Logique

NFP108/SEMS
2017-2018

Systèmes formels

O. Pons

8 octobre 2018

Logique formelle

- Logique formelle comprend
  - **langage** pour écrire des **formules**,
  - **interprétation** (sémantique) : donner du sens aux formules (définir les formules valides),
  - **système de déduction** : construire des preuves de formules (définir les formules prouvables).
- Méta-théorie
  - **contradictoire** : toute formule est-elle valide/prouvable ? (sinon la logique est dite cohérente)
  - **correction** (soundness) : toute formule prouvable est elle valide ?
  - **complétude** : toute formule valide est elle prouvable ?
  - **décidabilité** : un algorithme pour décider si une formule est valide ou non ?

Langages formels

- Exprimé avec des **Symboles** ou (caractères).
- Ensemble de **Mots** (ou chaîne de caractère).

- **Alphabet A** : ensemble de symboles
- **Mot sur A** : suite finie $(s_1, s_2, \ldots, s_n)$ de symboles de A
  élément A^n : mot de longueur n
- **Ensemble de tous les mots sur A** : $A^* = \underset{n \geq 0}{U} A^n$

Langages formels

- Opérations binaires sur A^* : concaténation

$$\begin{array}{lcl} A^p \times A^q & \rightarrow & A^{p+q} \\ u, v & \rightarrow & u.v \text{ (souvent } uv) \\ (u_1, ..., u_p), (v_1, ..., v_q) & \rightarrow & (u_1, ..., u_p, v_1, ..., v_q) \end{array}$$

  - associative,
  - élément neutre noté ϵ
  - stable
- $(A^*, ., \epsilon)$: **un monoïde**.

  on note a pour le mot (a), et $a_1 a_2 \ldots a_n$ pour $(a_1, a_2 \ldots, a_n)$
- **Langage** : partie de A^*.

Langages formels

- Exemple
  - Si $A = \emptyset, A^* = \{\epsilon\}$
  - Si $A = \{1\}$, A^* est isomorphe a $(\mathbb{N}, +, 0)$ Autant de langage d'alphabet $\{1\}$ que de partie de $\mathbb{N}$.
- Monoîde : manque de structuration !

$\Longrightarrow$ "Ajouter de la structure"

Récurrence / Induction

- Idéé : Restreindre les parties à considérer grace à des « règles de construction » (Constructeurs/Prédicats)
  Plus petit sous ensemble de A^* clos par application des constructeurs

Récurrence / Induction : exemple

- Les Entiers : *Nat*
  - Alphabet : $A = \{O, S, (,)\}$
  - Règles :
    1. $O \in Nat$
    2. Pour tout x si $x \in Nat$ alors $S(x) \in Nat$
  - Principe de récurrence
  - Preuve par récurrence
  - Fonctions définies par récurrence

Récurrence / Induction : exemple

- Les Expressions Arithmétiques : $ExpA$
  - Alphabet : $V = \{x, y, z \dots\}$ ensemble (dénombrable) de variables, $A = V \cup \{*, +, /, (,)\}$
    1. Tout élement de V est dans $ExpA$
    2. Pour tout x et y si $x \in ExpA$ et $y \in ExpA$ alors $(x * y) \in ExpA$
    3. Pour tout x et y si $x \in ExpA$ et $y \in ExpA$ alors $(x + y) \in ExpA$
    4. Pour tout x et y si $x \in ExpA$ et $y \in ExpA$ alors $(x / y) \in ExpA$
  - Principe d'Induction
  - Preuve par Induction
  - Fonctions définies par Induction