')\}\{x \mapsto c\} \\ = \forall y\forall x' p(c,y) \wedge p(f(c,y,x'),x') \end{array}$$

Exemple 24

$$\begin{array}{l} Sk(\exists x_1\forall x_2\forall x_3\exists x_4\forall x_5\exists x_6 p(x1,x2,x3,x4,x5,x6)) \\ = Sk(\forall x_2\forall x_3\exists x_4\forall x_5\exists x_6 p(x1,x2,x3,x4,x5,x6))\{x \mapsto c\} \\ = \forall x_2\forall x_3 Sk(\exists x_4\forall x_5\exists x_6 p(x1,x2,x3,x4,x5,x6))\{x \mapsto c\} \\ = \forall x_2\forall x_3 Sk(\forall x_5\exists x_6 p(x1,x2,x3,x_4,x5,x6))\{x_4 \mapsto f(x_1,x_2,x_3,x_5)\}\{x \mapsto c\} \\ = \forall x_2\forall x_3\forall x_5 Sk(\exists x_6 p(x1,x2,x3,x_4,x5,x6))\{x_4 \mapsto f(x_1,x_2,x_3,x_5)\}\{x \mapsto c\} \\ = \forall x_2\forall x_3\forall x_5 Sk(p(x1,x2,x3,x_4,x5,x6))\{x_6 \mapsto g(x_1,x_2,x_3,x_5)\}\{x_4 \mapsto f(x_1,x_2,x_3)\}\{x \mapsto c\} \\ = \forall x_2\forall x_3\forall x_5 p(x1,x2,x3,x_4,x5,x6)\{x_6 \mapsto g(x_1,x_2,x_3,x_4,x_5)\}\{x_4 \mapsto f(x_1,x_2,x_3)\}\{x \mapsto c\} \\ = \forall x_2\forall x_3\forall x_5 p(x1,x2,x3,x_4,x5,g(c,x_2,x_3,f(c,x_2,x_3),x_5,g(c,x_2,x_3,f(c,x_2,x_3),x_5)) \end{array}$$

Théorème 9 *Soit* φ *une formule prénexe close. On a :* $Sk(\varphi)$ *est satisfaisable ssi* φ *est satisfaisable.*

Preuve Nous allons montrer une propriété plus riche : toute formule φ est satisfiable dans une certaine structure ssi on peut enrichir cette structure avec des interprétations des fonctions de Skolem pour satisfaire $Sk(\varphi)$. Plus formellement, étant donnée une structure $\mathcal{A} = (D, \{f^{\mathcal{A}}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}}\}_{P\in\mathcal{P}})$ on dira qu'une structure $\mathcal{A}' = D', \{f^{\mathcal{A}'}\}_{f\in\mathcal{F}'}, \{P^{\mathcal{A}'}\}_{P\in\mathcal{P}})$ étend $\mathcal{A}$ si

- $\mathcal{F} \subseteq \mathcal{F}'$,
- $D = D'$,
- $f^{\mathcal{A}'} = f^{\mathcal{A}}$ pour tout symbole de fonctions $f \in \mathcal{F}$
- et $P^{\mathcal{A}'} = P^{\mathcal{A}}$ pour tout symbole de prédicat $P \in \mathcal{P}$.

Autrement dit, $\mathcal{A}'$ étend $\mathcal{A}$ si on a seulement ajouté des interprétations pour de nouveaux symboles de fonctions. Nous introduisons la propriété $\mathcal{P}r(\varphi)$ pour toute formule prénexe (pas forcément close) :

en notant $\mathcal{S}$ la signature de la formule φ, pour toute $\mathcal{S}$-structure $\mathcal{A}$, on a :
[pour toute affectation λ sur les variables libres de φ, $\mathcal{A}, \lambda \models \varphi$] ssi
[il existe une structure $\mathcal{A}'$ qui étend $\mathcal{A}$ avec pour toute affectation λ, $\mathcal{A}', \lambda \models Sk(\varphi)$].

$\boxed{\varphi \text{ sans quantificateur}}$ Nous avons :

$$\begin{array}{ll} \mathcal{A}, \lambda \models \varphi \text{ ssi } \mathcal{A}, \lambda \models Sk(\varphi) & \text{car } Sk(\varphi) = \varphi \\ \quad \text{ssi il existe } \mathcal{A}' \text{ étendant } \mathcal{A} \text{ avec } \mathcal{A}', \lambda \models Sk(\varphi) & \end{array}$$

car $Sk(\varphi)$ ne parle pas des nouveaux symboles de fonction qui sont interprétées dans $\mathcal{A}'$. D'où $\mathcal{P}r(\varphi)$.

$\boxed{\mathcal{P}r(\forall x\varphi)}$ Supposons $\mathcal{P}r(\varphi)$ et montrons $\mathcal{P}r(\forall x\varphi)$.

pour tout λ sur les variables libres de $\forall x\varphi$ $\mathcal{A}, \lambda \models \forall x\varphi$
ssi pour tout λ sur les variables libres de $\forall x\varphi$, pour tout $d \in D, (\mathcal{A}, \lambda[x := d] \models \varphi)$
ssi pour tout λ sur les variables libres de $\varphi, (\mathcal{A}, \lambda \models \varphi)$
ssi il existe $\mathcal{A}'$ étendant $\mathcal{A}$ avec pour tout λ sur les variables libres de $\varphi, \mathcal{A}', \lambda \models Sk(\varphi)$
ssi il existe $\mathcal{A}'$ étendant $\mathcal{A}$ avec pour tout λ sur les variables libres de $\forall x\varphi, \mathcal{A}', \lambda \models \forall x Sk(\varphi)$.

$\boxed{\mathcal{P}r(\exists x\varphi)}$ Supposons $\mathcal{P}r(\varphi)$. Montrons $\mathcal{P}r(\exists x\varphi)$.

$$\begin{aligned}
\text{pour tout } \lambda, \mathcal{A}', \lambda \models Sk(\exists x\varphi) \text{ ssi } & \text{ pour tout } \lambda, \mathcal{A}', \lambda \models Sk(\varphi)\{x \mapsto f_{x,\varphi}(\vec{y}_{x,\varphi})\} \\
\text{ssi } & \text{ pour tout } \lambda, \mathcal{A}', \lambda[x \mapsto [\![f_{x,\varphi}(\vec{y}_{x,\varphi})]\!]_{\mathcal{A}',\lambda}] \models Sk(\varphi) \\
\text{ssi } & \text{ pour tout } \lambda, \mathcal{A}, \lambda[x \mapsto [\![f_{x,\varphi}(\vec{y}_{x,\varphi})]\!]_{\mathcal{A}',\lambda}] \models \varphi \\
\text{implique } & \text{ pour tout } \lambda, \text{ il existe } d \in D, \mathcal{A}, \lambda \models \varphi \\
\text{implique } & \text{ pour tout } \lambda, \mathcal{A}, \lambda \models \exists x\varphi
\end{aligned}$$

Réciproquement, supposons pour tout $\lambda, \in D, \mathcal{A}, \lambda \models \exists x\varphi$. Ainsi, pour tout λ, il existe $d \in D$ tel que $\mathcal{A}, \lambda[x \mapsto d] \models \varphi$. On rappelle que λ est définie sur l'ensembles des variables libres $\vec{y}_{x,\varphi}$ de φ. Par l'axiome du choix, il existe une fonction $choix : D^{\vec{y}_{x,\varphi}} \to D$, qui donne un tel d pour chaque affectation λ. On étend alors la structure $\mathcal{A}$ en une structure $\mathcal{A}'$ en interprétant $f_{x,\varphi}$ par la fonction $choix$. On a alors :

$$\begin{aligned}
\text{pour tout } \lambda, \mathcal{A}, \lambda \models \exists x\varphi \text{ implique } & \text{ pour tout } \lambda, \text{ il existe } d \in D, \mathcal{A}, \lambda[x \mapsto d] \models \varphi \\
\text{implique } & \text{ pour tout } \lambda, \text{ il existe } d \in D, \mathcal{A}, \lambda[x \mapsto d] \models Sk(\varphi) \\
\text{implique } & \text{ pour tout } \lambda, \mathcal{A}', \lambda[x \mapsto choix(\lambda(\vec{y}_{x,\varphi})] \models Sk(\varphi) \\
\text{implique } & \text{ pour tout } \lambda, \mathcal{A}', \lambda[x \mapsto [\![f_{x,\varphi}(\vec{y}_{x,\varphi})]\!]_{],\models} Sk(\varphi) \\
\text{implique } & \text{ pour tout } \lambda, \mathcal{A}', \lambda \models Sk(\varphi)[x \mapsto f_{x,\varphi}(\vec{y}_{x,\varphi})] \\
\text{implique } & \text{ pour tout } \lambda, \mathcal{A}', \lambda \models Sk(\exists x\varphi)
\end{aligned}$$

□

7.3 Forme clausale

Définition 24 *Une formule est* en forme clausale *si elle est en forme standard de Skolem* $\forall x_1 \forall x_2 \ldots \forall x_n \psi(x_1, \ldots, x_n)$ *et si sa matrice* $\psi(x_1, \ldots, x_n)$ *est en forme normale conjonctive.*

Une formule en forme clausale $\forall x_1 \forall x_2 \ldots \forall x_n(\psi_1 \wedge \psi_2 \wedge \ldots \psi_m)$ peut être vue comme l'ensemble de clauses $\{\psi_1, \psi_2, \ldots, \psi_m\}$, puisqu'on a

$$\forall x(\psi_1 \wedge \psi_2 \wedge \ldots \psi_m) \equiv (\forall x\psi_1) \wedge (\forall x\psi_2) \wedge \ldots \wedge (\forall x\psi_m)$$

Exemple 25 *On considère la formule* $\exists x \forall y \forall x' \exists y'[p(x, y) \wedge p(y', x')]$ *de l'Exemple 23 pour laquelle on a obtenu la forme skolémisée*

$$\forall y \forall x' p(c, y) \wedge p(f(c, y, x'), x')]$$

À cette dernière on associe l'ensemble de clauses

$$\{\neg p(c, y), p(f(c, y, x'), x')\}.$$

Exemple 26 *On reprend la formule* $\exists x_1 \forall x_2 \forall x_3 \exists x_4 \exists x_6 p(x_1, x_2, x_3, x_4, x_5, x_6)$ *de l'Exemple 24 pour laquelle on a obtenu la forme skolémisée*

$$\forall x_2 \forall x_3 \forall x_5 p(x_1, x_2, x_3, x_4, x_5, g(c, x_2, x_3, f(c, x_2, x_3), x_5, g(c, x_2, x_3, f(c, x_2, x_3), x_5))$$

À cette dernière on associe l'ensemble de clauses singleton

$$\{p(x_1, x_2, x_3, x_4, x_5, g(c, x_2, x_3, f(c, x_2, x_3), x_5, g(c, x_2, x_3, f(c, x_2, x_3), x_5))\}.$$

Théorème 10 *Soit* $\mathcal{C}$ *un ensemble de clauses résultant de la mise en forme clausale d'une formule* φ*. Alors* φ *est insatisfiable ssi* $\mathcal{C}$ *est insatisfiable.*

Remarque 14 *Ce théorème forme la base de nombreux démonstrateurs automatiques utilisant une représentation des formules en forme de clauses. Il établit que la recherche de l'insatisfiabilité d'une formule* φ *est équivalente à la recherche d'insatisfiabilité de sa représentation en forme clausale* $\mathcal{C}$*.*

Cependant φ et $\mathcal{C}$ ne sont pas logiquement équivalentes *puisque les formules de* $\mathcal{C}$ *sont définies sur une signature étendue (dûe au procédé de Skolémisation).*

7.4 Théorème de Herbrand

Nous mettons en exergue une famille de structures qui repose sur les algèbres de termes clos et qui suffiront à déterminer l'insatisfaisabilité (ou de façon équivalente la validité) d'une formule du premier ordre.

On suppose que la signature $\mathcal{S} = \{\mathcal{F}, \mathcal{P}\}$ contient au moins un symbole de constante.

Exercice 13 *Pourquoi ?*

Définition 25 (Structure de Herbrand) *Une* structure (ou modèle) de Herbrand *est une* $\mathcal{S}$*-structure* $\mathcal{H}$ *dont la* $\mathcal{F}$*-algèbre est* $\mathcal{T}(\mathcal{F})$ *celle des termes clos.*

Par conséquent il y a une structure de Herbrand pour chaque façon d'interpréter les symboles de relations, et donc de choisir pour chaque $R \in \mathcal{P}_n$ quels n-uplet $(t_1, \ldots, t_n)$ sont dans $R^{\mathcal{H}}$. En posant l'ensemble des formules atomiques closes, appelé *base de Herbrand*,

$$\texttt{AtomesClos}(\mathcal{F}, \mathcal{P}) := \{R(t_1, \ldots, t_n) \,|\, R \in \mathcal{P}_n \text{ et } t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F})\}.$$

Une structure de Herbrand est donc complétement déterminée par un sous-ensemble A de $\texttt{AtomesClos}(\mathcal{F}, \mathcal{P})$ en posant

$$R^{\mathcal{H}} := \{(t_1, \ldots, t_n) \,|\, R(t_1, \ldots, t_n) \in A\}.$$

Exemple 27

Théorème 11 (de Herbrand) *Un ensemble de clauses est satisfaisable ssi il admet un modèle de Herbrand.*

Preuve Soit $\mathcal{C}$ un ensemble de clauses.

Le sens ($\Leftarrow$) est évident.

Supposons $\mathcal{C}$ satisfaisable dans une structure $\mathcal{A} = (D, \{f^{\mathcal{A}}\}_{f \in \mathcal{F}}, \{P^{\mathcal{A}}\}_{P \in \mathcal{P}})$.

On considère le modèle de Herbrand $\mathcal{H}_{\mathcal{A}}$ défini par l'ensemble

$$\{R(t_1, \ldots, t_n) \,|\, \in \texttt{AtomesClos}(\mathcal{F}, \mathcal{P}) \,|\, \mathcal{A} \models R(t_1, \ldots, t_n)\}.$$

On établit que pour toute clause $C \in \mathcal{C}$, et toute substitution (close) $\sigma : X \to \mathcal{T}(\mathcal{F})$, c'est aussi une $\mathcal{H}$-affectation, on a

$$\mathcal{A} \models C\sigma \text{ ssi } \mathcal{H}, \sigma \models C$$

On raisonne par induction sur C.

- Si $C = R(t_1, \ldots, t_n)$ alors $C\sigma = R(, \ldots,)$, donc
  $\mathcal{A} \models C\sigma$
  ssi $\mathcal{A} \models R(t_1\sigma, \ldots, t_n\sigma)$
  ssi $(t_1\sigma, \ldots, t_n\sigma) \in R^{\mathcal{H}}$
  ssi $\mathcal{H}, \sigma \models R(t_1, \ldots, t_n)$
- pour $C = \neg R(t_1, \ldots, t_n)$ on utilise le même raisonnement.
- le cas $C = C_1 \vee C_2$ est évident par induction.

□

7.5 Théorème de Herbrand et ses conséquences

Grâce au Théorème de Herbrand, on va pouvoir établir un lien entre formules du premier ordre et ensemble de formules du calcul propositionnel. On supoose que la signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ contient au moins un symbole de constante.

À une clause C on associe l'ensemble $H(C)$ de ses *instances de Herbrand* défini par

$$H(C) := \{C\sigma \,|\, \sigma : X \to \mathcal{T}(\mathcal{F})\}$$

et pour un ensemble de clauses $\mathcal{C}$ on pose $H(\mathcal{C}) := \bigcup_{C \in \mathcal{C}} H(C)$.

Exemple 28 *On considère la signature $\mathcal{S} = \{\{a(0), b(0), f(1)\, \{P(1), Q(2)\}\}$ et on considère l'ensemble de clause $\mathcal{C} - \{P(x) \vee Q(f(x), y), Q(x, x)\}$. Alors $H(\mathcal{C}) = \{P(a) \vee Q(f(a), a), P(a) \vee Q(f(a), b), P(b) \vee Q(f(b), a), P(b) \vee Q(f(b), b), P(f(a)) \vee Q(f(f(a)), a), P(f(a)) \vee Q(f(f(a)), b), \ldots, Q(a, a), Q(b, b), Q(f(a), f(a)), \ldots\}$*

Théorème 12 *Soit $\mathcal{C}$ un ensemble de clauses sur $\mathcal{S} = (\mathcal{F}, \mathcal{P})$. Alors, $\mathcal{C}$ est satisfaisable ssi l'ensemble $H(\mathcal{C})$ vu comme un ensemble de clauses propositionnelles sur l'ensemble de propositions $\texttt{AtomesClos}(\mathcal{F}, \mathcal{P})$ est satisfaisable.*

Preuve On pourra montrer que pour toute clause C dont l'ensemble fini de variables est $\{x_1, \ldots, x_k\}$, on a

$$\mathcal{H} \models \forall x_1 \ldots \forall x_k C \text{ ssi } \mathcal{H} \models H(C)$$

□

Exemple 29

Théorème 13 (Théorème de compacité de la logique des prédicats) *Un ensemble de formules du premier ordre est satisfaisables si, et seulement si, chacun de ses sous-ensembles finis est satisfaisable.*

Preuve On profite l'équisatisfaisabilté entre les formules et leur mise en forme clausale, du théorème de Herbrand et de la compacité du calcul propositionnel. □

Théorème 14 (Théorème de Lownheim-Skolem descendant) *Soit une signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$ dénombrable, et E un ensemble de formules closes sur $\mathcal{S}$. Alors E a un modèle si, et seulement si, E a un modèle dénombrable.*

Preuve Supposons que l'ensemble E a un modèle On peut skolémiser les formules de E en augmentant l'ensemble $\mathcal{F}$ en $\mathcal{F}_{sk}$. Soit $Sk(E)$ l'ensemble de formules obtenues. Remarquons que dans le procédé de skolémisation si $\mathcal{F}$ est dénombrable alors $\mathcal{F}_{sk}$ l'est aussi.

Comme E a un modèle ssi $Sk(E)$ a un modèle, d'après le Théorème de Herbrand $Sk(E)$ a un modèle de Herbrand $\mathcal{H}$ de domaine $\mathcal{T}(\mathcal{F}_{sk})$, qui est dénombrable. On considère la restriction $\mathcal{H}_{|\mathcal{F}}$ de la structure $\mathcal{H}$ aux symboles de fonctions de $\mathcal{F}$. Alors $\mathcal{H}_{|\mathcal{F}}$ est un modèle de E, et il est clairement dénombrable. □

Chapter 8

Systèmes de preuve

On peut se munir d'un nombre fini de principes de déduction, appelé *système de preuve* qui consiste en la donnée de :

- un ensemble d'*axiomes*, et
- des *règles*

qui, étant donnée une théorie (*i.e.,*), permettent de déduire

Il existe différents systèmes de preuves, parmi les plus standards on trouve :

- les *systèmes axiomatiques* (ou *systèmes de Hilbert*);
- la *déduction naturelle*;
- le *calcul des séquents*.

Nous verrons brièvement leur principe.

Il existe aussi un système de preuve pour une approche par *résolution*, analogue à celle considérée pour le calcul des propositions et qui requiert des formules en forme clausales que nous étudierons en détail en Section 8.3.

Tous ces systèmes ont la propriété d'être ;

- *corrects* au sens ou tout ce qu'on infère est vrai, et
- *complets* au sens où toute vérité admet une preuve dans le système de preuve considéré.

C'est en ce sens qu'on énonce :

Théorème 15 (Complétude de la logique du premier ordre) *Le calcul des prédicats est* complet*, c-à-d. qu'il existe des systèmes de preuve complets.*

8.1 Les systèmes de preuve (par preuve directe)

8.1.1 Les systèmes axiomatiques

Cette partie est à complétée en donnant la définition :

- des *systèmes axiomatiques* (ou *systèmes de Hilbert*);

- de la *déduction naturelle*;
- lu *calcul des séquents.*

avec des exemples de preuves dans certaines théories.

8.2 Déduction naturelle

La *déduction naturelle* est un système de preuve qui formalise les preuves rédigées par les mathématiciens, mais au prix d'un plus grand nombre de règles. On considère des jugements notés $\Gamma \vdash \phi$ où Γ est un ensemble fini de formules et ϕ est une formule. Un tel jugement se lit "en supposant Γ, j'ai démontré ϕ".

$$\text{axiome}\frac{}{\Gamma, A \vdash A} \qquad \text{affaiblissement}\frac{\Gamma \vdash A}{\Gamma, B \vdash A} \qquad \text{absurde}\frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}$$

	Règles d'introduction	Règles d'élimination
$\to$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash (A \to B)}$	$\dfrac{\Gamma \vdash A \quad \Gamma \vdash (A \to B)}{\Gamma \vdash B}$
$\wedge$	$\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash (A \wedge B)}$	$\dfrac{\Gamma \vdash (A \wedge B)}{\Gamma \vdash A}$ $\dfrac{\Gamma \vdash (A \wedge B)}{\Gamma \vdash B}$
$\vee$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash (A \vee B)}$ $\dfrac{\Gamma \vdash B}{\Gamma \vdash (A \vee B)}$	$\dfrac{\Gamma \vdash (A \vee B) \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$
$\neg$	$\dfrac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}$	$\dfrac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash \bot}$
$\exists$	$\dfrac{\Gamma \vdash A[x := t]}{\Gamma \vdash \exists x A}$	$\dfrac{\Gamma \vdash \exists x A \quad \Gamma, A \vdash C}{\Gamma \vdash C}$ où x non libre dans Γ, C
$\forall$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash \forall x A}$ où x non libre dans Γ	$\dfrac{\Gamma \vdash \forall x A(x)}{\Gamma \vdash A[x := t]}$
$=$	$\dfrac{}{\Gamma \vdash (t = t)}$	$\dfrac{\Gamma \vdash \phi(x := t) \quad \Gamma \vdash (t = u)}{\Gamma \vdash \phi(x := u)}$

Définition 26 *Soit Γ un ensemble fini de formules et φ une formule. Une preuve en déduction naturelle de $\Gamma \vdash \varphi$ est une suite finie de jugements telle que :*

- *tout jugement est obtenu à partir de l'application d'une règles sur des jugements précédents ;*
- *le dernier jugement est $\Gamma \vdash \varphi$.*

Exemple 30 *Voir Pravda.*

Notations 27 *Soit E un ensemble (potentiellement infini) de formules. On écrit $E \vdash_{DN} \varphi$ pour dire qu'il existe une preuve en déduction naturelle de $\Gamma \vdash \varphi$ pour Γ fini et inclus dans E.*

Théorème 16 (Correction de la déduction naturelle) *Si $E \vdash_{DN} \varphi$ alors $E \models \varphi$.*

Preuve Supposons $E \vdash_{DN} \varphi$. Il existe donc une preuve de $\Gamma_0 \vdash \varphi$ en déduction naturelle. Soit ℓ la longueur de cette preuve, i.e. la longueur de la suite finie. Pour tout $n \in \{1, \ldots, \ell\}$, on pose la propriété $\mathcal{P}r(n)$:

"si $\Gamma_n \vdash \varphi_n$ est le n-ème élément de la suite, alors $\Gamma_n \models \varphi_n$."

On démontre que pour tout $n \in \{1, \ldots, \ell\}$, la propriété $\mathcal{P}r(n)$ est vraie. Pour cela, on procède par récurrence forte sur n. Soit $n \in \{1, \ldots, \ell\}$ telle que $\mathcal{P}r(i)$ pour tout $i < n$. Montrons $\mathcal{P}r(n)$. Le reste de la démonstration est une étude de cas sur la règle appliquée pour obtenir le jugement $\Gamma_n \vdash \varphi_n$.

- Cas de l'introduction du $\exists$.

  Dans ce cas, le jugement $\Gamma_n \vdash \varphi_n$ est de la forme $\Gamma \vdash \exists x A$ et il existe un jugement qui le précède, à la position $i < n$, dans la preuve de la forme $\Gamma \vdash A[x := t]$.

  Par $\mathcal{P}r(i)$, on a $\Gamma \models A[x := t]$. Soit $\mathcal{A}$ une structure et λ une affectation telle que $\mathcal{A}, \lambda \models \Gamma$. Nous avons $\mathcal{A}, \lambda \models A[x := t]$. Mais alors par le lemme 7, $\mathcal{A}, \lambda[x \mapsto [\![t]\!]_{\mathcal{A},\lambda}] \models A$. Donc, $\mathcal{A}, \lambda \models \exists x A$.

- Cas de l'introduction du $\forall$.

  Cela revient à supposer $\Gamma \models A$ avec x non libre dans Γ, et à montrer que $\Gamma \models \forall x A$. Soit $\mathcal{A}$ une structure et λ une affectation telles que $\mathcal{A}, \lambda \models \Gamma$. Comme x est non libre dans Γ, on a pour tout $d \in D_{\mathcal{A}}$, $\mathcal{A}, \lambda[x \mapsto d] \models \Gamma$. Par hypothèse (de récurrence $\mathcal{P}r(i)$ où i est l'indice du jugement $\Gamma \vdash A$), on a $\mathcal{A}, \lambda[x \mapsto d] \models A$, et ce pour tout $d \in D_{\mathcal{A}}$. Ainsi, $\mathcal{A}, \lambda \models \forall x A$.

□

Exercice 14 *Finir la démonstration de la correction.*

Théorème 17 (Complétude de la déduction naturelle) *Si $E \models \varphi$ alors $E \vdash_{DN} \varphi$.*

Preuve Admis pour l'instant. □

8.3 La résolution : un système de preuve par réfutation

Le théorème de Herbrand nous permet de corréler calcul des propositions et logique du premier ordre.

Le principe du système de preuve par résolution pour la logique du premier ordre exploite le fait que l'ensemble des instances d'une clause C est un ensemble très régulier : cet ensemble $H(C)$ d'instances de Herbrand est d'ailleurs représenté par la clause C elle-même.

Lorsqu'on considère un ensemble de clauses $\mathcal{C}$ contradictoire, on sait qu'il existe une preuve de cette contradiction en appliquant la résolution à l'ensemble de clauses propositionnelles de $H(\mathcal{C})$ (Théorème 12). L'idée est de simuler la preuve de contradiction dans $H(\mathcal{C})$ par une preuve dans $\mathcal{C}$: pour deux clauses C et C' on exécute en une seule étape un pas de résolution entre toutes les instances de Herbrand de C et toutes les instances de Herbrand C'.

8.3.1 Règles

On considère donc le système de règles de la Figure 8.1 qui, comme pour le cas propositionnel, est composé d'une règle de coupure et d'une règle de factorisation, dans lesquelles on a fait apparaître explicitement les variables de chacune des clauses sous la forme $\forall \bar{x}$, $\forall \bar{y}$, et $\forall \bar{z}$, pour plus de clarté.

Dans la règle de coupure on suppose que $\bar{x} \cap \bar{y} = \emptyset$ ce qui ne perd pas de généralités quitte à renommer les variables de la clause $\neg P(\bar{t}) \vee C'$ par exemple.

$$(\textbf{Coupure})\frac{\forall \bar{x}(P(\bar{s}) \vee C) \qquad \forall \bar{y}(\neg P(\bar{t}) \vee C')}{\forall \bar{z}(C \vee C')\sigma} \quad \begin{cases} \bar{x} = Var(P(\bar{s}) \vee C) \\ \bar{y} = Var(P(\bar{t}) \vee C') \\ \bar{x} \cap \bar{y} = \emptyset \\ \bar{z} = Var((C \vee C')\sigma) \\ \sigma \text{ est un u.p. de } \bar{s} \stackrel{?}{=} \bar{t} \end{cases}$$

$$(\textbf{Factorisation})\frac{\forall \bar{x}(P(\bar{s}) \vee P(\bar{t}) \vee C)}{\forall \bar{z}(P(\bar{s}) \vee C)\sigma} \quad \begin{cases} \bar{x} = Var(P(\bar{s}) \vee P(\bar{t}) \vee C) \\ \bar{z} = Var((C \vee C')\sigma) \\ \sigma \text{ est un unif. princ. de } \bar{s} \stackrel{?}{=} \bar{t} \end{cases}$$

Figure 8.1: Règles de résolution en logique du premier ordre

Définition 28 *Soit $\mathcal{C}$ un ensemble de clauses et C une clause.*

On note $\mathcal{C} \vdash_{\mathbf{R1}} C$ si on peut dériver la clause C à partir de l'ensemble de clauses $\mathcal{C}$ en utilisant une des règles du système de résolution données en Figure 8.1.

On note $\mathcal{C} \vdash^*_{\mathbf{R1}} C$ si Cpeut être dérivée à partir de $\mathcal{C}$ en utilisant zéro ou plusieurs règles du système de résolution données en Figure 8.1.

Exercice 15 *Montrer que*

$$\{P(x,x), \neg P(x,y) \vee P(x,s(y)), \neg P(x,s(s(x))))\} \vdash_{\mathbf{R1}} P(x,s(x))$$

puis que

$$\{P(x,x), \neg P(x,y) \vee P(x,s(y)), \neg P(x,s(s(x))))\} \vdash^*_{\mathbf{R1}} \bot$$

8.3.2 Correction

On montre d'abord que les règles de la Figure 8.1 sont correctes.

Lemme 18 (Correction des règles de résolution) *Si $\mathcal{C} \vdash^*_{\mathbf{R1}} C$ alors C est une conséquence logique de $\mathcal{C}$.*

Preuve On note $\forall C$ la *cloture universelle* de la clause C c'est à dire la formule obtenue en préfixant par une quantification universelle sur toutes les variables qui apparaissent dans C.

On établit la correction des règles de résolution en montrant que pour toute structure $\mathcal{A}$,

- si $\mathcal{A} \models \forall C_1$ et $\mathcal{A} \models \forall C_2$ et qu'on a
$$\text{Coupure}\frac{C_1 \qquad C_2}{C}$$
alors $\mathcal{A} \models \forall C$, et
- si $\mathcal{A} \models \forall C$ et $\mathcal{A} \models \forall C'$ et qu'on a
$$\text{Factorisation}\frac{C}{C'}$$
alors $\mathcal{A} \models \forall C'$.

□

Remarque 15 *Attention, la réciproque du Lemme 18 est fausse. La clause $\forall x, T(x) \vee \neg T(x)$ est valide et donc conséquence logique de tout $\mathcal{C}$. Pourtant si le symbole de prédixcat T n'apparaît pas dans $\mathcal{C}$, on ne peut avoir $\mathcal{C} \vdash^*_{\mathbf{R1}} \forall x, T(x) \vee \neg T(x)$!*

8.3.3 Complétude

On montre maintenant que les règles de la Figure 8.1 sont complètes.

On procède en établissant plusieurs lemmes.

Lemme 19 (Lemme de relèvement à un pas) *Soient deux clauses C_1 et C_2 et soient $D_1 \in H(C_1)$ et $D_2 \in H(C_2)$.*

Coupure : si $\{D_1, D_2\} \vdash_{\mathbf{R}} D_3$ par la règle de coupure, alors il existe une clause C_3 telle que $\{C_1, C_2\} \vdash_{\mathbf{R1}} C_3$ avec $D_3 \in H(C_3)$ par la règle de coupure.

Factorisation : si $\{D_1\} \vdash_{\mathbf{R}} D_3$ par la règle de factoriation, alors il existe une clause C_3 telle que $\{C_1\} \vdash_{\mathbf{R1}} C_3$ avec $D_3 \in H(C_3)$ par la règle de factorisation.

Preuve au tableau.

□

On peut alors montrer par induction sur la longueur de la dérivation en utilisant le Lemme 19 que :

Lemme 20 (Lemme de relèvement) *Si $\mathcal{C}$ est un ensemble de clauses et que $H(\mathcal{C}) \vdash^*_{\mathbf{R}} D$ alors il existe une clause C telle que $\mathcal{C} \vdash^*_{\mathbf{R1}} C$ avec $D \in H(C)$.*

Théorème 21 *Soit $\mathcal{C}$ un ensemble de clauses.*

$$\mathcal{C} \text{ est insatifaisable si, et seulement si, } \mathcal{C} \vdash^*_{\mathbf{R1}} \bot$$

Preuve Par le Lemme 18, si $\mathcal{C} \vdash^*_{\mathbf{R1}} \bot$ alors $\mathcal{C}$ n'est pas satisfaisable.

Réciproquement, si $\mathcal{C}$ n'est pas satisfaisable, alors d'après le Théorème 12 l'ensembles de clauses propositionnelles $H(\mathcal{C})$ ne l'est pas non plus. Puisque le système de résolution du calcul propositionnel est complet, on a $H(\mathcal{C}) \vdash^*_{\mathbf{R}} \bot$. Le Lemme 20 de relèvement nous donne une clause C telle que $\mathcal{C} \vdash^*_{\mathbf{R1}} C$ avec $\bot \in H(C)$. Mais alors le seule clause C dont une instance est $\bot$ est $\bot$ elle-même. Ce qui conclut. □

8.3.4 Bilan sur la résolution

Contrairement à la déduction naturelle, qui est directe (pour montrer $\varphi_1, \ldots, \varphi_n \models \varphi$ je démontre $\varphi_1, \ldots, \varphi_n \vdash_{DN} \varphi$), la résolution est un système de preuve *par réfutation*. Résumons comment utiliser la méthode de résolution.

Méthode de résolution
Pour montrer que $\varphi_1, \ldots, \varphi_n \models \varphi$ à l'aide de la résolution :

1. je considère les formules $\varphi_1, \ldots, \varphi_n$ et $\neg\varphi$,
   car je conçois que $\varphi_1, \ldots, \varphi_n \models \varphi$ ssi $\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}$ insatisfiable ;
2. je les passe en forme prénexe ;
3. je skolémise ;
4. je transforme en clauses ;
5. j'applique les règles de coupure et factorisation jusqu'à obtenir $\bot$.

Exemple 31 *Soit φ_1 la formule $\forall x\ ((\exists y\ \neg R(x,y)) \rightarrow \exists y\ (R(x,y) \wedge R(y,x)))$ et φ_2 la formule $\forall x\ \forall y\ \forall z\ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$.*

Soit φ la formule $\exists x\ R(x,x)$. On souhaite utiliser la méthode de résolution pour démontrer que $\varphi_1, \varphi_2 \models \varphi$.

1. *Je considère* φ_1, φ_2 *et* $\neg\varphi$.
2. $\varphi_1 \equiv \forall x \exists y \forall z (\neg R(x,z) \ \rightarrow (R(x,y) \wedge R(y,x)))$;
   $\varphi_2 \ \equiv \forall x \ \forall y \ \forall z \ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$;
   $\neg\varphi \equiv \forall x, \neg R(x,x)$.
3. *Les formes Skolemisées correspondantes à respectivement* φ_1, φ_2 *et* $\neg\varphi$ *sont :*
   venant de φ_1*:* $\forall x \forall z (\neg R(x,z) \ \rightarrow (R(x,f(x)) \wedge R(f(x),x)))$
   venant de φ_2*:* $\forall x \ \forall y \ \forall z \ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$
   venant de $\neg\varphi$*:* $\forall x, \neg R(x,x)$
4. *On obtient alors les clauses :*
   venant de φ_1*:* $R(x,z) \vee R(x,f(x))$ *et* $R(x,z) \vee R(f(x),x)$;
   venant de φ_2*:* $\neg R(x,y) \vee \neg R(y,z) \vee R(x,z)$
   venant de $\neg\varphi$*:* $\neg R(x,x)$

5. *On applique les règles de coupure et factorisation (les renommages ne sont pas indiqués) :*

(a) $R(x,z) \vee R(x,f(x))$	*donnée*
(b) $R(x,z) \vee R(f(x),x)$	*donnée*
(c) $\neg R(x,y) \vee \neg R(y,z) \vee R(x,z)$	*donnée*
(d) $\neg R(x,x)$	*donnée*
(e) $R(x,f(x))$	*coupure sur (a) et (d) avec* $[z := x]$
(f) $R(f(x),x)$	*coupure sur (b) et (d) avec* $[z := x]$
(g) $\neg R(f(x),z) \vee R(x,z)$	*coupure sur (e) et (c) avec* $[y := f(x)]$
(h) $R(x,x)$	*coupure sur (f) et (g) avec avec* $[z := x]$
(i) $\bot$	*coupure sur (h) et (d)*

Chapter 9

Théories du premier ordre

Étant donné un ensemble de formules E, on note $Cn(E)$ l'ensemble de toutes les conséquences logiques de E :

$$Cn(E) := \{\varphi \in \mathcal{L}^1 \mid E \models \varphi\}.$$

Exemple 32 *Considérons* $E := \{\forall x, R(x,x)\}$. *Alors* $Cn(E)$ *contient aussi les formules* $\forall x \exists y R(x,y)$, $\forall x \forall y (x = y) \to R(x,y)$, *toutes les validités, etc.*

9.1 Notion de théorie du premier ordre

Une théorie est un ensemble de formules, qui est clos par conséquence logique.

Définition 29 *Une* théorie *du premier ordre est un ensemble* $\mathcal{T}$ *de formules tel que* $Cn(\mathcal{T}) = \mathcal{T}$.

Remarque 16 *Dans la Définition 29 on peut aussi ne mettre que* $Cn(\mathcal{T}) \subseteq \mathcal{T}$ *puisqu'on pour tout ensemble de formules* E *on a toujours* $E \subseteq Cn(E)$.

Notez qu'une théorie contient toutes les validités. Il existe une plus petite théorie : celle qui contient uniquement les validités. À l'autre extrême, on trouve la théorie qui consiste en l'ensemble $\mathcal{L}^1$ tout entier; c'est la seule théorie insatisfaisable.

9.2 Théories à partir de structures et classes de structures

Définition 30 (Théorie d'une structure) *Pour structure* $\mathcal{A}$, *on définit la théorie de* $\mathcal{A}$ *par*

$$\mathit{Th}(\mathcal{A}) := \{\varphi \in \mathcal{L}^1 \mid \mathcal{A} \models \varphi\}.$$

La théorie d'une structure $\mathcal{A}$ est l'ensemble des formules qui sont vraies dans $\mathcal{A}$.

Définition 31 (Théorie d'une classe de structures) *Pour toute classe de structures* $\mathcal{K}$, *on définit la théorie de* $\mathcal{K}$ *par*

$$\mathit{Th}(\mathcal{K}) := \{\varphi \in \mathcal{L}^1 \mid \mathcal{A} \models \varphi, \text{ pour toute structure } \mathcal{A} \in \mathcal{K}\}.$$

En particulier, $\mathit{Th}(\mathcal{A}) = \mathit{Th}(\{\mathcal{A}\})$.

Si pour un ensemble E de formules, on note $\mathcal{K}_E$ la classe des modèles de l'ensemble E de formules, alors $\mathtt{Th}(\mathcal{K}_E)$ coïncide avec l'ensemble de toutes les conséquences logiques de E : $\mathtt{Th}(\mathcal{K}_E) = Cn(E)$.

9.3 Axiomatisation

Définition 32 (Théorie axiomatique) *Pour un ensemble récursif de formules* $\mathtt{Ax} \subseteq \mathcal{L}^1$*, appelé* ensemble d'axiomes*, on définit la* théorie axiomatisée *par* $\mathtt{Ax}$ *comme l'ensemble de formules* $\mathtt{Th}(\mathtt{Ax})$ *défini par*

$$\mathtt{Th}(\mathtt{Ax}) := \{\varphi \in \mathcal{L}^1 \mid \mathtt{Ax} \models \varphi\}.$$

Exercice 16 *Vérifier que les Définitions 30 et 32 produisent des théories.*

On peut s'intéresser à la classe des structures qui sont modèles d'une théorie axiomatisée.

Exemple 33 *L'ensemble fini de formules suivant caractérise la classe des groupes :* $\mathtt{Ax}_{groupes}$ $\begin{cases} \forall x \forall y \forall z[(x*y)*z = x*(y*z)] \\ \forall x[(x*e=x) \wedge (e*x=x)] \\ \forall x[(x*i(x)=e) \wedge (i(x)*x=e)] \end{cases}$

Exercice 17 *Proposer un ensemble fini d'axiomes pour le classes des corps.*

Plus généralement, on se pose des problèmes de déduction, tels que savoir si une formule donnée φ est conséquence logique de $\mathtt{Ax}$ ($\mathtt{Ax} \models \varphi$), c-à-d. de savoir si la propriété φ est vraie dans toute la classe $\mathcal{K}_{\mathtt{Th}(\mathtt{Ax})}$.

Exemple 34 *La formule* $\forall x \forall y[x*y = e \rightarrow y = i(x)]$ *qui exprime que dans un groupe l'inverse à droite est unique est vraie de tous les groupes, ce qui s'exprime par*

$$\mathtt{Ax}_{groupes} \models \forall x \forall y[x*y = e \rightarrow y = i(x)]$$

ou encore

$$\forall x \forall y[x*y = e \rightarrow y = i(x)] \in \mathtt{Th}(\mathtt{Ax}_{groupes})$$

Définition 33 (Théorie complète) *Une théorie* $\mathcal{T}$ *est* complète *si pour toute formule* φ*, on a* $\varphi \in \mathcal{T}$ *ou* $\neg\varphi \in \mathcal{T}$*.*

Exemple 35 *Par exemple, pour n'importe quelle structure* $\mathcal{A}$*, la théorie* $\mathtt{Th}(\mathcal{A})$ *est complète. A contrario, on remarque que seuls les groupes qui admettent un élément d'ordre 2 satisfont la formule* $\forall x(x \neq e \wedge x*x = e)$*. Par conséquent ni* $\forall x(x \neq e \wedge x*x = e)$ *ni sa négation ne sont conséquences logiques de* $\mathtt{Th}(\mathtt{Ax}_{groupes})$*, de sorte que cette théorie n'est pas complète.*

Exemple 36 *Dans le même esprit que l'Exemple 35, on peut établir que la théorie des corps n'est pas complète : par exemple en considérant la formules* $\mathbf{1}+\mathbf{1} = \mathbf{0}$*. En revanche, et la preuve est complexe, on peut montrer que la théorie des corps algébriquement clos de caractéristique nulle est complète.*

Cette théorie est axiomatique : on l'obtient en considérant les axiomes des corps auxquels on ajoute l'infinité d'axiomes suivants : $\mathbf{1}+\mathbf{1} \neq \mathbf{0}$*,* $(\mathbf{1}+\mathbf{1})+\mathbf{1} \neq \mathbf{0}$*,* $((\mathbf{1}+\mathbf{1})+\mathbf{1})+\mathbf{1} \neq \mathbf{0}$*, ... Notons que cet ensemble d'axiomes est certes infini, mais récursif.*

On dit que deux structures $\mathcal{A}$ et $\mathcal{B}$ sont *élémentairement équivalentes* si pour toute formule $\varphi \in \mathcal{L}^1$ on a $\mathcal{A} \models \varphi$ ssi $\mathcal{B} \models \varphi$. Il est alors facile d'établir que pour une classe $\mathcal{K}$ de structures, $\mathtt{Th}(\mathcal{K})$ est complète si, et seulement si, deux structures quelconques de $\mathcal{K}$ sont élémentairement équivalentes.

Exemple 37 *On peut montrer que* $(\mathbb{Q}, <)$ *et* $(\mathbb{R}, <)$ *sont élémentairement équivalentes.*

Il est facile de montrer que :

Proposition 22 *Une théorie* $\mathcal{T}$ *est complète si, et seulement si, deux modèles quelconques de* $\mathcal{T}$ *sont élémentairement équivalents.*

Preuve Soit $\mathcal{T}$ une théorie complète, et soit deux modèles $\mathcal{A}$ et $\mathcal{B}$ de $\mathcal{T}$. Soit φ une formule. Comme $\mathcal{T}$ est complète, on sait que $\mathcal{T} \models \varphi$ ou alors $\mathcal{T} \models \neg\varphi$. Ainsi, soit $\mathcal{A} \models \varphi$ et $\mathcal{B} \models \varphi$, ou alors $\mathcal{A} \models \neg\varphi$ et $\mathcal{B} \models \neg\varphi$. Ainsi, $\mathcal{A}$ et $\mathcal{B}$ sont élémentairement équivalentes.

Réciproquement, soit $\mathcal{T}$ une théorie telle que deux modèles quelconques de $\mathcal{T}$ sont élémentairement équivalents. Soit φ une formule. Si $\mathcal{T}$ n'est pas satisfiable, alors la question réglée : $\mathcal{T}$ est complète. Sinon, elle est satisfiable, et considérons un modèle $\mathcal{A}$. Deux cas : soit $\mathcal{A} \models \varphi$, ou alors $\mathcal{A} \models \neg\varphi$. Sans perte de généralité, plaçons nous dans le cas où $\mathcal{A} \models \varphi$. Considérons maintenant n'importe quel modèle $\mathcal{B}$ de $\mathcal{T}$. Comme $\mathcal{A}$ et $\mathcal{B}$ sont élémentairement équivalents, on a $\mathcal{B} \models \varphi$. Ainsi, $\mathcal{T} \models \phi$. Le cas $\mathcal{A} \models \neg\varphi$ donne $\mathcal{T} \models \neg\phi$. Donc $\mathcal{T}$ est complète. □

Définition 34 (Théorie axiomatisable) *Une théorie $\mathcal{T}$ est* axiomatisable *(resp.* finiment axiomatisable*) s'il existe un ensemble récursif (resp. fini) de formules $\mathtt{Ax}$ tel que $\mathcal{T} = \mathtt{Th}(\mathtt{Ax})$.*

Remarque 17 *De façon équivalente pour la définition d'une théorie finiment axiomatisable en Définition 34, on peut demander l'existence d'une seule formule θ telle que $\mathcal{T} = Cn(\theta)$, en prenant pour θ la conjonction finie des éléments de $\mathtt{Ax}$.*

Exemple 38 *La théorie des groupes est finiment axiomatisable. Celles des corps l'est également.*

Théorème 23 *Si $\mathtt{Th}(\mathtt{Ax})$ est finiment axiomatisable, alors il existe $\mathtt{Ax}_0 \subseteq \mathtt{Ax}$ fini telle que $\mathtt{Th}(\mathtt{Ax}_0) = \mathtt{Th}(\mathtt{Ax})$.*

Preuve Si $\mathtt{Th}(\mathtt{Ax})$ est finiment axiomatisable alors (d'après la Remarque 17) il existe une formule θ tel que $\mathtt{Th}(\mathtt{Ax}) = Cn(\theta)$. En général, $\theta \notin \mathtt{Ax}$, mais en revanche $\mathtt{Ax} \models \theta$. On utilise alors le théorème de compacité (Théorème 13) pour exhiber un sous-ensemble fini $\mathtt{Ax}_0 \subseteq \mathtt{Ax}$ tel que $\mathtt{Ax}_0 \models \theta$. Ainsi

$$Cn(\theta) \subseteq \mathtt{Th}(\mathtt{Ax}_0) \subseteq \mathtt{Th}(\mathtt{Ax})$$

ce qui conclut la preuve. □

Dans la suite de cette section, nous ne considérons que des signatures *raisonnables*, au sens où l'ensemble de toutes les formules est dénombrable.

Remarque 18 *On peut prendre dénombrable ou encore imposer que les ensembles $\{< f, n > \mid f \in \mathcal{F}_n\}$ et $\{< R, n > \mid R \in \mathcal{P}_n\}$ sont récursivement énumérables.*

Théorème 24

1. *Une théorie (sur une signature raisonnable) axiomatisable est récursivement énumérable;*
2. *Une théorie (sur une signature raisonnable) axiomatisable et complète est récursive.*

Nous montrons le Théorème 24 au travers des sous-sections qui suivent et qui répondent, au passage, à des problèmes particuliers.

9.3.1 Récursive énumérabilité des formules valides

Compte tenu des notations de la section précédente, l'ensemble $\mathtt{Th}(\emptyset)$ représente l'ensemble des formules de $\mathcal{L}^1_{\mathcal{S}}$ qui sont valides.

Théorème 25 *Sur une signature raisonnable, l'ensemble $\mathtt{Th}(\emptyset)$ est récursivement énumérable.*

Preuve Considérons un système de preuve correct et complet (lecteurs, vous êtes libres : résolution, déduction naturelle, etc.). Si une formule est valide, il en existe une preuve dans ce système. Ainsi, en énumérant toutes les preuves, on pourra énumérer toutes les validités.

Plus techniquement, on peut énumérer toutes les séquences finies $\varphi_1, \varphi_2, \ldots, \varphi_k$ de formules. Pour chaque séquence, on vérifie qu'elle est conforme à l'application de règles de notre système de preuve (*i.e.,* que c'est une déduction). Si ce n'est pas le cas, on passe à la séquence suivante, sinon on ajoute la formule φ_k dans liste des validités. □

Le corollaire suivant établit le Point 1. du Théorème 24 :

Corollaire 26 *Si E est un ensemble récursif de formules, alors l'ensemble $Cn(E)$ est récursivement énumérable.*

Preuve D'après le Théorème 25, on peut énumérer les validités. Dès lors qu'une validité à la forme $\varphi_1 \to \varphi_2 \to \ldots \varphi_n \to \psi$, on vérifie si chaque $\varphi_i \in E$ (ce qui est possible puisque E est récursif) et dans ce cas ajoute ψ à la liste des conséquences logiques de E. □

D'après le Théorème 25, l'ensemble $\mathtt{Th}(\emptyset)$ est récursivement énumérable. Par contre, il n'est pas récursif. Pour le comprendre, paraphrasons notre question en terme du problème de décision suivant :

VALIDE

Entrée : une formule $\varphi \in \mathcal{L}^1_{\mathcal{S}}$.
Sortie : φ est-elle valide ?

Dire que l'ensemble $\mathtt{Th}(\emptyset)$ est récursif est équivalent à dire que le problème VALIDE est décidable puisque φ est valide si, et seulement si, $\varphi \in \mathtt{Th}(\emptyset)$.

Si effectivement une formule $\varphi \in \mathtt{Th}(\emptyset)$, nous pouvons utiliser l'énumérateur de $\mathtt{Th}(\emptyset)$ (voir Théorème 25). En revanche, lorsque φ n'est pas valide, cet énumérateur ne nous est d'aucune utilité car comme on va le voir dans la section suivante, il ne peut existe de critère pour l'arrêter et décréter que la formule φ n'admet pas de preuve.

9.4 Théories complètes

Théorème 27 *Soit E un ensemble récursif de formules tel que pour toute formule $\varphi \in E$, on a $E \models \varphi$ ou $E \models \neg\varphi$. Alors l'ensemble $Cn(E)$ est récursif.*

Preuve Nous allons donner un algorithme qui, sur une formule donnée φ, répond à la question "$\varphi \in Cn(E)$?", c-à-d. "$E \models \varphi$?". L'algorithme énumère l'ensemble $Cn(E)$ (c'est possible grâce au Corollaire 26). Si la formule ϕ apparaît, on répond oui. Sinon, par hypothèse, la formule $\neg\phi$ apparaîtra, et on répond non quand $\neg\phi$ apparaît. □

Le Point 2. du Théorème 24 est un corollaire de ce dernier théorème.

Nous résumons la situation en Figure 9.1.

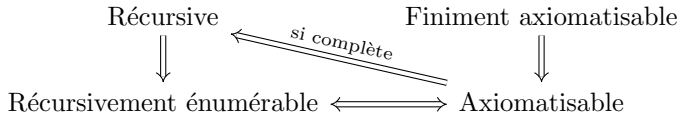


Figure 9.1: Propriétés des théories pour une signature raisonnable.

Exercice 18 *Soit $\mathcal{T}$ une théorie récursivement énumérable. Montrer qu'elle admet une axiomatisation récursive, i.e. un ensemble récursif $\mathtt{Ax}$ de formules tel que $\mathcal{T} = \mathtt{Th}(\mathtt{Ax})$.*

9.5 Théorie des nombres

9.6 Théories axiomatisées particulières

9.6.1 Théorie de l'égalité

$\mathtt{Ax}_=$

$$\begin{cases} \forall x x = x \\ \forall x \forall y (x = y) \rightarrow (y = x) \\ \forall x \forall y \forall z (x = y \wedge y = z) \rightarrow x = z \\ \forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (\bigwedge_i (x_i = y_i)) \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n) \text{ pour tout } f \in \mathcal{F}_n \\ \forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (\bigwedge_i (x_i = y_i)) \rightarrow [R(x_1, \ldots, x_n) \leftrightarrow R(y_1, \ldots, y_n))] \text{ pour tout } R \in \mathcal{P}_n \end{cases}$$

Dans les deux prochaines sections, On s'intéresse à la signature pariculière suivante ;
Les symboles de fonctions sont :

- $\mathbf{0}(0)$ pour dénoter l'entier 0,
- $\mathbf{1}(0)$ pour le successeur,
- $+(2)$ pour la somme,
- $\times(2)$ pour la multiplication,

avec l'unique symbole de relation $=$ (pour l'égalité).

On convient d'utiliser une notation infixe pour décrire les termes sur cette signature.

9.6.2 Arithmétique de Preburger

$$\mathtt{Ax}_{Presb} \begin{cases} \psi \text{ pour toute } \psi \in \mathtt{Ax}_{=} \\ \forall x (x + \mathbf{1} \neq \mathbf{0}) \\ \forall x [(x = \mathbf{0}) \vee \exists y (x = y + \mathbf{1})] \\ \forall x \forall y [x + \mathbf{1} = y + \mathbf{1} \rightarrow x = y] \\ \forall x (x + \mathbf{0} = x) \\ \forall x \forall y (x + (y + \mathbf{1}) = (x + y) + \mathbf{1})) \\ (\varphi(\mathbf{0}) \wedge (\forall y \varphi(y) \rightarrow \varphi(y + \mathbf{1}))) \rightarrow \forall x \phi(x) \text{ pour toute formule } \varphi \end{cases}$$

Théorème 28 *$\mathtt{Th}(\mathtt{Ax}_{Presb})$ est récursive.*

Preuve Elle est axiomatisable et complète (Presbuger 1929), on applique alors le Théorème 24. □

Un modèle particulier de $\mathtt{Th}(\mathtt{Ax}_{Presb})$ est la structure de domaine $\mathbb{N}$ où les symboles de la signature sont interprétés naturellement : $\mathbf{0}$ est l'entier 0, $\mathbf{1}$ est l'entier 1, etc.

Théorème 29 *Le problème de décision*
Arithmétique de Presburger

Entrée : *une formule φ close sur la signature $\sigma = \{\mathbf{0}, \mathbf{1}, +, =\}$*
Sortie : *$\mathbb{N} \models \varphi$?*

est décidable.

Preuve Voir la section sur les structures automatiques. □

Corollaire 30 *La théorie $\mathtt{Th}((\mathbb{N}, \mathbf{0}, \mathbf{1}, +, =))$ est récursive.*

9.6.3 Arithmétique de Peano

L'ensemble d'axiomes étend $\mathtt{Ax}_{Presb}$:

$$\mathtt{Ax}_{Peano} \begin{cases} \psi \text{ pour toute } \psi \in \mathtt{Ax}_{Presb} \\ \forall x, x \times \mathbf{0} = x \\ \forall x, y, x \times (y + \mathbf{1}) = (x \times y) + x \end{cases}$$

Théorème 31 (admis) *$\mathcal{Th}(\mathcal{Ax}_{Peano})$ est indécidable.*

Théorème 32 (admis) *Le problème de décision*
Arithmétique de Peano

Entrée : *une formule φ close sur la signature $\sigma = \{\mathbf{0}, \mathbf{1}, +, \times, =\}$*
Sortie : *$\mathbb{N} \models \varphi$?*

est indécidable.

9.6.4 Théorie des ordres denses

Chapter 10

Théorie des modèles finis et Jeux d'Ehrenfeucht-Fraïssé

10.1 Utilisation du Théorème de Compacité

Le Théorème de Compacité de la logique du premier s'énonce pour la classe de toutes les $\mathcal{S}$-structures, mais il devient faux lorsqu'on se restreint au structures finis. En effet, on a le théorème suivant.

Théorème 33 (Échec du Théorème de Compacité sur les structures finies) *Il existe une théorie $\mathcal{T}$ telle que $\mathcal{T}$ n'a pas de modèle fini, et tout sous-ensemble fini de $\mathcal{T}$ admet un modèle fini.*

Preuve On définit $\mathcal{T} := \{\lambda_n \in \mathcal{L}^1 \mid n \in \mathbb{N}\}$ sur la signature vide (mais où les formules peuvent utiliser l'égalité), où λ_n énonce que le domaine à au moins n éléments distincts. □

On peut toutefois exploiter le Théorème de Compacité pour prouver l'inexprimabilité de certaines propriétés en logique du premier ordre. C'est le cas de la propriété **Pair** définie par :

Pair($\mathcal{A}$) (*i.e.,* la structure $\mathcal{A}$ a la propriété **Pair**) ssi le domaine de $\mathcal{A}$ est de cardinal pair.

Noter que la propriété **Pair** n'a de sens que pour les structures finies.

Proposition 34 *La propriété* **Pair** *sur les $\emptyset$-structures n'est pas exprimable en logique du premier ordre.*

Preuve Si la propriété **Pair** est exprimable par une formule $\varphi_{\mathbf{Pair}} \in \mathcal{L}^1_\emptyset$, on définit les deux théories

$$\mathcal{T}_1 := \{\varphi_{\mathbf{Pair}}\} \cup \{\lambda_n | n \in \mathbb{N}\} \text{ et } \mathcal{T}_2 := \{\neg\varphi_{\mathbf{Pair}}\} \cup \{\lambda_n | n \in \mathbb{N}\}$$

Par le Théorème de Compacité, ces deux théories sont satisfaisables, et n'ont que des modèles infinis. Par le Théorème de Lowenheim-Skolem, elles ont des modèles dénombrables, soient $\mathcal{A}_1$ et $\mathcal{A}_2$, respectivement. Puisque la signature est vide, ces deux modèles sont isomorphes, $\mathcal{A}_1 \cong \mathcal{A}_2$, et satisfont donc les mêmes formules, ce qui contredit $\mathcal{A}_1 \models \varphi_{\mathbf{Pair}}$ et $\mathcal{A}_2 \models \neg\varphi_{\mathbf{Pair}}$. □

Toutefois la Proposition 34 repose sur le fait que les structures pour une signature vide. Quid du cas des ordres linéraires finis, donc avec un symbole de relation binaire $<$ où les théories $\mathcal{T}_1$ et $\mathcal{T}_1$ sont enrichies par les axiomes des ordres linéraires ? Le preuve ci-dessus ne marche plus car on peut très bien avoir deux ordres linéaires dénombrables qui ne sont pas isomorphes, typiquement $(\mathbb{N}, <)$ et $(\mathbb{Q}, <)$.

Dans ce chapitre nous décrivons les jeux de Ehrenfeucht-Fraïssé comme un outil puissant permettant d'établir l'inexprimabilité d'une propriété en logique du premier ordre sur la classe des structures finies.

10.2 Notions préliminaires et rappels

Définition 35 *On définit le* rang (de quantificateurs) *d'une formule $\varphi \in Fpo$, noté $\mathtt{rq}(\varphi)$, comme un entier défini par induction :*

- $\mathtt{rq}(\varphi) = 0$ *si φ n'a pas de quantificateur;*
- $\mathtt{rq}(\neg\varphi) = \mathtt{rq}(\varphi)$;
- $\mathtt{rq}(\varphi \wedge \psi) = \mathtt{rq}(\varphi \vee \psi) = \mathtt{rq}(\varphi \to \psi) = \max(\mathtt{rq}(\varphi), \mathtt{rq}(\psi))$;
- $\mathtt{rq}(\exists x\varphi) = \mathtt{rq}(\forall x\varphi) = 1 + \mathtt{rq}(\varphi)$.

On note $\mathcal{L}^1_n := \{\varphi \in \mathcal{L}^1 \mid \mathtt{rq}(\varphi) \leq n\}$ le sous-ensemble des formules de rang au plus n.

On rappelle la définition suivante.

Définition 36 *Soit une signature $\mathcal{S}$. Deux $\mathcal{S}$-structures $\mathcal{A}$ et $\mathcal{B}$ sont* élémentairement équivalentes, *noté $\mathcal{A} \equiv \mathcal{B}$, si pour toute formule close $\varphi \in \mathcal{L}^1$, $\mathcal{A} \models \varphi$ si, et seulement si, $\mathcal{B} \models \varphi$.*
On définit pour chaque $n \in \mathbb{N}$, l'équivalence $\mathcal{A} \equiv_n \mathcal{B}$ par : pour tout formule close $\varphi \in \mathcal{L}^1_n$, $\mathcal{A} \models \varphi$ si, et seulement si, $\mathcal{B} \models \varphi$.

Lemme 35 *$\mathcal{A} \equiv \mathcal{B}$ si, et seulement si, $\mathcal{A} \equiv_n \mathcal{A}$, pour tout $n \in \mathbb{N}$.*

On rappelle qu'une théorie $\mathcal{T}$ est complète si deux modèles quelconques de $\mathcal{T}$ sont élémentairement équivalents.
On se fixe une signature $\mathcal{S} = (\mathcal{F}, \mathcal{P})$, implicite dans la section.

Définition 37 *Un* isomorphisme *de $\mathcal{A}_1 = (D_1, \{f^{\mathcal{A}_1}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}_1}\}_{P\in\mathcal{P}})$ dans $\mathcal{A}_2 = (D_2, \{f^{\mathcal{A}_2}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}_2}\}_{P\in\mathcal{P}})$ est une bijection $h : D_1 \to D_2$ telle que pour tous $d_1, \ldots, d_n, d \in D_1$ on a :*

1. *pour tout $f \in \mathcal{F}_n$, $h(f^{\mathcal{A}_1}(d_1, \ldots, d_n) = d$ ssi $f^{\mathcal{A}_2}(h(d_1), \ldots, h(d_n)) = h(d)$, et*
2. *pour tout $R \in \mathcal{P}_n$, $(d_1, \ldots, d_n) \in R^{\mathcal{A}_1}$ ssi $(h(d_1), \ldots, h(d_n)) \in R^{\mathcal{A}_2}$.*

On notera $\mathcal{A}_1 \cong \mathcal{A}_2$ s'il existe un isomorphisme de $\mathcal{A}_1$ dans $\mathcal{A}_2$.

Remarque 19 *La relation $\cong$ est une relation d'équivalence sur les $\mathcal{S}$-structures.*

Clairement, $\mathcal{A}_1 \cong \mathcal{A}_2$ implique $\mathcal{A} \equiv \mathcal{B}$, mais la reciproque est fausse en général, prendre par exemple les structures $(\mathbb{Q}, \leq)$ et $(\mathbb{R}, \leq)$. Toutefois, on peut établir que c'est le cas pour les structures finies.

Lemme 36 *Pour toute structure finie $\mathcal{A}$, il existe une formule close $\chi_{\mathcal{A}}$ telle que $\mathcal{B} \models \chi_{\mathcal{A}}$ si, et seulement si, $\mathcal{B} \cong \mathcal{A}$.*

Preuve On le montre lorsque $\mathcal{A}$ est un graphe, *i.e.,* pour une signature avec un seul symbole de relation binaire E. Soit donc $\mathcal{A} = (\{a_1, \ldots, a_n\}, E)$[1]. On pose

$$\chi_{\mathcal{A}} := \exists x_1 \ldots \exists x_n \left(\begin{array}{l} (\bigwedge_{i\neq j} \neg(x_i = x_j)) \wedge (\forall x \bigvee_i (x = x_i)) \\ \wedge (\bigwedge_{(a_i,a_j)\in E} E(x_i, x_j)) \wedge (\bigwedge_{(a_i,a_j)\notin E} \neg E(x_i, x_j)) \end{array} \right)$$

et on vérifie facilement que $\mathcal{B} \models \chi_{\mathcal{A}}$ implique $\mathcal{B} \cong \mathcal{A}$. □

Un corollaire immédiat du Lemme 36 est que deux structures finies $\mathcal{A}$ et $\mathcal{B}$élémentairement équivalentes sont isomorphes.

Corollaire 37 *Sur les structures finies, isomorphisme et équivalence élémentaire coincident.*

Remarque 20 *L'isomorphisme de deux structures finies est un problème complexe : il est dans* NP*, mais on ne sait pas à ce jour s'il est dans* P *ou* NP*-complet.*

[1] Rigoureusement, on devrait écrire $E^{\mathcal{A}}$ au lieu de E.

10.3 Jeux d'Ehrenfeucht-Fraïssé

10.3.1 Notion d'isomorphisme partiel de structures

Définition 38 *Un* isomorphisme partiel *de* $\mathcal{A}_1 = (D_1, \{f^{\mathcal{A}_1}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}_1}\}_{P\in\mathcal{P}})$ *dans* $\mathcal{A}_2 = (D_2, \{f^{\mathcal{A}_2}\}_{f\in\mathcal{F}}, \{P^{\mathcal{A}_2}\}_{P\in\mathcal{P}})$ *est une fonction (partielle) injective* $h : D_1 \to D_2$ *telle que pour tous* $d_1, \ldots, d_n, d \in dom(h)$ *on a les Points 1. et 2. de la Definition 37.*

On peut représenter un isomorphisme partiel h de domaine fini $\{d_1, \ldots, d_n\}$ par son graphe

$$\{(d_1, h(d_1)), \ldots, (d_n, h(d_n))\}.$$

Exemple 39 *Soit* $\mathcal{S} = \{\emptyset, \{E(2), =\}\}$. *On considère la* $\mathcal{S}$*-structure* $\mathcal{A}$ *qui consiste en un graphe non-orienté* G *à* 5 *sommets a* $1, \ldots, 5$ *reliés en anneau. La* $\mathcal{S}$*-structure* $\mathcal{B}$ *consiste en deux copies disjointes de* G. *Montrer que, quels que soient les deux sommets* b_1, b_2 *choisis dans le domaine de* $\mathcal{B}$, *on peut trouver deux sommets dans* G *tels que la fonction définie par* $h(a_1) = b_1$ *et* $h(a_2) = b_2$ *soit un isomorphisme partiel de* $\mathcal{A}$ *dans* $\mathcal{B}$. *Qu'en est-il pour* 3 *sommets ?*

10.3.2 Définition des jeux d'Ehrenfeucht-Fraïssé

Deux joueurs Spoiler (joueur **I**) et Duplicator (joueur **II**), une arène donnée par une paire de $\mathcal{S}$-structures $\mathcal{A}$ et $\mathcal{B}$ de domaines $A = \{a, a_1, \ldots\}$ et $B = \{b, b_1, \ldots\}$, respectivement. Le joueur Spoiler essaie de montrer que les structures ne sont pas isomorphes alors que Duplicator essaie de montrer qu'elles le sont mais en le justifiant par "petits bouts".

Pour chaque $n \in \mathbb{N}$, on définit le jeu, noté $EF_n(\mathcal{A}, \mathcal{B})$, qui se déroule comme suit : Le joueur **I** (Spoiler) choisit un élément a_1 de A. Le joueur **II** (Duplicator) doit alors répondre en donnant un élément $b_1 \in B$ tel que $\{(a_1, b_1)\}$ soit un isomorphisme partiel. Alternativement le joueur **I** peut choisir un élément b_1 de B, et alors le joueur **II** doit choisir un élément $a_1 \in A$ tel que $\{(a_1, b_1)\}$ soit un isomorphisme partiel. L'élément $\{(a_1, b_1)\}$ est appelé la *position* dans le jeu après le premier tour.

Si le jeu peut atteindre une position $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ (après n tours donc) alors Duplicator gagne, sinon, c'est que Duplicator a été bloqué avant le $n$ième tour, et c'est alors Spoiler qui gagne. Noter que chaque position atteinte doit décrire (le graphe d') un isomorphisme partiel de $\mathcal{A}$ vers $\mathcal{B}$.

On peut aussi définir le jeu $EF_\infty(\mathcal{A}, \mathcal{B})$ dans lequel on poursuit potentiellement indéfiniment la partie (i.e. tant que **II** n'est pas bloqué), mais ce jeu n'a pas beaucoup d'intérêt lorsqu'on s'intéresse à la classe des structures finies, comme c'est le cas dans ce chapitre.

Pour la suite, on reste informel pour les notions de *stratégie* et de *stratégie gagnante* auxquelles on donne leur sens habituel dans un jeu à deux joueurs.

Exemple 40 *Considérons deux* $\emptyset$*-structures* $\mathcal{A}$ *et* $\mathcal{B}$, *c'est à dire des structures que une signature* $\mathcal{S} = \emptyset$, *qui sont donc justes des domaines. Alors le joueur* ***II*** *a une stratégie gagnante dans le jeu* $EF_n(\mathcal{A}, \mathcal{B})$ *dès lors que* $\mathcal{A}$ *et* $\mathcal{B}$ *ont au moins* n *éléments.*

L'intuition est qu'en ne jouant qu'un nombre fini de tours (et même un nombre infini de tours), il n'est pas possible d'identifier le cardinal du domaine si ce dernier est infini. Prendre par exemple $\mathcal{A} = \mathbb{N}$ *et* $\mathcal{B} = \mathbb{R}$.

Exemple 41 *Soit* $\mathcal{A}$ *un ordre linéaire à 3 éléments et* $\mathcal{B}$ *un ordre linéaire à 4 éléments. Combien de tours faut-il à* ***I*** *pour battre* ***II****?*

Proposition 38 *Chaque jeu* $EF_n(\mathcal{A}, \mathcal{B})$ *est* déterminé, i.e., *un des deux joueurs possède une stratégie gagnante. Il en est de même pour* $EF_\infty(\mathcal{A}, \mathcal{B})$.

Preuve Supposons que **I** n'a pas de stratégie gagnante dans $EF_n(\mathcal{A}, \mathcal{B})$. Donc quelque soit son premier choix, **II** a une réponse pour atteindre une position dans laquelle **I** n'a pas de stratégie gagnante dans le jeu résultant à $n-1$ tours. Et le même argument s'applique à ce dernier jeu.

Ainsi **II** a une stratégie pour maintenir l'invariant "**I** n'a pas de stratégie gagnante dans le jeu partant de la position courante", c'est à dire que **II** a une stratégie qui garantie de ne pas perdre. De ppart les conditions de victoire du jeu $EF_n(\mathcal{A}, \mathcal{B})$, cette stratégie de **II** est gagnante.

On peut mener le même raisonnement pour le jeu $EF_\infty(\mathcal{A}, \mathcal{B})$. □

On remarquera que les jeux $EF_n(\mathcal{A}, \mathcal{B})$ et $EF_n(\mathcal{B}, \mathcal{A})$ (resp. $EF_\infty(\mathcal{A}, \mathcal{B})$ et $EF_\infty(\mathcal{B}, \mathcal{A})$) sont "équivalents" au sens où Duplicator a une stratégie gagnante dans le premier si, et seulement si, il a une stratégie gagnante dans le deuxième. On note $\mathcal{A} \approx_{EF_n} \mathcal{B}$ (resp. $\mathcal{A} \approx_{EF_\infty} \mathcal{B}$) lorsque Duplicator a une stratégie gagnante dans $EF_n(\mathcal{A}, \mathcal{B})$ (resp. $EF_\infty(\mathcal{A}, \mathcal{B})$). On remarquera que, d'après la Proposition 38, $\mathcal{A} \not\approx_{EF_n} \mathcal{B}$ signifie que le joueur **I** a une stratégie gagnante dans $EF_n(\mathcal{A}, \mathcal{B})$.

Lemme 39 *$\approx_{EF_n}$ et $\approx_{EF_\infty}$ sont des relations d'équivalence entre structures.*
Preuve □

On peut établir de nombreuses propriétés sur les relations d'équivalences $\approx_{EF_n}$.

Proposition 40 *Pour toutes $\mathcal{S}$-structures $\mathcal{A}$ et $\mathcal{B}$, et tout entier n :*

- *$\mathcal{A} \approx_{EF_0} \mathcal{B}$ (correspondant à l'isomorphisme partiel de domaine vide);*
- *$\mathcal{A} \approx_{EF_n} \mathcal{A}$, et plus généralement, $\mathcal{A} \cong \mathcal{B}$ implique $\mathcal{A} \approx_{EF_n} \mathcal{B}$;*
- *Si $\mathcal{A} \approx_{EF_n} \mathcal{B}$ et $m < n$, alors $\mathcal{A} \approx_{EF_m} \mathcal{B}$; de sorte que si $\mathcal{A} \not\approx_{EF_n} \mathcal{B}$ et $m > n$, alors $\mathcal{A} \not\approx_{EF_m} \mathcal{B}$;*
- *si $|A| \leq n$ et $\mathcal{A} \approx_{EF_{n+1}} \mathcal{B}$, alors $\mathcal{A} \cong \mathcal{B}$;*

Preuve □

Exercice 19 *Montrer que $\mathcal{A} \approx_{EF_\infty} \mathcal{B}$ implique $\mathcal{A} \approx_{EF_n} \mathcal{B}$, pour tout n. Montrer que la réciproque est fausse (On pourra remarquer que $(\mathbb{N}, <) \not\approx_{EF_\infty} (\mathbb{N} + \mathbb{Q}, <)$).*

10.3.3 Exemples de jeux d'Ehrenfeucht-Fraïssé

Jeux EF sur les ensembles

Proposition 41 *Pour tout $n \in \mathbb{N}$, si $|A|, |B| \geq n$ alors $\mathcal{A} \approx_{EF_n} \mathcal{B}$.*

Preuve La stratégie pour le joueur **II** est la suivante : supposons qu'on a déjà joué i tours et que la position est $\{(a_1, b_1), \ldots, (a_i, b_i)\}$, et supposons que **I** choisisse un élément a_{i+1} dans A. Si $a_{i+1} = a_j$ pour un $j \leq i$, alors **II** choisit $b_{i+1} = b_j$, sinon **II** choisit $b_{i+1} \in B \setminus \{b_1, \ldots, b_i\}$ qui existe puisque $|B| \geq n$. □

Jeux EF sur les ordres linéaires On pourra jouer avec `https://trkern.itch.io/efglo` et voir que pour toute paire d'ordres linéaires finis distincts, et tout entier n, il faut garantir un nombre d'éléments dans les domaines des structures pour que **II** ait une stratégie gagnante.

Théorème 42 *Soit $n \in \mathbb{N}$, et deux ordres linéaires $\mathcal{A} = (A, <)$ et $\mathcal{B} = (B, <)$. Si $|A|, |B| \geq 2^n$ alors $\mathcal{A} \approx_{EF_n} \mathcal{B}$.*
Preuve □

10.3.4 Théorème d'Ehrenfeucht-Fraïssé

On admettra le théorème fondamental suivant.

Théorème 43 (Ehrenfeucht-Fraïssé) *Soient deux $\mathcal{S}$-structures $\mathcal{A}$ et $\mathcal{B}$ où $\mathcal{S}$ est purement relationnelle (*i.e., *pas de symbole de fonction). Alors,*

$$\mathcal{A} \equiv_n \mathcal{B} \text{ ssi } \mathcal{A} \approx_{EF_n} \mathcal{B} \tag{10.1}$$

$$\mathcal{A} \equiv \mathcal{B} \text{ ssi } \mathcal{A} \approx_{EF_\infty} \mathcal{B} \tag{10.2}$$

Exercice 20 *Expliquer pourquoi l'Equation* (10.2) *n'a pas grand intérêt si l'une au moins des structure est finie.*

On exploite le Théorème 43 pour établir que certaines propriétés sur les structures finis ne sont pas exprimables en logique du premier ordre.

10.4 Application des jeux d'Ehrenfeucht-Fraïssé

10.4.1 Méthode de preuve pour l'inexprimabilité de propriétés sur les structures finies

En ce qui concerne les structures finies, on ne peut pas comme, pour les cas des structures arbitraires, établir l'inexprimabilité d'une propriété en exhibant deux structures $\mathcal{A}$ et $\mathcal{B}$ élémentairement équivalentes mais qui pourtant sont distinquées par cette propriété. En effet, par le Corollaire 37, deux structures finies élémentairement équivalentes sont forcément isomorphes, et de fait satisfont les mêmes propriétés.

L'approche classique pour prouver l'inexprimabilité d'une propriété dans les structures finies est néanmoins très proche de cette idée : Au lieu d'exhiber deux structures $\mathcal{A}$ et $\mathcal{B}$, on exhibe deux familles de structures $\{\mathcal{A}_k \mid k \in I\!N\}$ et $\{\mathcal{B}_k \mid k \in I\!N\}$ telles que

- $\mathcal{A}_k \equiv_k \mathcal{B}_k$, *i.e.,* (Théorème 43 d'Ehrencfeucht-Fraïssé) $\mathcal{A}_k \approx_{EF_k} \mathcal{B}_k$, et
- $\mathcal{A}_k$ a la propriété $\mathbb{P}$ mais $\mathcal{B}_k$ ne l'a pas.

De telles familles montrent que $\mathbb{P}$ n'est pas exprimable en logique du premier ordre :

Corollaire 44 *Soit $\mathcal{S}$ une signature. Une propriété $\mathbb{P}$ sur les $\mathcal{S}$-structures n'est pas exprimable dans $\mathcal{L}^1_{\mathcal{S}}$ si pour tout $k \in I\!N$, il existe deux $\mathcal{S}$-structures finies $\mathcal{A}_k$ et $\mathcal{B}_k$ telles que $\mathcal{A}_k \approx_{EF_k} \mathcal{B}_k$, et $\mathcal{A}_k$ a la propriété $\mathbb{P}$ mais $\mathcal{B}_k$ ne l'a pas.*

Preuve Si $\mathbb{P}$ était exprimable en $\mathcal{L}^1_{\mathcal{S}}$ alors il existerait une formule $\varphi_{\mathbb{P}}$, et soit $k = \mathtt{rq}(\varphi_{\mathbb{P}})$. Mais alors les structures $\mathcal{A}_k$ et $\mathcal{B}_k$ seraient telles que $\mathcal{A}_k \approx_{EF_k} \mathcal{B}_k$, de sorte que lorsque $\mathcal{A}_k$ a la propriété $\mathbb{P}$, $\mathcal{B}_k$ l'a aussi. Contradiction. □

Remarque 21 *La méthode basée sur deux familles de structures peut s'appliquer à d'autres logiques dès lors qu'on dispose d'une partition infinie des formules de la logique (ici c'est la partition $\mathcal{L}^1_0, \mathcal{L}^1_1, \ldots$ selon le rang de quantificateurs) et d'une une technique pour justifier que deux structures s'accordent sur les formules d'une même classe de la partition (ici ce sont les jeux d'Ehrenfeucht-Fraïssé).*

Le Corollaire 44 s'étend en le Corollaire 45 qui concernent des propriétés "non-closes" appelées *requêtes*.

On introduit d'abord quelques notations : étant donnée une $\mathcal{S}$-structure $\mathcal{A}$, et $a \in A$ on note $(\mathcal{A}; a)$ la structure sur la signature qui étend $\mathcal{S}$ avec le symbole de constante a interprété dans $(\mathcal{A}; a)$ comme a lui-même, *i.e.,* $a^{(\mathcal{A};a)} = a$. On étend cette notation à un tuple $\vec{a}$ d'élements de A.

Définition 39 (Requête) *Une* requête *d'arité ℓ sur les $\mathcal{S}$-structures est une application, notée $\mathbb{R}$, qui à chaque $\mathcal{S}$-structure $\mathcal{A}$ associe un sous-ensemble $\mathbb{R}(\mathcal{A})$ de A^ℓ, et qui est close par isomorphisme : si $\mathcal{A} \cong \mathcal{B}$ par l'isomorphime $h : A \to B$, alors $\mathbb{R}(\mathcal{B}) = h(\mathbb{R}(\mathcal{A}))$.*

Une requête $\mathbb{R}$ d'arité ℓ est exprimable dans $\mathcal{L}^1_{\mathcal{S}}$ s'il existe une formule $\varphi_{\mathbb{R}} \in \mathcal{L}^1_{\mathcal{S}}$ à au plus ℓ variables libres telle que $\mathbb{R}(\mathcal{A}) = \{\vec{a} \in A^\ell \mid (\mathcal{A}; \vec{a}) \models \varphi_{\mathbb{R}}(\vec{a})\}$, pour toute $\mathcal{S}$-structure $\mathcal{A}$.

Formellement, une propriété *est une requête d'arité* 0.

Exemple 42 *Dans les graphes, avec un unique symbole E de relation binaire, on peut considérer la propriété* **Connexe** *qui caractérise les graphes connexes, ou encore la requête* **Reach** *d'arité* 2 *dite d'*atteignabilité*, qui décrit l'ensemble des paires de sommets d'un graphe reliés par un chemin.*

Exemple 43 *Dans la classe des $\emptyset$-structures,* i.e., *les ensembles, on peut considérer la propriété* **Fin** *qui caractérise les ensemble finis, ou encore dans la classe des $\emptyset$-structures finies, la propriété* **Pair** *pour les ensembles finis de cardinal pair.*

Corollaire 45 *Une requête $\mathbb{R}$ d'arité ℓ n'est pas exprimable dans $\mathcal{L}^1(x_1, \ldots, x_\ell)$ si pour tout $k \in \mathbb{N}$, il existe deux $\mathcal{S}$-structures finies $\mathcal{A}_k$ et $\mathcal{B}_k$, et deux ℓ-tuples $\vec{a}$ et $\vec{b}$ tels que*

- $(\mathcal{A}; \vec{a}) \approx_{EF_k} (\mathcal{B}; \vec{b})$, *et*
- $\vec{a} \in \mathbb{R}(\mathcal{A}_k)$ *et* $\vec{b} \notin \mathbb{R}(\mathcal{B}_k)$.

10.4.2 Quelques exemples d'inexprimabilité en logique du premier ordre

On a utilisé le Théorème de Compacité pour établir que la propriété **Pair** n'est pas exprimable dans la classe des ensembles finies, et on a justifier qu'on ne pouvait rien conclure pour les ordres linéaires. On peut à présent utiliser une preuve basée sur le Théorème d'Ehrenfeucht-Fraïssé.

Théorème 46 *La propriété* **Pair** *pour des ordres linéaires finis n'est pas exprimable en logique du premier ordre.*

Preuve On prend les ordres linéaires $\mathcal{A}_k$ et $\mathcal{B}_k$ de cardinaux respectifs 2^k et $2^k + 1$ qui sont satisfont les hypothèses du Corollaire 44. □

Si le Théorème de Compacité permet d'établir que la propriété **Connexe** n'est pas exprimable en logique du premier ordre sur la classe de tous les graphes, le Théorème d'Ehrenfeucht-Fraïssé permet d'établir qu'elle ne l'est pas non plus pour les graphes finis :

Théorème 47 *La propriété* **Connexe** *dans les graphes finis n'est pas exprimable en logique du premier ordre.*

Preuve On tranforme chaque ordre linéaire fini en un graphe fini de sorte que cet ordre linéaire fini est de cardinal pair si, et seulement si, le graphe associé n'est pas connexe. On conclut en utilisant le Théoreme 46 d'inexprimabilité de la propriété **Pair** pour des ordres linéaires finis. □