Modélisation - Résumé

October 20, 2023

THEVENET Louis

Table des matières

1. Logique propositionnelle

Définition 1.1:

La logique propositionnelle ne parle que de vérité :
- elle ne permet pas de faire référence à des objets, ou à des notions,
- elle ne permet pas de mettre objets ou notions en rapport.

2. Logique des prédicats

Définition 2.1:

C'est l'ajout des quantificateurs, des relations et des structures à la logique propositionnelle.

Extension de la logique des propositions :
- Univers $\mathcal{U}$ (objets mathématiques ou informatiques)
- Algèbre de termes (représentation des objets) : constantes et opérateurs sur $\mathcal{U}$
- Quantificateurs pour variables dans $\mathcal{U}$: $\forall, \exists$
- Relations sur $\mathcal{U}$ (permet aussi de représenter les termes)

Mais aussi :
- $\bot$ $\top \neg \land \lor \to \leftrightarrow \mathcal{P}()$
- Ensembles dénombrables de symboles :
  - Variables $\mathcal{V}$
  - Relations (prédicats) $\mathcal{R}$ munie d'une arité $\in \mathbb{N}^*$
  - Propositions $\mathcal{P}$ (relations d'arité 0)

    - Fonctions $\mathcal{F}$ munie d'une arité $\in \mathbb{N}^*$
    - Constantes $\mathcal{C}$ (fonctions d'arité 0)
- Lieurs : $\forall, \exists$
- Paramètres des relations et fonctions : $(,)$

2.1. Ordres de la logique des prédicats

Définition 2.1.1 :

- Ordre supérieur : les lieurs peuvent quantifier des termes, des relations, des propositions, des fonctions, des constantes
- Premier ordre (First Order Logic, FOL) : Les lieurs ne peuvent quantifier que des termes
- Second ordre (SOL) : on peut quantifier sur des ensembles de termes

Exemple : du premier ordre avec $\mathcal{V} = \{m, n\}, \mathcal{R}_1 = \{\text{entier}\}, \mathcal{R}_2 = \{\text{egal}\}$

$$\forall m.(\text{entier}(m) \to (\text{impair}(m) \leftrightarrow (\exists n.(\text{entier}(n) \land \text{egal}(m, \text{somme}(\text{produit}(\text{deux}, n), \text{un}))))))$$

Exemple : du second ordre avec (g, o) est un groupe

$$\forall g.\forall o.\ \text{groupe}(g, o) \leftrightarrow \begin{cases} \forall x_1.\forall x_2.g(x_1) \land g(x_2) \to g(o(x_1, x_2)) \\ \land \exists e.g(e) \land \begin{cases} \forall x.g(x) \to \text{egal}(o(x, e), x) \land \text{egal}(o(e, x), x) \\ \land \forall x_1.\forall x_2.\forall x_3.g(x_1) \land g(x_2) \land g(x_3) \to \text{egal}(o(o(x_1, x_2), x_3), o(x_1, o(x_2, x_3))) \\ \land \forall x_1.g(x_1) \to \exists x_2.g(x_2) \land \text{egal}(o(x_1, x_2), e) \land \text{egal}(o(x_2, x_1), e) \end{cases} \end{cases}$$

3. Spécification algébrique

Définition 3.1: Typage des constantes et opérateurs

Soit $\mathcal{S}$ un ensemble dénombrable de symboles, ce sont les **sortes** utilisées pour distinguer les termes possédant les mêmes caractéristiques, ainsi on classe

- les termes : $\mathcal{T} = \bigcup_{s\in\mathcal{S}} \mathcal{T}_s$
- les constantes : $\mathcal{C} = \bigcup_{s\in\mathcal{S}} \mathcal{C}_s$
- les variables : $\mathcal{V} = \bigcup_{s\in\mathcal{S}} \mathcal{V}_s$
- L'arité des fonctions prend en compte la sorte des paramètres et du résultat :

$$\forall n \in \mathbb{N}.\mathcal{F}_n = \bigcup_{s\in\mathcal{S},\forall i\in[1,...,n].s_i\in\mathcal{S}} \mathcal{F}_{(s_1\times...\times s_n)\mapsto s}$$

Ainsi l'arité prend en compte les **sortes**

Exemple : Vision ensembliste

Soit $\mathcal{S}$ un ensemble dénombrable de sortes, $\mathcal{T}$ est le plus petit ensemble tel que :
- $\forall s \in \mathcal{S}.\forall c \in \mathcal{C}_s.c \in \mathcal{T}_s$
- $\forall s_1, ..., s_n, s \in \mathcal{S}.\forall f \in \mathcal{F}_{s_1\times...\times s_n\mapsto s}.\forall t_1, ..., t_n \in \mathcal{T}_{s_1} \times ... \times \mathcal{T}_{s_n}.f(t_1, ..., t_n) \in \mathcal{T}_s$

Exemple : Entiers naturels de Peano

$$\begin{gathered}\text{nat} \in \mathcal{S}\\ \text{zero} \in \mathcal{C}_{\text{nat}}\\ \text{successeur} \in \mathcal{F}_{\text{nat}\,\mapsto\,\text{nat}}\end{gathered}$$

L'ensemble des termes est la plus petite solution de l'équation :

$$\mathcal{T}_{\text{nat}} = \{\text{zero}\} \cup \{\text{successeur}(n) \mid n \in \mathcal{T}_{\text{nat}}\}$$

Définition 3.2: Termes avec variables

On note $\mathcal{T}[\mathcal{V}]$ l'ensemble des termes avec variables **partitionné selon les sortes**, il est le plus petit ensemble tel que :
- $\forall s \in \mathcal{S}.\forall c \in \mathcal{C}_s.c \in \mathcal{T}[\mathcal{V}]_s$
- $\forall s \in \mathcal{S}.\forall x \in \mathcal{V}_s.x \in \mathcal{T}[\mathcal{V}]_s$
- $\forall s_1, ..., s_n, s \in \mathcal{S}.\forall f \in \mathcal{F}_{s_1\times...\times s_n\mapsto s}.\forall t_1, ..., t_n \in \mathcal{T}[\mathcal{V}]_{s_1} \times ... \times \mathcal{T}[\mathcal{V}]_{s_n}.f(t_1, ..., t_n) \in \mathcal{T}[\mathcal{V}]_s$

Exemple : Arithmétique de Peano

- On modélise $\mathbb{N}$ par :
  - zero $\in \mathcal{C}_0 (\overline{0} = \emptyset)$
  - successeur $\in \mathcal{F}_1 (\overline{n+1} = \{\overline{n} \cup \overline{n}\})$
- Puis $\mathbb{Z}$ par $\mathbb{N}^2$ avec :
  - $(n, 0)$ modélise $\mathbb{Z}^+$
  - $(0, n)$ modélise $\mathbb{Z}^-$

4. Variables libres

Exemple :

$$\begin{aligned}
&\mathrm{VL}(\forall x.(\varphi \leftrightarrow \exists y.\psi)) \\
&= \mathrm{VL}(\varphi \leftrightarrow \exists y.\psi) \setminus \{x\} \\
&= (\mathrm{VL}(\varphi) \cup \mathrm{VL}(\exists y.\psi)) \setminus \{x\} \\
&= (\mathrm{VL}(\varphi) \cup (\mathrm{VL}(\psi) \setminus \{y\})) \setminus \{x\}
\end{aligned}$$

5. Substitution

Exemple :

$[f(x,y)/x]$: *$f(x,y)$ remplace x*

$$\begin{aligned}
&[f(x,y)/x]((x \to y) \land \exists y.(x \lor ((\forall x.\varphi) \to y))) \\
&= ([f(x,y)/x](x \to y)) \land [f(x,y)/x](\exists y.(x \lor ((\forall x.\varphi) \to y))) \\
&= ([f(x,y)/x](x) \to [f(x,y)/x](y)) \land [f(x,y)/x](\exists y.(x \lor ((\forall x.\varphi) \to y))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x][z,y](x \lor (((\forall x.\varphi) \to y)))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x]([z,y](x) \lor [z,y](((\forall x.\varphi) \to y)))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x](x \lor ((\forall x.[z/y](\varphi)) \to [z,y](y)))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x](x \lor ((\forall x.[z/y](\varphi)) \to [z,y](y)))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x](x \lor ((\forall x.[z/y](\varphi)) \to z))) \\
&= (f(x,y) \to y) \land (\exists z.[f(x,y)/x](x \lor ((\forall x.[z/y](\varphi)) \to z))) \\
&= (f(x,y) \to y) \land (\exists z.([f(x,y)/x](x) \lor [f(x,y)/x]((\forall x.[z/y](\varphi)) \to z))) \\
&= (f(x,y) \to y) \land (\exists z.(f(x/y) \lor [f(x,y)/x]((\forall x.[z/y](\varphi)) \to z))) \\
&= (f(x,y) \to y) \land (\exists z.(f(x/y) \lor ((\forall x.[z/y]\varphi) \to z)))
\end{aligned}$$

6. Preuves de programme

Exemple : spécification formelle (pré-condition, post-condition)

```
{0 ≤ N}
x := 0;
y := 0;
while x != N do
    y := y + 2 * x + 1;
    x := x + 1
od
{y = N²}
```

6.1. Preuve de correction

Théorème 6.1.1:

- Chaque étape intermédiaire est annotée par une propriété de l'état de la mémoire
- Chaque instruction I est
  - précédée d'une pré-condition φ
  - suivie d'une post-condition ψ
- Chaque instruction annotée doit satisfaire les règles de la logique de Hoare : $\{\varphi\}I\{\psi\}$
  - Correction partielle : φ est satisfaite et l'exécution termine alors ψ est satisfaite après exécution
  - Correction totale : φ est satisfaite alors l'exécution termine et ψ est satisfaite après exécution
- On représente les propriété sur l'état de la mémoire avec la **logique équationnelle** (i.e. premier ordre + spécifications algébriques)

6.1.1. Preuve de correction partielle

Exemple : Preuve de correction **partielle** de l'élevation au carré (invariant : $y = x^2$)

Si on veut que ψ_x soit vraie après avoir fait $(x \leftarrow E)$, il faut que qu'elle soit vraie pour E, i.e., on fait appraître E dans φ (*)

$$
\begin{array}{l}
\{0 \le N\} \\
\{0 = 0^2\} \\
x := 0; \\
\{0 = x^2\} \\
y := 0; \\
\{y = x^2\} \\
\text{while } x \neq N \text{ invariant } y = x^2 \text{ do} \\
\quad \{y = x^2 \land x \neq N\} \\
\quad (*)\{y + 2 \times x + 1 = (x+1)^2\} \\
\quad y := y + 2 \times x + 1; \\
\quad \left\{y = (x+1)^2\right\} \\
\quad x := x + 1; \\
\quad \{y = x^2\} \\
\text{od} \\
\{y = x^2 \land \neg(x \neq N)\} \\
\{y = N^2\}
\end{array}
$$

6.1.2. Preuve de terminaison

Exemple : Preuve de correction **totale** de l'élévation au carré (invariant : $y = x^2$)

Elle sera totale car on a déjà prouvé la correction partielle. On pourrait combiner les preuves en remplaçant les ... par la preuve par invariant précédente.

$\{0 \leq N\}$

$\{... \land (N - 0) \in \mathbb{N}\}$

$x := 0;$

$\{... \land N - x \in \mathbb{N}\}$

$y := 0;$

$\{N - x \in \mathbb{N}\}$

while $x \neq N$ invariant $y = x^2$ variant $N - x$ do

  $\{... \land x \neq N \land (N - x) \in \mathbb{N} \land V = N - x\}$

  $y := y + 2 \times x + 1;$

  $\{... \land (N - (x + 1)) \in \mathbb{N} \land N - (x + 1) < V\}$

  $x := x + 1;$

  $\{... \land (N - x) \in \mathbb{N} \land N - x < V\}$

od

$\{...\}$

$\{y = N^2\}$

Puis

$$0 \leq N \to 0 = 0^2 \land (N - 0) \in \mathbb{N}$$

$$\begin{cases} y = x^2 \\ \land\, x \neq N \\ (N - x) \in \mathbb{N} \\ (N - x) = V \end{cases} \to \begin{cases} y + 2 \times x + 1 = (x + 1)^2 \\ (N - (x + 1)) \in \mathbb{N} \\ (N - (x + 1)) < V \end{cases}$$

$$y = x^2 \land \neg(x \neq N)) \to y = N^2$$