Logique des propositions (2)

F. Alexandre

École Supérieure d'Informatique et Applications de Lorraine

January 21, 2009

- Algorithme de mise sous forme clausale[1]
- Système formel basé sur les clauses et la résolution
- Complétude de la résolution
- Modélisation de problème logique

[1] MATHÉMATIQUES DISCRÈTES. Automates, langages, logique et décidabilité. Pierre Marchand. DUNOD. Pierre.Marchand@esial.uhp-nancy.fr

- algorithme utilisant la méthode des attributs
- les formules de $Prop(P)$ définies à partir de la grammaire algébrique suivante : $(N,\ T,\ \rightarrow,\ Y)$
  - $N = \{Y,\ X\}$,
  - $T = P \cup \{\mathcal{F},\ \mathcal{V},\ \neg,\ \Rightarrow,\ \wedge,\ \vee,\ (,\ )\}$
  - $Y \rightarrow X$
    $X \rightarrow \mathcal{V}\ |\mathcal{F}\ |a\ |\neg X\ |X \Rightarrow X\ |X \wedge X\ |X \vee X\ |(X)$
    où $a \in P$
- les autres connecteurs logiques s'expriment en utilisant $\neg$ et $\vee$

- Spécification :
  - Données : α une formule de $Prop(P)$
  - Résultat : $C(\alpha)$ un ensemble de clauses dont la conjonction est équivalente à α
- mise sous forme d'un arbre abstrait de la formule α
- attributs considérés :
  - ν attribut qui compte le nombre de négations qui porte sur chaque nœud de l'arbre abstrait de la formule. Comme deux négations "s'annulent" (i.e. $\neg\neg p \Leftrightarrow p$) il suffit de compter le nombre de négations "modulo" 2. Cet attribut est hérité, c'est-à-dire qu'on propage sa valeur dans l'arbre abstrait en "descendant" dans l'arbre.
  - C attribut qui calcule l'ensemble de clauses associées à chaque nœud de l'arbre. Cet attribut est synthétisé, c'est-à-dire qu'on propage sa valeur en remontant dans l'arbre en partant des feuilles jusqu'à la racine.

- Définition de l'attribut ν

Règle	$Y \rightarrow X$	$X_1 \rightarrow \neg X_2$	$X_1 \rightarrow X_2 \vee X_3$
Calcul de ν	$\nu(X) = 0$	$\nu(X_2) = 1 - \nu(X_1)$	$\nu(X_2) = \nu(X_1)$ $\nu(X_3) = \nu(X_1)$
Règle	$X_1 \rightarrow X_2 \wedge X_3$	$X_1 \rightarrow X_2 \Rightarrow X_3$	$X_1 \rightarrow (X_2)$
Calcul de ν	$\nu(X_2) = \nu(X_1)$ $\nu(X_3) = \nu(X_1)$	$\nu(X_2) = 1 - \nu(X_1)$ $\nu(X_3) = \nu(X_1)$	$\nu(X_2) = \nu(X_1)$

- les règles de la forme $X \rightarrow X$ *op* X sont écrites sous la forme $X_1 \rightarrow X_2$ *op* X_3, on peut ainsi définir comment se propage l'attribut ν sur les différentes branches de l'arbre

- Définition de l'attribut C

Règle	$Y \rightarrow X$	$X_1 \rightarrow X_2 \wedge X_3$	$X_1 \rightarrow X_2\ op\ X_3$ $op = \vee$ ou $\Rightarrow$
Calcul de C	$C(Y) = C(X)$	$C(X_1) =$ si $\nu(X_1) = 0$ alors $C(X_2) \cup C(X_3)$ sinon $C(X_2) \otimes C(X_3)$	$C(X_1) =$ si $\nu(X_1) = 0$ alors $C(X_2) \otimes C(X_3)$ sinon $C(X_2) \cup C(X_3)$
Règle	$X_1 \rightarrow \neg X_2$ $X_1 \rightarrow (X_2)$	$X \rightarrow \mathcal{V}$	$X \rightarrow \mathcal{F}$
Calcul de C	$C(X_1) = C(X_2)$	$C(X) =$ si $\nu(X) = 0$ alors $\emptyset$ sinon $\{\square\}$	$C(X) =$ si $\nu(X) = 0$ alors $\{\square\}$ sinon $\emptyset$
Règle	$X \rightarrow a$		
Calcul de C	$C(X) =$ si $\nu(X) = 0$ alors $\{a\}$ sinon $\{\neg a\}$		

- La construction de l'ensemble des clauses se fait en utilisant la relation de subsomption, on élimine les clauses qui sont subsumées par une autre clause de l'ensemble.
- Mettre sous forme clausale la formule suivante :
  - $\alpha_1 = (a \wedge (b \vee c)) \Leftrightarrow ((a \wedge b) \vee (a \wedge c))$
  - $\alpha_2 = (p \Rightarrow r) \Rightarrow ((q \Rightarrow r) \Rightarrow ((p \vee q) \Rightarrow r))$

Proposition

Soit α une formule de $Prop(P)$, α est une tautologie si et seulement l'ensemble $C(\alpha)$ obtenue par l'algorithme de mise sous forme clausale est égal à $\emptyset$

Exemple : montrer que la formule suivante est une tautologie.

- Objectif : montrer que $\mathcal{A} \models \alpha$
- Ce qui est équivalent à montrer que $\mathcal{A} \cup \{\neg\alpha\}$ est contradictoire
- En considérant les formes clausales si $\mathcal{A} = \{\beta_1, \ldots, \beta_n\}$, on veut montrer que l'ensemble de clauses $(\bigcup_{i=1}^{n} C(\beta_i)) \cup C(\neg\alpha)$ est contraditoire
- Utiliser un système formel utilisant des clauses : règle de résolution

Definition

Soit P un ensemble de variables propositionnelles et $\mathcal{C}(P)$ l'ensemble des clauses construites sur P. Le système formel $(\mathcal{C}(P),\ \emptyset,\ \mathcal{R})$

où $\mathcal{R} = \{\ \dfrac{c_1 \lor a \lor c_2, \quad c_3 \lor \neg a \lor c_4}{c_1 \lor c_2 \lor c_3 \lor c_4}(resolution)\ \}$

est un système basé sur la règle de résolution (c_1, c_2, c_3, c_4 sont des clauses a est une variable propositionnelle).

Proposition

La règle de résolution est valide.

Démonstration: il faut montrer que $\{c_1 \lor a \lor c_2, \quad c_3 \lor \neg a \lor c_4\} \models c_1 \lor c_2 \lor c_3 \lor c_4$
Soit δ telle que $[c_1 \lor a \lor c_2](\delta) = 1$ et $[c_3 \lor \neg a \lor c_4](\delta) = 1$
on a $[c_1 \lor c_2](\delta) = 1$ ou $\delta(a) = 1$

- si $[c_1 \lor c_2](\delta) = 1$ alors $[c_1 \lor c_2 \lor c_3 \lor c_4](\delta) = 1$
- si $\delta(a) = 1$ alors $[\neg a](\delta) = 0$ comme $[c_3 \lor \neg a \lor c_4](\delta) = 1$
  on a $[c_3 \lor c_4](\delta) = 1$ et donc $[c_1 \lor c_2 \lor c_3 \lor c_4](\delta) = 1$

Théorème de Robinson

Soit C un ensemble de clauses, C est contradictoire si et seulement s'il existe une démonstration de $\square$ avec hypothèses dans C. On a donc les équivalences suivantes :

$$\begin{gathered} \mathcal{A} \models \alpha \\ \textit{ssi} \\ \mathcal{A} \cup \{\neg\alpha\} \textit{ est contradictoire} \\ \textit{ssi} \\ C(\mathcal{A}) \cup C(\neg\alpha) \textit{ est contradictoire} \\ \textit{ssi} \\ C(\mathcal{A}) \cup C(\neg\alpha) \vdash_{\textit{Resolution}} \square \end{gathered}$$

Démonstration

Ce théorème établit la validité et la complétude du système formel basé sur la résolution. La complétude exprime que si un ensemble de clauses est contradictoire il est possible de trouver une démonstration menant à la clause vide $\square$. (voir démonstration livre PM)

On a le fait suivant :
si l’assemblée nationale refuse de voter la loi alors la grève ne s’arrêtera pas à moins qu’elle ne dure depuis plus d’un an et que le président de la firme démissionne.
Peut-on déduire que la grève n’arrêtera pas si l’assemblée nationale refuse de voter la loi et si la grève vient juste de commencer ? Déterminons les variables propositionnelles suivantes p, q, r et s suivantes :

- p : l’assemblée refuse de voter la loi
- q : la grève est finie
- r : le président de la firme démissionne
- s : la grève dure depuis plus d’un an

On exprime le fait par $p \Rightarrow (\neg q \vee (s \wedge r))$
peut-on déduire que $(p \wedge \neg s) \Rightarrow \neg q$,

c'est-à-dire que $\{p \Rightarrow (\neg q \vee (s \wedge r))\} \models (p \wedge \neg s) \Rightarrow \neg q$

c'est-à-dire $\{p \Rightarrow (\neg q \vee (s \wedge r)),\ \neg((p \wedge \neg s) \rightarrow \neg q)\}$ est contradictoire

- mise sous forme clausale des deux formules (en utilisant l'algorithme de mise sous forme clausale)
  $C(p \rightarrow (\neg q \vee (s \wedge r))) = \{\neg p \vee \neg q \vee r,\quad \neg p \vee \neg q \vee s\}$
  $C(\neg((p \wedge \neg s) \rightarrow \neg q)) = \{p,\ \neg s,\ q\}$
- On montre en utilisant la règle de résolution que l'ensemble de clauses $\{\neg p \vee \neg q \vee r,\quad \neg p \vee \neg q \vee s,\ p,\ \neg s,\ q\}$ est contradictoire.
- On a

$$\frac{\neg p \vee \neg q \vee s, \quad p}{\neg q \vee s}(res)$$

$$\frac{\neg q \vee s, \quad q}{s}(res)$$

$$\frac{s, \quad \neg s}{\square}(res)$$

Logique du premier ordre (1)

- logique des propositions : notions entrevues (sémantique, systèmes formels, mise sous forme clausale), pouvoir d'expression limité
- logique du premier ordre :
  - possibilité d'utiliser d'exprimer des propriétés sur des domaines infinis
  - possibilités de formaliser une grande partie des mathématiques en logique du premier ordre (avec égalité) (structure algébrique : groupe, anneaux, corps, théorie des ensembles)
  - logique du premier ordre, logique universelle (point de passage obligé pour étudier d'autres logiques)
  - logique du premier ordre : notions générales (sémantique, systèmes formels, mise sous forme clausale,...)

- Langage du premier ordre
- Termes (principe d'induction)
- Substitution, filtrage, unification
- Atomes, formules (Principe d'induction pour les formules)
- Variables libres, variables liées, formules polies
- Application d'une substitution à une formule (pb de capture)
- Clôture universelle, clôture existentielle d'une formule
- Vers la sémantique et l'interprétation des formules

Definition (Langage du premier ordre)

Un langage $\mathcal{L}$ du premier ordre est la donnée des ensembles de symboles suivants :

- un ensemble X de symboles de variables $X = \{x,\ y,\ z,\ \ldots\}$
- une suite d'ensemble deux à deux disjoints de symboles de fonctions $F = (\mathbb{F}_n)_{n\in\mathbb{N}}$, chaque élément de $\mathbb{F}_n$ est un symbole de fonction d'arité n. Les éléments de F sont notés $f,\ g,\ \varphi, \ldots$. Les symboles de fonctions d'arité 0 sont appelés des symboles de constantes, on les note $\{a,\ b,\ c\ , \ldots\}$
- une suite d'ensemble deux à deux disjoints de symboles de relations $R = (\mathbb{R}_n)_{n\in\mathbb{N}}$, chaque élément de $\mathbb{R}_n$ est un symbole de relation d'arité n. Les éléments de R sont notés $p,\ q,\ r \ldots$
- le symbole d'égalité $=$ qui est un symbole de relation que l'on distingue des autres symboles de R
- l'ensemble des connecteurs logiques utilisés dans la logique des propositions : $\{\Rightarrow,\ \vee,\ \wedge,\ \neg,\ \mathcal{V},\ \mathcal{F},\ \Leftrightarrow, \ldots\}$
- deux connecteurs $\forall$ ("pour tout") et $\exists$ ("il existe")
- des symboles de ponctuation (et) et ,

Definition (Termes)

Soient X un ensemble de symboles de variables et F un ensemble de symboles de fonctions l'ensemble des termes construits sur X et F est défini inductivement de la façon suivante :

- les variables et les constantes sont des termes
- si $t_1, \ldots, t_n$ sont des termes alors $f(t_1, \ldots, t_n)$ est un terme.

On note $T(F, X)$ l'ensemble de termes construits sur F et X.
Les termes peuvent être représentés par des arbres étiquetés.

Remarques et notations

- les termes peuvent se construire par la grammaire algébrique suivante :
  $\mathcal{T} = X \;|S_c\; |f(\mathcal{T}, \ldots, \mathcal{T})$
- un terme est dit **clos** s'il est sans variable
- si t est un terme $V(t)$ est l'ensemble des variables ayant des occurrences dans t

Exemples

Soient $X = \{x,\ y,\ z, \ldots\}$ un ensemble de variables et
$F = \{f,\ g,\ s,\ a,\ b,\ c\ \}$ un ensemble de symboles fonctionnels,
tels que $arité(f) = 2$, $arité(g) = 2$, $arité(s) = 1$, $arité(a) = 0$, $arité(b) = 0$, $arité(c) = 0$

- a, x, c sont des termes, (a et c sont des termes clos)
- $s(y)$, $s(c)$, $f(a,\ x)$, $g(a,\ c)$, $f(y,\ y)$, $g(x,\ z)$, $f(a,\ a)$ sont des termes
- $s(f(a,\ x))$ est un terme
- $t = f(g(a,\ x),\ f(s(z),\ f(x,\ b)))$ est un terme, $V(t) = \{x,\ z\}$

Principe d'induction structurel sur les termes

Proposition

Soit une propriété $P(t)$ dépendant d'un terme t, pour montrer que $P(t)$ est vraie pour tout terme t, il suffit de montrer les deux assertions suivantes :

- cas de base
  - pour toute constante c, $P(c)$ est vrai
  - pour toute variable x, $P(x)$ est vrai
- cas général
  pour tout terme t_1,...,t_n et tout symbole f,
  $P(t_1)$ et ... et $P(t_n)$ implique $P(f(t_1, \dots, t_n))$

Definition (Taille d'un terme)

La taille d'un terme est le nombre d'occurrences de symboles apparaissant dans ce terme. Formellement :

$taille(x) = 1$ si x est une variable
$taille(c) = 1$ si c est une constante

$$taille(f(t_1, \ldots, t_n)) = 1 + \sum_{i=1}^{n} taille(t_i)$$

Exemple

$u = g(f(s(a),\ s(x)),\ f(s(z),\ f(x,\ b)))$ est un terme, $taille(u) = 12$

Soient un ensemble de variables X et un ensemble de symboles fonctionnels F

Definition (Substitution)

Une substitution est une application σ de X vers $T(F, X)$

$$\sigma : X \to T(F, X)$$

Le domaine de σ est $dom(\sigma) = \{x; x \in X \text{ et } \sigma(x) \neq x\}$.
Une substitution de domaine fini est notée $\{x_1 \mapsto t_1, \ \ldots, \ x_n \mapsto t_n\}$

Exemple

- $\sigma_1 = \{x \mapsto f(z, \ b), \ z \mapsto s(f(y, \ s(x)))\}$,
  on a donc $\sigma_1(x) = f(z, \ b)$ et $\sigma_1(z) = s(f(y, \ s(x)))$
  $dom(\sigma_1) = \{x, \ z\}$
  $\sigma_1(y) = y$ car $y \notin dom(\sigma_1)$
- $\sigma_2 = \emptyset$ est la substitution vide (c'est-à-dire la fonction identique telle que pour toute variable x de X, $\sigma_2(x) = x$)
- on considère en général des substitutions dont les domaines sont finis

Definition

On étend l'application des substitutions aux termes de la façon suivante, si σ est une substitution :

- $\sigma(x)$ est défini si x est une variable
- $\sigma(c) = c$ si c est une constante
- $\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$ (on dit que σ est un homomorphisme)

Exemple

Soient la substitution σ définie par $\sigma = \{x \mapsto g(y,\ s(a)),\ z \mapsto g(f(a,\ b), x)),\ v \mapsto s(z)\}$ et le terme $t = f(g(x,\ y),\ f(s(z),\ f(b,\ s(x))))$

$\sigma(t) =$
$\sigma(f(g(x,\ y),\ f(s(z),\ f(b,\ s(x))))) = f(\sigma(g(x,\ y)),\ \sigma(f(s(z),\ f(b,\ s(x))))) =$
$f(g(\sigma(x),\ \sigma(y)),\ f(\sigma(s(z)),\ \sigma(f(b,\ s(x))))) =$
$f(g(\sigma(x),\ \sigma(y)),\ f(s(\sigma(z)),\ f(\sigma(b),\ \sigma(s(x))))) =$
$f(g(\sigma(x),\ \sigma(y)),\ f(s(\sigma(z)),\ f(\sigma(b),\ s(\sigma(x))))) =$
$f(g(g(y,\ s(a)),\ y),\ f(s(g(f(a,\ b),\ x)),\ f(b,\ s(g(y,\ s(a))))))$

Definition (Filtrage)

Soient t et t' deux termes, on dit que t filtre t' si et seulement s'il existe une substitution σ telle que $\sigma(t) = t'$.

Exemples

- $t = b$, $t' = a$
  - il n'existe pas de filtre de t vers t',
  - il n'existe pas de filtre de t' vers t
- $t = x$, $t' = y$
  - $\{x \mapsto y\}$ est un filtre de t vers t'
  - $\{y \mapsto x\}$ est un filtre de t' vers t
- $t = x$, $t' = f(x,\ s(g(z,\ a)))$,
  - la substitution $\{x \mapsto f(x,\ s(g(z,\ a)))$ est une filtre de t vers t'
  - il n'existe pas de filtre de t' vers t
- $t = f(g(x,\ y),\ s(z))$, $t' = f(g(s(b),\ f(a,\ c)),\ s(s(b)))$,
  $\sigma = \{x \mapsto s(b),\ y \mapsto f(a,\ c),\ z \mapsto s(b)\}$ est un filtre de t vers t'

Definition (Unification)

Soient t et t' deux termes, t et t' s'unifient si et seulement s'il existe une substitution σ telle que $\sigma(t) = \sigma(t')$, une telle substitution s'appelle un unificateur de t et t'.

Exemples

- $t = x$, $t' = y$, $\{x \mapsto y\}$, $\{y \mapsto x\}$ et $\{x \mapsto z,\ y \mapsto z\}$ sont des unificateurs de t et t'
- a et b ne sont pas unifiables
- $f(x,\ y)$ et $g(a,\ x)$ ne sont pas unifiables
- $t = f(x,\ g(z,\ s(z)))$, $t' = f(s(b),\ g(a,\ y))$ $\{x \mapsto s(b),\ z \mapsto a,\ y \mapsto s(a)\}$ est un unificateur de t et t'

Remarques

- deux termes dont les symboles fonctionnels de tête sont différents ne sont pas unifiables (exemple : $f(x,\ s(a))$ et $g(s(b),\ y)$ ne sont pas unifiables)
- une variable et un terme contenant cette variable ne sont pas unifiables (exemple : x et $s(x)$ ne sont pas unifiables)
- il existe des algorithmes d'unification permettant de déterminer si deux termes sont unifiables et si c'est le cas déterminent un unificateur de ces deux termes

Definition

Soient X un ensemble de symboles de variables, F un ensemble de symbole de fonctions et R un ensemble de symbole de relations, un atome est de la forme $r(t_1, \ldots, t_n)$ où

- r est un symbole de relation d'arité n et
- $t_1, \ldots, t_n$ sont des termes de $T(F, X)$.

Remarques et exemples

- dans la définition d'un atome le symbole de relation n'apparaît qu'une fois (en tête de l'atome)
- les notions de substitution, filtrage et unification peuvent être aisément étendues aux atomes
- $X = \{x, y, z, u\}$, $F = \{f, g, s, a, b, c\}$, $R = \{p, q, r\}$ (on suppose que l'arité de p est 1 et celles de q et r sont 2)
  - $p(a)$ est un atome
  - $q(a, s(x))$ est un atome
  - $f(x, a)=s(y)$ est un atome (car $=$ est un symbole de relation, notation infixée)
  - $r(f(g(s(x), a), s(b)), g(s(x), c))$ est un atome
  - $r(p(a), x)$ n'est pas un atome car $p(a)$ n'est pas un terme

Definition (Formules)

Soit $\mathcal{L}$ un langage du premier ordre, l'ensemble des formules (de la logique du premier ordre) de $\mathcal{L}$ est défini par la grammaire suivante :

$$Y = \mathcal{A}tom \ |Y \vee Y \ |Y \wedge Y \ |Y \Rightarrow Y \ |\neg Y \ |\exists x Y \ |\forall x Y$$

où x parcourt l'ensemble des variables.

Remarques

- L'ensemble des formules est le plus petit ensemble contenant les atomes et tel que si f_1 et f_2 sont des formules alors $f_1 \vee f_2$, $f_1 \wedge f_2$, $f_1 \Rightarrow f_2$, $\neg f_1$, $\exists x \ f_1$ et $\forall x \ f_1$ sont des formules
- les formules peuvent être vues comme des arbres dont les nœuds internes sont étiquetés par les connecteurs ($\vee$, $\wedge$, $\Rightarrow$ et $\neg$) ou les quantificateurs ($\exists x$, $\forall x$) et les feuilles par des atomes
- pour démontrer une propriété P sur les formules il suffit de prouver P pour les atomes et de prouver $P(f_1 \vee f_2)$, $P(f_1 \wedge f_2)$, $P(f_1 \Rightarrow f_2)$, $P(\neg f_1)$, $P(\exists x \ f_1)$ et $P(\forall x \ f_1)$, en supposant que $P(f_1)$ et $P(f_2)$ sont vraies
- un littéral est soit un atome (i.e. de la forme $r(t_1, \ldots, t_n)$) soit la négation d'un atome (i.e. de la forme $\neg r(t_1, \ldots, t_n)$)

- les quantificateurs $\exists$ et $\forall$ sont prioritaires par rapport aux connecteurs logiques
- les connecteurs logiques sont par ordre de priorité décroissante : $\neg$, puis $\vee$ et $\wedge$ et enfin $\Rightarrow$ et $\Leftrightarrow$
- on fusionne les listes de quantificateurs identiques, $\exists x_1 \exists x_2 \forall x_3 \forall x_4 \forall x_5 f$ peut être abrégé en $\exists x_1 x_2 \forall x_3 x_4 x_5 f$
- $f_1 = \exists x\ p(x,\ y) \vee r(x)$ et $f_2 = \exists x(p(x,\ y) \vee r(x))$ sont deux formules différentes

Exemples

- $X = \{x,\ y,\ z\}$, $F = \emptyset$, $R = \{r\}$
  $\forall x\ r(x,\ x)$
  $\forall x\ \forall y\ r(x,\ y) \Rightarrow r(y,\ x)$
  $\forall x\ \forall y\ \forall z\ r(x,\ y) \wedge r(y,\ z) \Rightarrow r(x,\ z)$
- arithmétique de Peano : $X = \{x,\ y,\ z\}$, $F = \{0,\ s,\ +,\ \times\}$, arité$(0) = 0$, arité$(s) = 1$, arité$(+) = 2$, arité$(\times) = 2$
  (A_1) : $\forall x\ \neg(s(x) = 0)$
  (A_2) : $\forall x\ (x = 0 \vee \exists y\ x = s(y))$
  (A_3) : $\forall x\ \forall y\ s(x) = s(y) \Rightarrow x = y$
  (A_4) : $\forall x\ 0 + x = x$
  (A_5) : $\forall x\ \forall y\ s(x) + y = s(x + y)$
  (A_6) : $\forall x\ 0 \times x = 0$
  (A_7) : $\forall x\ \forall y\ s(x) \times y = x \times y + y$

Definition

Soit f une formule de la logique du premier ordre, l'ensemble des variables libres de f, noté $VL(f)$ est défini inductivement de la façon suivante selon la forme de la formule :

- $VL(r(t_1, \ldots, t_n)) = V(t_1) \cup \ldots V(t_n)$ si $r(t_1, \ldots, t_n)$ est un atome
- $VL(t_1 = t_2) = V(t_1) \cup V(t_2)$ si t_1 et t_2 sont des termes
- $VL(\neg f) = VL(f)$ si f est une formule
- $VL(f_1 \vee f_2) = VL(f_1) \cup VL(f_2)$ si f_1 et f_2 sont des formules (de même $VL(f_1 \wedge f_2) = VL(f_1 \Rightarrow f_2) = VL(f_1) \cup VL(f_2)$)
- $VL(\exists x\ f_1) = VL(f_1) \setminus \{x\}$ si x est une variable et f_1 est une formule
- $VL(\forall x\ f_1) = VL(f_1) \setminus \{x\}$ si x est une variable et f_1 est une formule

Remarques et exemples

- une variable est libre si elle possède une occurrence qui n'est pas sous l'influence d'un quantificateur
- une formule f est dite close si et seulement si $VL(f) = \emptyset$
- lorsqu'on applique une substitution à une formule, on substitue seulement les variables libres de cette formule
- $f = \forall x\ (x.y = y.x)$, $VL(f) = \{y\}$
- $g = (\forall x\ \exists y\ (x.z = z.y)) \wedge (x = z.z)$, $VL(g) = \{x, z\}$
- $h = \forall x\ (y = 0)$, $VL(h) = \{y\}$

Definition (Variable liée d'une formule)

L'ensemble des variables liées (ou muettes) $VM(f)$ d'une formule f du premier ordre est défini selon la forme de la formule de la façon suivante :

- $VM(r(t_1, \ldots, t_n)) = \emptyset$ si r est un symbole de relation et $t_1, \ldots, t_n$ sont des termes (i.e. $r(t_1, \ldots, t_n)$ est un atome), de même $VM(t_1 = t_2) = \emptyset$
- $VM(\neg f) = VM(f)$ si f est une formule
- $VM(f_1 \vee f_2) = VM(f_1) \cup VM(f_2)$ si f_1 et f_2 sont des formules, de même $VM(f_1 \wedge f_2) = VM(f_1 \Rightarrow f_2) = VM(f_1) \cup VM(f_2)$
- $VM(\forall x \ f) = VM(f) \cup \{x\}$ si x est une variable et f une formule
- $VM(\exists x \ f) = VM(f) \cup \{x\}$ si x est une variable et f une formule

Remarques et exemples

- les variables liées (ou muettes) sont celles qui sont sous l'influence d'un quantificateur,
- $f = \forall x \ (x.y = y.x)$, $VM(f) = \{x\}$ $g = (\forall x \ \exists y \ (x.z = z.y)) \wedge (x = z.z)$, $VM(g) = \{x, \ y\}$
- $h = \forall x \ (y = 0)$, $VM(h) = \{x\}$

Definition (Formule polie)

Une formule f est polie si et seulement si :

- $VL(f) \cap VM(f) = \emptyset$ (une variable n'est pas à la fois libre et liée)
- deux occurrences d'une même variable liée correspondent à la même occurence de quantificateur

Exemples et remarques

- $\forall x\ \exists y\ r(x,\ y) \vee r(x, z)$ n'est pas une formule polie car la variable x et à la fois *libre* et *liée*
- $(\forall x\ r(x,\ y)) \vee (\exists x\ p(x,\ z))$ n'est pas une formule polie, car il existe deux occurrences liées de la variable x qui ne correspondent pas à la même occurrence de quantificateur
- on peut rendre une formule polie en renommant systématiquement les variables liées par de nouvelles variables (i.e. des variables qui n'apparaissent nulle part ailleurs dans la spécification)
- il est important de mettre les formules sous forme polie, cela évite les problèmes de capture lors de l'application des substitutions
- en mathématiques on respecte en général la première condition de la définition, mais la deuxième est moins respectée

Soit la formule

$$f = \forall x \ p(x, \ y)$$

on a $VL(f) = \{y\}$ et $VM(f) = \{x\}$
Appliquer une substitution σ sur la formule f revient à appliquer σ aux variables libres de f. Par exemple :

- si $\sigma = \{y \mapsto a\}$ on obtient $\sigma(f) = \forall x \ p(x, \ a)$
- si $\sigma = \{y \mapsto s(z)\}$ on obtient $\sigma(f) = \forall x \ p(x, \ s(z))$
- si $\sigma = \{y \mapsto s(x)\}$ on obtient $\sigma(f) = \forall x \ p(x, \ s(x))$ cette application de substitution est fausse car on a phénomène de capture, ici l'occurence x de la variable x a été capturée par le quantificateur $\forall x$, elle est ainsi devenue liée alors qu'elle aurait du rester libre.
  Pour pallier ce phénomène de capture, il est nécessaire de renommer auparavant la variable liée x, la formule f devient donc $\forall x_1 \ p(x_1, \ y)$ où x_1 est une nouvelle variable, n'apparaissant pas ailleurs dans les spécifications.
  On peut alors appliquer la substitution $\sigma = \{y \mapsto s(x)\}$ et l'on obtient $\sigma(f) = \forall x_1 \ p(x_1, \ s(x))$

Definition

- Une formule **close** est une formule sans variables libres.
- Soit f une formule dont les variables libres sont $x_1, \ldots, x_n$. La clôture universelle de f est la formule $\forall x_1 \ldots \forall x_n\ f$
- Soit f une formule dont les variables libres sont $x_1, \ldots, x_n$. La clôture existentielle de f est la formule $\exists x_1 \ldots \exists x_n\ f$

Soit la formule

$$\varphi = p(x,\ a) \wedge \exists y \exists z\ p(y,\ z)$$

pour donner une sémantique à φ il est nécessaire de

- dire dans quel ensemble varient les variables de φ
- donner une représentation des symboles de la formule : **p**, **a**

On doit donc définir la notion d'**interprétation** (Voir prochain cours)