LOGIQUE COMPUTATIONNELLE & PROLOG – 3

CALCUL DES PRÉDICATS – I

3.1

La notion du prédicat

Pour pouvoir utiliser la logique en tant que mode de calcul il faut d'abord procéder à une conceptualisation de la partie du monde qui nous intéresse, c'est-à-dire établir les objets du monde et leurs inter-relations. Il s'agit, en réalité, de procéder à une *représentation de la connaissance*. Les objets sur lesquels nous allons exprimer notre connaissance constituent l'*univers du discours*. La représentation de la connaissance est la mise en relation des plusieurs objets de l'univers du discours par l'intermédiaire de la formulation des fbf.

Le calcul propositionnel ne permet pas d'examiner les relations intra-propositionnelles. Par exemple la phrase « tous les élèves aiment la logique » sera vue en calcul propositionnele comme une proposition p. On ne pourra donc pas la distinguer d'une autre du type « il existe des élèves qui aiment la logique ». qui sera, elle aussi, notée p. Il faut donc introduire dans le langage des symboles qui désignent soit l'existence, soit l'universalité et, aussi, des variables.

Nous avons ainsi un cas particulier de propositions qu'on pourrait considérer comme étant des *fonctions propositionnelles logiques* et qui sont des fonctions au sens classique du terme mais dont le résultat est une valeur de vérité - 0 (faux) ou 1 (vrai). Ces fonctions propositionnelles logiques on les appellera, par la suite, *prédicats*.

Il existe bien sûr toujours des fonctions au sens classique du terme, c-à-d. des fonctions dont le résultat est un objet de l'univers du discours et non pas une valeur de vérité. Pour les distinguer des autres fonctions, qui sont les prédicats, nous les appellerons *foncteurs*.

Nous allons maintenant développer un langage qui permettra la construction et l'étude des prédicats.

3.2

Les éléments du langage

Pour l'étude des prédicats nous allons définir un langage formel $\mathcal{L}_1$ dont l'alphabet A_1 est composé des éléments suivants :

- Un ensemble **V**, au plus dénombrable, des variables qui seront notées $x, y, \ldots$ ou $X, Y, \ldots$.
- Un ensemble Ξ, au plus dénombrable, des constantes qui seront notées $a, b, \ldots$.
- Un ensemble **F** de fonctions $f : \mathbf{V} \times \Xi \to \mathbf{V} \cup \Xi$ d'arité quelconque [1]. On appelle ces fonctions des *foncteurs* que l'on notera $f, g, \ldots$ ou $F, G, \ldots$.
- Un ensemble **P** de fonctions d'arité quelconque $f : \mathbf{V} \times \Xi \to \{Vrai, Faux\}$ qui seront appelées *prédicats* et qui seront notés par $p, q, \ldots$ ou $P, Q, \ldots$. Un prédicat particulier est la relation d'égalité qui sera notée soit de façon fonctionnelle $eg(x, y)$, soit, lorsqu'il n'y a pas risque de confusion, $x = y$ [2].
- Un ensemble **L** des connecteurs et quantificateurs :
  - Connecteur logique unaire : la négation $\neg$
  - Connecteurs propositionnels binaires :
    - Conjonction : $\wedge$
    - Disjonction : $\vee$
    - Implication : $\to$
    - Équivalence (ou double implication) : $\leftrightarrow$
  - Quantificateurs :
    - Quantificateur existentiel : $\exists$
    - Quantificateur universel : $\forall$
- Les séparateurs (symboles auxiliaires) : $($, $)$, $[$, $]$.

Les séparateurs ne font pas partie, à proprement parler, du langage. Leur présence permet de faciliter la lecture des formules.

Les éléments du langage déterminent un alphabet $A_1 = \{\mathbf{V}, \Xi, \mathbf{F}, \mathbf{P}, \mathbf{L}\}$.

Les termes

La brique élémentaire du calcul des prédicats est le *terme*.

DÉFINITION 3.0.3 *Un* terme *est défini de façon itérative comme suit :*

1. L'arité d'une fonction est le nombre d'arguments de la fonction. Remarquons qu'une constate est une fonction d'arité zéro.

2. Il faut clairement distinguer entre *égalité* "=" et *équivalence* "≡". La égalité entre deux termes est un prédicat qui indique si oui ou non les deux termes ont la même valeur. L'équivalence entre deux fbf indique que les deux fbf ont la même table de vérité. La question peut se poser si deux termes égaux sont équivalents. La réponse est non. Par exemple nous avons "10= 3+7" mais par rapport au prédicat *nombre pair* il n'y a pas d'équivalence. De même deux fbf équivalentes ne sont pas égales. Par exemple la fbf $a \vee \neg a$ est équivalente à la fbf $a \vee (a \to b)$ car toutes les deux sont des tautologies, mais elles ne sont pas égales

- *une constante est un terme ;*
- *une variable est un terme ;*
- *si f est un foncteur d'arité n et $t_1, t_2, \ldots, t_n$ sont des termes, alors $f(t_1, t_2, \ldots, t_n)$ est un terme.*

Tout terme est obtenu par application des règles précédentes un nombre fini de fois. L'ensemble de termes sera noté par $\mathbb{T}$.

On peut construire, à partir des termes, des formules bien formées (fbf).

DÉFINITION 3.0.4 *Soit $\mathbb{T}$ l'ensemble des termes sur un alphabet A_1. L'ensemble $\mathbb{F}$ des* formules bien formées *(par rapport à A_1) est le plus petit ensemble tel que :*

- *Si p est un prédicat d'arité n et $t_1, t_2, \ldots, t_n$ sont des termes, alors $p(t_1, t_2, \ldots, t_n) \in \mathbb{F}$.*
- *Si F , $G \in \mathbb{F}$ alors les constructions suivantes :*
  - $\neg F$
  - $F \vee G$, $F \wedge G$
  - $F \rightarrow G$, $F \leftrightarrow G$

  sont aussi des fbf.
- *Si $F \in \mathbb{F}$ et X est une variable, alors $(\forall X) F \in \mathbb{F}$ et $(\exists X) F \in \mathbb{F}$.*

L'équivalent de l'atome est donné par la définition suivante :

DÉFINITION 3.0.5 *Si p est un prédicat d'arité n et $t_1, t_2, \ldots, t_n$ sont des termes, alors $p(t_1, t_2, \ldots, t_n)$ est une* formule atomique.

DÉFINITION 3.0.6 *Le triplet*

$$\mathcal{L}_1 = \{A_1, \mathbb{T}, \mathbb{F}\}$$

est le langage d'ordre un *ou le* langage du calcul des prédicats

Par construction $\mathbb{F}$ est un ensemble dénombrable, défini itérativement.

Portée des quantificateurs

Les quantificateurs ont une *portée*.

DÉFINITION 3.0.7 *La* portée *d'un quantificateur dans une formule est la partie de la formule qui se trouve sous l'influence du quantificateur.*
Une variable d'une formule qui est sous la portée d'un quantificateur de la même variable est appélée variable liée *(ou* bornée*). Sinon elle est une* variable libre.
Une formule qui n'a pas des variables libres s'appelle formule close. *Une formule qui n'a pas des variables s'appelle* formule filtrée.
Une formule sans quantificateurs s'appelle formule ouverte.

3.3

Substitution

Dans une une fbf, des variables peuvent être remplacées (substituées) par des termes ou, encore, par des fbf.

DÉFINITION 3.0.8 *Une* substitution *est un ensemble fini de couples de la forme* $(x_1/A_1, x_2/A_2, \ldots, x_n/A_n)$ *où chaque* A_i *est soit un terme, soit une fbf et chaque* x_i *une variable telle que* $x_i \neq A_i$ *et* $A_i \neq A_j$ *si* $x_i \neq x_j$.

La substitution vide sera notée par ε.

La substitution peut être considérée comme une application de l'ensemble des variables V dans celui des termes et des fbf. Ainsi soit l'ensemble E qui contient les termes ou fbf $A_1, A_2, \ldots, A_n$. La substitution $\sigma = (x_1/A_1, x_2/A_2, \ldots, x_n/A_n)$ permet d'avoir la fbf $E\sigma$ qui est le résultat de l'application σ à E et qui est obtenu en remplaçant chaque occurrence dans E de x_i par A_i , $i = 1, \ldots, n$. $E\sigma$ est appelé une *exemplification* (*instance*) de E.

DÉFINITION 3.0.9 *Soient* $\sigma = (x_1/A_1, x_2/A_2, \ldots, x_n/A_n)$ *et* $\tau = (y_1/B_1, y_2/B_2, \ldots, y_m/B_m)$ *deux substitutions. La composition* $\sigma\tau$ *de* σ *et* τ *est obtenue à partir de l'ensemble* $\{x_1/A_1\tau, x_2/A_2\tau, \ldots, x_n/A_n\tau, y_1/B_1, y_2/B_2, \ldots, y_m/B_m\}$ *en supprimant*

– *tous les* $x_i/A_i\tau$ *pour lesquels on a* $x_i = A_i\tau$, $i = 1, \ldots, n$
– *tous les* y_j/B_j *pour lesquels on a* $y_j \in \{x_1, x_2, \ldots, x_n\}$, $j = 1, \ldots, m$.

EXEMPLE 3.0.1 $\sigma = (x/Z, y/W)$, $\tau = (x/a, Z/b, W/y) \Rightarrow \sigma\tau = (x/b, Z/b, W/y)$

La composition des substitutions n'est pas en général commutative.

Les propriétés des substitution sont les suivantes :

PROPRIÉTÉ 3.0.1 $E(\sigma\tau) = (E\sigma)\tau$

PROPRIÉTÉ 3.0.2 $(\rho\sigma)\tau = \rho(\sigma\tau)$

PROPRIÉTÉ 3.0.3 $\varepsilon\theta = \theta\varepsilon = \theta$

PROPRIÉTÉ 3.0.4 *Si* $\models E$, *alors* $\models E\sigma$

3.4

Interprétation sémantique - Modèles

La première tâche du calcul des prédicats est de donner un sens aux fbf, c'est-à-dire d'établir la valeur de vérité de ces formules.

Considérons un langage $\mathcal{L}$ du 1er ordre et un univers du discours $\mathcal{U}$. Nous allons essayer de mettre en association les éléments du langage d'une part et les éléments de l'univers du discours, d'autre part. Cette correspondance entre éléments du langage et éléments de l'univers du discours constitue ce qu'on appelle une *interprétation* I du langage ou, encore, *structure*. On peut envisager que l'interprétation ne couvre pas la totalité de l'univers du discours $\mathcal{U}$ mais seulement un domaine $\mathbb{D}(I) \subset \mathcal{U}$. Notons que cette correspondance n'est pas une application au sens habituel du terme mais plutôt un ensemble d'applications, chacune de ses applications étant spécifique à un type d'éléments de $\mathcal{L}$. On a donc la

DÉFINITION 3.0.10 (Interprétation) *Une* interprétation I *sur un langage* $\mathcal{L}$ *est un domaine non vide* $\mathbb{D}(I)$ *(qui parfois est noté* $|\ I\ |$*), appelé* domaine de l'interprétation*, et une application qui associe :*

- *chaque constante* $a \in \Xi$ *avec un élément* $a_I \in \mathbb{D}(I)$ *;*
- *chaque foncteur* $f \in \mathbf{F}$ *d'arité* n *avec une fonction* $f_I : \mathbb{D}(I)^n \to \mathbb{D}(I)$ *;*
- *chaque prédicat* $p \in \mathbf{P}$ *d'arité* n *avec une relation* $p_I \subseteq \mathbb{D}(I)^n$.

Grâce à cette définition de l'interprétation, on peut attribuer des valeurs de vérité à une fbf en utilisant les valeurs de vérité de ses composantes qui sont soit des fbf à leur tour, soit des termes.

Il y a tout de même un problème : les termes peuvent contenir des variables. Or comme calculer la valeur de verité d'une variable ? En effet, pour pouvoir interpréter une variable x il faudrait savoir quel objet désigne exactement x. Mais dans ce cas x ne serait plus une variable. On s'en sort de cette situation embarrassante par un artifice du type suivant : les variables du langage sont interprétées comme étant des éléments variables de l'univers du discours ! Et pour formaliser cette belle interprétation on utilise ce qu'on appelle l'*assignation* des variables par rapport à une interprétation, qui est une application $\overline{\varphi}_I : V \to \mathbb{D}(I)$ où V l'ensemble des variables du langage $\mathcal{L}$ c'est-à-dire formellement :

DÉFINITION 3.0.11 (Sémantique des variables) *On appelle assignation d'un ensemble des variables* $W \subset V$ *relativement à une interprétation* I *, une application* $\overline{\varphi}_I : W \to \mathbb{D}(I)$.

EXEMPLE 3.0.2 *Considérons un langage* $\mathcal{L}$ *avec une constante* a*, une variable* x*, un foncteur unaire* f *et un prédicat binaire* p*. On peut envisager la formule* $p(f(a), a)$*. La transformation de cette formule par l'interprétation* I *serait* $p_I(f_I(a_I), a_I)$*. Mais qu'en est-il de la transformation de la formule* $p(f(x), a)$ *? Si on procède à l'assignation* x_I *de la variable* x*, où* x_I *une variable indiquant un élément quelconque de l'univers du discours, alors on peut écrire pour l'interprétation :* $p_I(f_I(x_I), a_I)$*. Bien sûr si on pose* $x = a$*, alors les deux interprétations sont identiques.*

L'assignation des variables nous permet maintenant de définir la signification d'un terme.

DÉFINITION 3.0.12 (Sémantique des termes) *La* signification [3] φ_I *d'un ensemble des termes* $\mathbb{T}$ *relativement à une interprétation* I *, est définie comme suit :*

- *si* $t \in \mathbb{T}$ *est une constante, alors* $\varphi_I(t) = t_I$ *;*
- *si* $t \in \mathbb{T}$ *est une variable, alors* $\varphi_I(t) = \overline{\varphi}_I(t)$ *;*

3. La terminologie n'est pas stabilisée en français. Certains auteurs utilisent le terme *assignation* pour signification, tandis que d'autres auteurs préfèrent parler d'*interprétation*. Il est curieux de constater que personne n'utilise la traduction du terme anglais *meaning* (= signification) qui est, à mon avis, très éclairante ici.

– *si* $t \in \mathbb{T}$ *est un terme de la forme* $f(t_1, t_2, \ldots, t_n)$*, alors* $\varphi_I(t) = f_I(\varphi_I(t_1), \varphi_I(t_2), \ldots, \varphi_I(t_n))$.

On peut maintenant, grâce aux deux définitions précédentes, calculer la valeur de vérité d'une fbf. Formellement nous avons la définition suivante :

DÉFINITION 3.0.13 (Sémantique des fbf) *La* valeur de vérité de la signification d'une fbf F relativement à une interprétation I [4] *, est définie comme suit :*

– *Si la fbf est de la forme* $F = p(t_1, t_2, \ldots, t_n)$ *, alors* $I(F)$ *est égale à la valeur de vérité de* $p_I(\varphi_I(t_1), \varphi_I(t_2), \ldots, \varphi_I(t_n))$.
– *Si la fbf* F *a une des formes* $\neg G$ *,* $G \vee H$ *,* $G \wedge H$ *,* $G \rightarrow H$ *,* $G \leftrightarrow H$*, alors* $I(F)$ *est égale à la valeur de vérité de la forme correspondante.*
– *Si la fbf* F *est de la forme* $\forall x G(x, y, z, \ldots)$*, alors* $I(F) = 1$ *si* $\forall \sigma = (x/a)$ *avec* $a \in \mathbb{D}(I)$ *nous avons* $I(G\sigma) = 1$ *(vraie). Sinon* $I(F) = 0$ *(fausse).*
– *Si la fbf* F *est de la forme* $\exists x G(x, y, z, \ldots)$*, alors* $I(F) = 1$ *si* $\exists \sigma = (x/a)$ *avec* $a \in \mathbb{D}(I)$ *nous avons* $I(G\sigma) = 1$ *(vraie). Sinon* $I(F) = 0$ *(fausse).*

EXEMPLE 3.0.3 *Si* $\mathbb{D}(I)$ *est l'ensemble des nombres réels avec la relation d'égalité "=" qui correspond au prèdicat* eg *du langage* $\mathcal{L}_1$*, alors la formule atomique* $F = \mathrm{eg}(a, b)$ *a comme valeur de vérité* $I(F)$*, la valeur de vérité de la relation* $a = b$.

De cette définition on déduit que la valeur de vérité d'une fbf close, par rapport à une interprétation, dépend seulement de cette interprétation et elle est indépendante de l'assignation des variables.

Satisfiabilité – Validité

DÉFINITION 3.0.14 *Une fbf* F *est* satisfiable *ou* sémantiquement consistante *s'il existe une interprétation* I *telle que la valeur de vérité de* F *par rapport* I *est ègale à* 1*. L'interprétation est alors un* modèle *de* F *et l'on note par* $I \models F$.
Une fbf qui ne possède pas de modèle est appelée sémantiquement inconsistante *ou* insatisfiable.

DÉFINITION 3.0.15 *Une fbf* F *qui est vraie pour toute interprétation est appelée* formule valide *et sera notée par* $\models F$.

Examinons maintenant la notion de la conséquence sémantique.

DÉFINITION 3.0.16 *Soit la fbf close* B *et un ensemble des fbf closes* $A = \{A_1, A_2, \ldots, A_n\}$ [5]. B *est une* conséquence sémantique *des* $A_1, A_2, \ldots, A_n$ *, que l'on note par* $A \models B$ *ou* $A_1, A_2, \ldots, A_n \models B$*, si pour toute interprétation* I *telle que* $I(A_i) = 1 \forall i = 1, \ldots, n$*, on a* $I(B) = 1$.

4. ou, de façon plus succincte, la *valeur de vérité de* F *relativement à* I.
5. Il faut prendre l'habitude d'interpréter un ensemble $\{A_1, A_2, \ldots, A_n\}$ de fbf comme étant une conjonction $A_1 \wedge A_2 \wedge \ldots \wedge A_n$

En d'autres termes, on peut dire que B doit être vérifiée pour tout modèle de A.

En général il est difficile de vérifier si une fbf close B est une conséquence logique d'un ensemble de fbf closes A, car il faut vérifier B pour tout modèle de A. Une autre façon de démontrer que $A \models B$ est de montrer que $\neg B$ est fausse pour tout modèle de A ou, ce qui revient au même, que l'ensemble de fbf $A \cup \{\neg B\}$ est insatisfiable, c'est-à-dire n'a pas de modèle. Ce procédé est plus facile parce qu'il suffit de trouver un modèle de A qui n'est pas un modèle pour B. Formellement nous avons :

THÉORÈME 3.0.2 (de l'insatisfiabilité) *Soient A ensemble de fbf closes et B fbf close. Alors $A \models B$ si et seulement si $A \cup \{\neg B\}$ est insatisfiable.*

Une notion importante pour la sémantique des fbf est celle de l'équivalence :

DÉFINITION 3.0.17 *Deux fbf F et G sont* logiquement équivalentes *si et seulement si elles ont la même valeur de vérité pour toute interprétation I.*

3.5 Examen des quantificateurs

Considérons un prédicat p quelconque, que nous prendrons pour la circonstance et sans perte de généralité, d'arité un. La fbf $\forall x p(x)$ a comme valeur de vérité pour toute interprétation d'un domaine quelconque $\mathbb{D}(I)$ la valeur $\min\left\{p_I / x_I \in \mathbb{D}(I)\right\}$, c'est-à-dire la valuation d'une fbf universellement quantifiée sur la variable x est la valuation minimale de cette formule pour toutes les interprétations x_I de x appliquées à l'interprétation p_I du prédicat p. Dans la mesure où $\min\left\{p_I / x_I \in \mathbb{D}(I)\right\} = \bigwedge\left\{p_I / x_I \in \mathbb{D}(I)\right\}$ on peut dire que le quantificateur universel $\forall$ est une généralisation du connecteur $\bigwedge$. De même la valuation de la fbf $\exists x p(x)$est donnée par la valeur $\max\left\{p_I / x_I \in \mathbb{D}(I)\right\} = \bigvee\left\{p_I / x_I \in \mathbb{D}(I)\right\}$, on peut dire que le quantificateur existentiel $\exists$ est une généralisation du connecteur $\bigvee$.

(1) $\neg \forall x A(x) \equiv \exists x \neg A(x)$

(2) $\forall x A(x) \equiv \neg \exists x \neg A(x)$

(3) $\neg \exists x A(x) \equiv \forall x \neg A(x)$

(4) $\exists x A(x) \equiv \neg \forall x \neg A(x)$

Pour la distributivité des quantificateurs nous avons les relations suivantes :

(1) $\models \forall x(\varphi \wedge \psi) \leftrightarrow \forall x \varphi \wedge \forall x \psi$

(2) $\models \exists x(\varphi \vee \psi) \leftrightarrow \exists x \varphi \vee \exists x \psi$

(3) $\models \forall x(\varphi(x) \vee \psi) \leftrightarrow \forall x \varphi(x) \vee \psi$ si x n'est pas une variable libre de ψ.

(4) $\models \exists x(\varphi(x) \wedge \psi) \leftrightarrow \exists x \varphi(x) \wedge \psi$ si x n'est pas une variable libre de ψ.

Attention : Les formules suivantes :
– $\forall x(\varphi(x) \vee \psi(x)) \rightarrow \forall x \varphi(x) \vee \forall x \psi(x)$

– $\exists x \varphi(x) \land \exists x \psi(x) \rightarrow \exists x(\varphi(x) \land \psi(x))$

ne sont pas vraies.

3.6

Exercices de Logique Computationnelle

EXERCICE 3.1 *On se place dans l'ensemble des nombres réels. Écrire, en utilisant le langage des prédicats, les propositions suivantes :*

(1) Pour tout réel il existe un autre réel qui est plus grand.

(2) Il existe un réel qui est plus grand que tout autre réel.

(3) Chaque réel positif est un carré.

(4) Si un réel est plus petit qu'un autre réel, alors il existe un troisième réel, différent de deux précédents, et qui est entre les deux.

EXERCICE 3.2 *Donner l'équivalent en français de deux fbf suivantes :*

(1) $\exists X\,(p(X) \land (\forall Y\, p(Y) \rightarrow X = Y))$

(2) $(\exists X\, p(X)) \land (\forall X \forall Y\, p(X) \land p(Y) \rightarrow X = Y)$

EXERCICE 3.3 *On se place dans le cadre des nombres entiers et on suppose que nous disposons des opérations (foncteurs) d'addition (+) et de multiplication (×). Donner, en langage des prédicats, la défintion des prédicats suivants :*

(1) pair (x) = x *est un nombre pair.*

(2) div (x,y) = x *divise* y*, c'est-à-dire que* y/x *est un nombre entier.*

(3) premier (x) = x *est un nombre premier.*

(4) $x \equiv y$ *modulo* n*.*

EXERCICE 3.4 *Considérons un langage* $\mathcal{L}$ *et un prédicat unaire* $p \in \mathcal{L}$*. Supposons que le domaine de définition de ce prédicat est l'ensemble* $D = \{1,2\}$*.*

(1) Établissez pour le prédicat p *les quatre interprétations* I_i *;* $i = 1,\ldots,4$ *possibles et distinctes en utilisant comme domaine d'interprétation l'ensemble* D*.*

(2) Trouver un modèle pour la fbf

$$p(x) \lor \forall y\,(p(y) \rightarrow q)$$

EXERCICE 3.5 *Considérons un ensemble des termes* $\mathbb{T}$ *qui contient la constante zéro (l'élément neutre de l'addition), le foncteur unaire* s *(l'élément successeur) et le foncteur binaire plus (l'addition de deux valeurs). On cherche à établir une signification relativement à une interprétation* I *dont son domaine* $\mathbb{D}(I)$ *est l'ensemble des entiers non négatifs muni de l'opération de l'addition* $+$*.*

(1) Quelle doit-être, selon vous, l'interprétation des termes $zéro$ *,* s *et* $plus$ *?*

(2) Calculer la signification du terme $plus(s(zero), X)$ *où* X *une variable avec assignation*

$$\overline{\varphi}_I(X) = \begin{cases} 0 & \text{si } X \neq \text{ nombre entier} \\ \\ |X| & \text{si } X = \text{ nombre entier} \end{cases}$$

EXERCICE 3.6 *Considérons l'univers du discours* $\{\bigcirc, *\}$ *avec*

- *les constantes* a, b
- *les variables* x, y, z
- *la relation binaire* f

Considérons une interprétation I *telle que*

- *l'interprétation des constantes est :* $a_I = \bigcirc, b_I = *$
- *l'assignation des variables est :* $\varphi_I(x) = \bigcirc, \varphi_I(y) = \bigcirc, \varphi_I(z) = *$
- *l'interprétation de la relation est :* $f_I = \{\langle \bigcirc, * \rangle, \langle *, * \rangle\}$

Calculer la valeur de vérité des termes suivants :

(1) $\neg f(b, a)$

(2) $f(a, b) \wedge f(b, a)$

(3) $f(a, b) \vee f(b, a)$

(4) $f(a, b) \rightarrow f(b, a)$

(5) $f(a, b) \leftrightarrow \neg f(b, a)$

EXERCICE 3.7 *Soit* $\mathcal{L}$ *un langage du 1er ordre. Considérons la formule*

$$\forall x \exists y p(x, y) \rightarrow \exists y \forall x p(x, y)$$

Est-ce que cette formule peut être satisfaite pour n'importe quelle interprétation de $p(x, y)$ *?*

3.7

Repésentation des nombres naturels en Prolog

Considérons l'univers de discours $\mathcal{U}$ composé de l'ensemble de nombres naturels et de l'opération de l'addition, munie de son élément neutre 0. Si on veut construire un langage $\mathcal{L}_0$ dont on chercherait une interprétation dans $\mathcal{U}$, on doit avoir un foncteur équivalent à l'opération de l'addition, ainsi que l'élément neutre. Plaçons-nous dans le cadre d'un Prolog pur, c'est-à-dire d'un Prolog qui ne contient pas des formes arithmétiques. Convenons d'appeler `add/3` l'opération d'addition dans $\mathcal{U}$[6]. Notons aussi par `zero` l'élément neutre de `add` qui est, par ailleurs, une des constantes du langage $\mathcal{L}_0$. Il nous faut aussi un prédicat, que nous appellerons `nat/1` pour caractériser les nombres naturels. On pourra ainsi écrire :

```
nat(zero).
```

pour indiquer que `zero` est un nombre naturel. La question qui se pose maintenant concerne les autres nombres naturels, à savoir de quelle manière nous allons représenter ces nombres.

6. `add/3` signifie que le foncteur `add` est d'arité 3, c'est-à-dire que le nombre de ses arguments est 3.

Depuis Peano, au moins, on sait que nous pouvons construire l'ensemble des nombres naturels à partir de 0 et de l'opérateur de succession $s(X) = X + 1$, où X est un nombre naturel. On peut utiliser cette même technique pour le langage $\mathcal{L}_0$. On se dote donc d'un foncteur `s/1` qui représente l'opérateur de succession dans $\mathcal{L}_0$ et, par conséquent, le programme complet de caractérisation des nombres naturels s'écrit :

```
nat(zero).
nat(s(X)) :- nat(X).
```

On peut, par exemple, poser la question

```
?- nat(s(s(s(s(zero))))).
```

et on aura comme réponse `yes`. Mais le plus rigolo c'est quand on pose la question

```
?- nat(X).
```

Dans ce cas on recupère comme réponse

```
X = zero ;
X = s(zero) ;
X = s(s(zero)) ;
X = s(s(s(zero))) ;
X = s(s(s(s(zero)))) ;
X = s(s(s(s(s(zero)))));
. . . . . . . .
```

c'est-à-dire l'ensemble des nombres naturels.

Avant d'avancer en programmation, essayons de voir, sous l'aspect logique des prédicats, ce que nous venons de construire. Nous avons une logique du 1er ordre $\mathcal{L}_0$ dont les termes sont $\mathbb{T} = \{\texttt{zero}, \texttt{s/1}, \texttt{add/3}, \texttt{X}, \texttt{Y}\}$ et nous avons aussi le prédicat `nat/1`. L'interprétation I que nous avons adoptée conduit à la signification ϕ_I suivante des termes : (cf. Définition 4.3.3, p.52 du poly de logique) :

- $\texttt{zero} \rightarrow \phi_I(\texttt{zero}) = \texttt{zero}_I = 0$
- s(X) $\rightarrow \phi_I\left(\texttt{s}(\overline{\phi}_I(\texttt{X}))\right) = \texttt{s}_I(\overline{\phi}_I(\texttt{X})) = \overline{\phi}_I(\texttt{X}) + 1$
- add(X,Y,Z) $\rightarrow \phi_I\left(\texttt{add}(\overline{\phi}_I(\texttt{X}), \overline{\phi}_I(\texttt{Y}), \overline{\phi}_I(\texttt{Z}))\right) = \texttt{add}_I\left(\texttt{add}(\overline{\phi}_I(\texttt{X}), \overline{\phi}_I(\texttt{Y}), \overline{\phi}_I(\texttt{Z})\right)$ d'où $\overline{\phi}_I(\texttt{X}) + \overline{\phi}_I(\texttt{Y}) = \overline{\phi}_I(\texttt{Z})$.

Pour écrire le programme de l'addition de deux naturels en Prolog nous allons utiliser les deux remarques suivantes :

(1) Si on ajoute 0 à une valeur X, le résultat est X. En Prolog on a
```
add(zero,X,X).
```
(2) Si on a X+Y=Z, alors (X+1) + Y = (Z+1). En Prolog on a
```
add(s(X),Y,s(Z)) :- add(X,Y,Z).
```

On peut, par exemple, faire l'ddition

```
?- add(s(s(s(s(s(zero))))), s(s(zero)), Z).
```

et avoir comme réponse

```
Z = s(s(s(s(s(s(s(zero)))))))
```

Mais on peut aussi faire la soustraction entre le troisième terme et le premier

```
?- add(s(s(s(s(s(zero))))), Y, s(s(s(s(s(s(s(zero))))))).
```

avec réponse

```
Y = s(s(zero))
```

et aussi entre le troisième terme et le deuxième

```
?- add(X, s(s(zero)), s(s(s(s(s(s(s(zero))))))).
```

avec réponse

```
X = s(s(s(s(s(zero)))))\\
```

3.8 Exercices de Prolog

EXERCICE 3.8 *Écrire le programme Prolog* `mult(X,Y,Z)` *qui calcule le produit de deux naturels* `X`, `Y` *et stocke le résultat dans* `Z`.

Exemple : `mult(s(s(zero)), s(s(s(zero))), Z)` donnera comme résultat `Z = s(s(s(s(s(s(zero))))))`.

EXERCICE 3.9 *En utilisant le programme précédent, calculer le résultat de la division de deux naturels* `X`, `Y`. *On fait l'hypothèse que* `X` *divise* `Y`, *c'est-à-dire que* Y/X *est un entier.*

EXERCICE 3.10 *Écrire le programme Prolog* `egal(X,Y)` *qui teste si les variables* `X`, `Y` *ont la même valeur.*

EXERCICE 3.11 *Écrire le programme Prolog* `pair(X)` *qui teste si la valeur de* `X` *est paire. On fera l'hypothèse que* `zero` *est pair.*

EXERCICE 3.12 *Même chose pour le prédicat* `impair(X)`.

EXERCICE 3.13 *Écrire le programme Prolog* `ppe(X,Y)` *qui teste si* `X` *est plus petit ou égal à* `Y`.