

Modélisation - Résumé

October 20, 2023

THEVENET Louis

Table des matières

1. Logique propositionnelle	1
2. Logique des prédicats	1
2.1. Ordres de la logique des prédicats	2
3. Spécification algébrique	2
4. Variables libres	3
5. Substitution	3
6. Preuves de programme	4
6.1. Preuve de correction	4
6.1.1. Preuve de correction partielle	5
6.1.2. Preuve de terminaison	5

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 :

- $\perp \top \neg \wedge \vee \rightarrow \leftrightarrow \mathcal{P}()$
- 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 $\mathcal{V} = \{m, n\}, \mathcal{R}_1 = \{\text{entier}\}, \mathcal{R}_2 = \{\text{egal}\}$

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

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

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

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 :

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

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 :

- $\forall s \in \mathcal{S}. \forall c \in \mathcal{C}_s. c \in \mathcal{T}_s$
- $\forall s_1, \dots, s_n, s \in \mathcal{S}. \forall f \in \mathcal{F}_{s_1 \times \dots \times s_n \mapsto s}. \forall t_1, \dots, t_n \in \mathcal{T}_{s_1} \times \dots \times \mathcal{T}_{s_n}. f(t_1, \dots, t_n) \in \mathcal{T}_s$

Exemple : Entiers naturels de Peano

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

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 :

- $\forall s \in \mathcal{S}. \forall c \in \mathcal{C}_s. c \in \mathcal{T}[\mathcal{V}]_s$
- $\forall s \in \mathcal{S}. \forall x \in \mathcal{V}_s. x \in \mathcal{T}[\mathcal{V}]_s$
- $\forall s_1, \dots, s_n, s \in \mathcal{S}. \forall f \in \mathcal{F}_{s_1 \times \dots \times s_n \mapsto s}. \forall t_1, \dots, t_n \in \mathcal{T}[\mathcal{V}]_{s_1} \times \dots \times \mathcal{T}[\mathcal{V}]_{s_n}. f(t_1, \dots, t_n) \in \mathcal{T}[\mathcal{V}]_s$

Exemple : Arithmétique de Peano

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

4. Variables libres

Exemple :

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

5. Substitution

Exemple :

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

$$\begin{aligned}
& [f(x, y)/x]((x \rightarrow y) \wedge \exists y.(x \vee ((\forall x.\varphi) \rightarrow y))) \\
&= ([f(x, y)/x](x \rightarrow y)) \wedge [f(x, y)/x](\exists y.(x \vee ((\forall x.\varphi) \rightarrow y))) \\
&= ([f(x, y)/x](x) \rightarrow [f(x, y)/x](y)) \wedge [f(x, y)/x](\exists y.(x \vee ((\forall x.\varphi) \rightarrow y))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x][z, y](x \vee ((\forall x.\varphi) \rightarrow y))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x]([z, y](x) \vee [z, y]((\forall x.\varphi) \rightarrow y))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x](x \vee ((\forall x.[z/y](\varphi)) \rightarrow [z, y](y)))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x](x \vee ((\forall x.[z/y](\varphi)) \rightarrow [z, y](y)))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x](x \vee ((\forall x.[z/y](\varphi)) \rightarrow z))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.[f(x, y)/x](x \vee ((\forall x.[z/y](\varphi)) \rightarrow z))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.([f(x, y)/x](x) \vee [f(x, y)/x]((\forall x.[z/y](\varphi)) \rightarrow z))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.(f(x/y) \vee [f(x, y)/x]((\forall x.[z/y](\varphi)) \rightarrow z))) \\
&= (f(x, y) \rightarrow y) \wedge (\exists z.(f(x/y) \vee ((\forall x.[z/y](\varphi)) \rightarrow z)))
\end{aligned}$$

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 propriétés 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 apparaître E dans φ (*)

```
{0 ≤ N}
{0 = 0²}
x := 0;
{0 = x²}
y := 0;
{y = x²}
while x ≠ N invariant y = x² do
    {y = x² ∧ x ≠ N}
    (*){y + 2 × x + 1 = (x + 1)²}
    y := y + 2 × x + 1;
    {y = (x + 1)²}
    x := x + 1;
    {y = x²}
od
{y = x² ∧ ¬(x ≠ N)}
{y = N²}
```

6.1.2. Preuve de terminaison

Exemple : Preuve de correction **totale** de l'élevation 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 ≤ N}
{... ∧ (N - 0) ∈ ℕ}
x := 0;
{... ∧ N - x ∈ ℕ}
y := 0;
{N - x ∈ ℕ}
while x ≠ N invariant y = x2 variant N - x do
    {... ∧ x ≠ N ∧ (N - x) ∈ ℕ ∧ V = N - x}
    y := y + 2 × x + 1;
    {... ∧ (N - (x + 1)) ∈ ℕ ∧ N - (x + 1) < V}
    x := x + 1;
    {... ∧ (N - x) ∈ ℕ ∧ N - x < V}
od
{...}
{y = N2}

```

Puis

$$0 \leq N \rightarrow 0 = 0^2 \wedge (N - 0) \in \mathbb{N}$$

$$\left\{ \begin{array}{l} y = x^2 \\ \wedge x \neq N \\ (N - x) \in \mathbb{N} \\ (N - x) = V \end{array} \right\} \rightarrow \left\{ \begin{array}{l} y + 2 \times x + 1 = (x + 1)^2 \\ (N - (x + 1)) \in \mathbb{N} \\ (N - (x + 1)) < V \end{array} \right.$$

$$y = x^2 \wedge \neg(x \neq N) \rightarrow y = N^2$$