Logique

Xavier Urbain

2023-2024

Il y aura une page web...

Surveillez vos emails (rappels, informations. . .)

Vous m'écrivez ? ⇝ "LIFLC" + groupe dans le sujet politesse de base

depuis `@etu.univ-lyon1.fr`

Modalités de contrôle des connaissances : essentiellement CC

- Plus ou moins une interro "surprise" par TD → absence 0
- TP noté
- ECA

SILENCE DANS L'AMPHI

Sources et références :

- David/Nour/Raffalli : *Introduction à la logique*
- . . .
- Cours : Mijoule, Forest, Coquery, Goubault-Larrecq

On verra. . .

Pourquoi

Définition du raisonnement formel et abstrait

⇝ exprimer propriétés et démonstrations

Parce que

⇝ https://betterembsw.blogspot.com/p/potentially-deadly-automotive-software.html

⇝ https://raygun.com/blog/costly-software-errors-history/ (coût 2×10^{12} USD par an)

⇝ https://www.cs.tau.ac.il/~nachumd/horror.html. . .

Rapide. . . ok mais Correct ?

Spécification et vérification du comportement des prog.

- Univoque
- Claire

⇝ aller vers du code correct

Pourquoi

Succès UE

Validation si :

- Règles de raisonnement
- Induction
- Spécification au premier ordre

Échec si :

- Vous ne savez pas lire
- Vous ne connaissez pas votre cours

Comment

Déjà : ne pas confondre chiffres et nombres. . .

Bases de la prog. $\leftrightsquigarrow$ Bases du raisonnement

En particulier avec systèmes de types (à la ML)

Discipline de types	$\leftrightsquigarrow$	Logique
Type	$\leftrightsquigarrow$	Formule
Programme	$\leftrightsquigarrow$	Preuve
Calcul	$\leftrightsquigarrow$	Cut elim.

Cadre fonctionnel typé

Comment

Bases de la prog. $\leftrightsquigarrow$ Bases du raisonnement

Définition d'objets et d'ensembles

- Par décision $\rightsquigarrow$ LIFLF
- Par construction $\rightsquigarrow$ LIFLC

En LF : fonctions de reconnaissance. . . En LC : preuves de correction

Comment

Bases de la prog. $\leftrightsquigarrow$ Bases du raisonnement

TP : manipulation des deux $\rightsquigarrow$ (Gallina) Coq

`https://coq.inria.fr/` au moins 8.13

N'allez pas lire la doc !

Installation via opam

Fonctionnel, fortement typé, avec filtrage. . .

En moyenne plus facile que LIFAP2 : pas d'excuse

Rappels/notations

Notation : $E = \{x, y, z \dots\}$ $E = \{x \mid \dots\}$

Appartenance (à E) : $x \in E$ — x est dans E

Inclusion : $E \subseteq F$ — tout élément dans E est dans F

Égalité : double inclusion

Union : $G = E \cup F$ — $x \in G$ ssi ($x \in E$ OU BIEN $x \in F$)

Intersection : $G = E \cap F$ — $x \in G$ ssi ($x \in E$ ET AUSSI $x \in F$)

Différence : $G = E \setminus F$ — $x \in G$ ssi ($x \in E$ ET AUSSI $x \notin F$)

$I \cup (J \cap K) = (I \cup J) \cap K$? — $(I \cap J) \cap K = I \cap (K \cap J)$?

$I \subseteq (J \cap I) \cup K$? — $(I \cap J) \cup (J \cap K) \subseteq J$?

Rappels/notations

Notation : $E = (x, y, z \dots)$ — positions distinguées $\rightsquigarrow$ projections

Prod. cartésien : $E_1 \times \dots \times E_n = \{(x_1, \dots, x_n) \mid x_i \in E_i$ pour $i \in \{1, \dots, n\}\}$

positions...

Relation : $R \subseteq E_1 \times \dots \times E_n$ (relation n-aire)

$(e_1, \dots, e_n) \in R$ noté $R(e_1, \dots, e_n)$ — préfixe/infixe

Soit $R \subseteq E \times E$:

- Si pour chaque $R(x, y)$, alors $R(y, x)$: symétrique
- Si pour chaque $x \in E$, alors $R(x, x)$: réflexive
- Si pour chaque $R(x, y)$ et $R(y, z)$, alors $R(x, z)$: transitive

Antisymétrique si jamais de symétrie sauf égalité

Antiréflexive si jamais de réflexion

Rappels/notations

Notation : $E = (x, y, z \dots)$ — positions distinguées $\rightsquigarrow$ projections

Prod. cartésien : $E_1 \times \dots \times E_n = \{(x_1, \dots, x_n) \mid x_i \in E_i$ pour $i \in \{1, \dots, n\}\}$

positions...

Relation : $R \subseteq E_1 \times \dots \times E_n$ (relation n-aire)

$(e_1, \dots, e_n) \in R$ noté $R(e_1, \dots, e_n)$ — préfixe/infixe

---

Fonction : f relation sur $E_1 \times E_2$ — dans ce cas $f : E_1 \to E_2$

telle que pour chaque $x \in E_1$, au plus $y \in E_2$ tel que $(x, y) \in f$ — $y = f(x)$

- Au moins y : totale et — E_1 domaine
- Pour chaque $y \in E_2$,
  - au plus $x \in E_1, f(x) = y$: injective
  - au moins $x \in E_1, f(x) = y$: surjective

  } bijective

Rappels/notations

Fonctions / relations : rôles différents

Fonction « à plusieurs arguments » :

$f : E \to F$ avec $E = \{E_0 \times E_1 \times \dots \times E_{n-1} \times E_n\}$ — $f((x_0, \dots, x_n))$

Sous forme curryfiée :

$f : E_0 \to (E_1 \to \dots (E_{n-1} \to (E_n \to F) \dots)$ — $(\dots (f(x_0))(x_1) \dots)(x_n)$

Ensembles inductifs

Fondations math… prog… $\leadsto$ objets, ensembles

Construction ? règles… $(u_1, \dots, u_n) \in E \times \cdots \times E \to v \in E$

Schéma d'induction : (E, R) où R règles sur E $B = \{v \in E \mid \to v \in R\}$

Partie close pour (E, R) :

$F \subseteq E$ telle que pour toute $(u_1, \dots, u_n) \to v \in R,$ si tous $u_i \in F$ alors $v \in F$

Ensemble inductif de (E, R) :

la plus petite partie close pour (E, R)

$\mathbb{N}$ close de $(\mathbb{R}, \{\to 0;\ n \to n+1\})$, $\mathbb{R}$ aussi mais $\mathbb{N}$ la plus petite

Exemple listes finies et infinies…

Pas unicité du schéma : $\mathbb{N}$ induit par$(\mathbb{R}, \{\to 0;\ n \to 2n;\ n \to 2n+1\})$

Ensembles inductifs — Caractérisation 1

Proposition.

$X \subseteq E$ induit de (E, R) :

$$X = \bigcap_{Y \in \mathcal{F}} Y \qquad \text{avec } \mathcal{F} = \{Y \subseteq E \text{ close pour } (E, R)\}$$

$X \supseteq \bigcap_{Y \in \mathcal{F}} Y$ car $X \in \mathcal{F}$

$X \subseteq \bigcap_{Y \in \mathcal{F}} Y$ car X plus petite close (tout $Y \supseteq X$)

Preuve par induction

Idée : objets vérifiant la propriété = partie close (donc $\supseteq$ la plus petite)

Théorème.

X induit par (E, R), P propriété sur X telle que :

- $P(v)$ pour tout $v \in B$ $(\to v \in R)$
- $P(v)$ pour tous $u_1, \dots, u_n$ t. q. $(u_1, \dots, u_n) \to v \in R$ et $P(u_1) \dots P(u_n)$

alors $P(x)$ pour tout $x \in X$

Soit $U = \{x \in X \mid P(x)\}$ (on montre en fait $U = X$)

$U \subseteq X$ par déf.

$U \supseteq X$ car U close et $X = \bigcap_{Y \in \mathcal{F}} Y$ (pour $\mathcal{F}$: parties closes)

Ensembles inductifs — Fonctions

Idée : Y sur lequel on sait calculer la fonction est clos

Fonction $f : X \to F$ pour X induit par (E, R) :

- $f(v) \in F$ pour tout $v \in B$ $(\to v \in R)$
- $f(v)$ avec $f(u_1) \dots f(u_n)$ pour tous $u_1 \dots u_n$ et $(u_1, \dots, u_n) \to v \in R$

alors on a défini $f(x)$ pour tout $x \in X$

Rq. — Éventuellement cas supplémentaires… pas forcément struct.

Ex. double

Ensembles inductifs — Fonctions, exemple

$\mathbb{N}$ isomorphe à induit de $(\{S,O\}^\star, \{\ \to O;\ x \to S(x)\})$

(penser O vaut 0 et S vaut opération « + 1 »)

Donner les valeurs par cas de construction :

double(n) :

- Si n est O alors O (n vient de première règle)
- Si n est $S(n')$ alors $S(S($double$(n')))$ (n vient de deuxième règle)

(et on sait faire double(n'))

(car on montre domaine = partie close)

(donc forcément $n' \in$ domaine)

Bilan cours 1

Ensemble inductif : plus petite partie close par règles

Preuve par induction : établir comme close la partie pertinente

Fonctions f sur E inductif : établir comme clos le domaine où f correcte