Automates et Calculabilité

– Preuves . . . –

Session 2: 27 oct 2003

Uwe Nestmann

EPFL-I&C

Changements pour les exercices ?

Fortement demandé par les étudiants . . .

- Accès au corrigé, au moins dans la semaine suivante.

Alors:

- Assistance pendant l'heure d'exercices.
- Pas de rendus, pas de corrections individuelles.

Nous avons déjà réagi. Nous vous fournirons un corrigé de deux des trois exercices de la première session.
Le troisième sera présenté au cours.

Qu'est-ce que c'est, une preuve?

A proof is a *form of communication*.
When you write a proof
you are trying to convince someone else
(or possibly yourself)
that one statement follows from some others.

Epstein, Carnielli: “Computability” (2000)

- l'approche informelle
- l'approche formelle
- l'approche “mathématique”

L'approche informelle

In the USA of the 1990's it became popular to teach proof as a matter of personal feelings about the statement.
While it is good to feel the truth of a statement you need to use, important techniques of proof are no longer mastered . . .

Hopcroft, Motwani, Ullman: “Introduction to Automata Theory, Languages and Computation” (2001)

L'approche formelle

A *formal proof*
of a *proposition*
is a chain of *logical deductions*
leading to the propositions from a *base set of axioms*.

Meyer: “Mathematics for Computer Science”
(MIT course note, 2002)

L'approche formelle

Le premier but de la logique
est de caractériser les raisonnement corrects,
du point de vue de leur *forme* seulement.

Comme toute espèce de discours,
un raisonnement présente deux aspects:

- celui de sa forme (ses propriétés syntaxiques)
- et [celui] de sa signification (propriétés sémantiques).

La logique — tout au moins la logique élémentaire —
ne s'intéresse qu'à la *forme*,
et fait abstraction de toute signification.

Zahnd: “Logique élémentaire” (1998)

Avantages de l'approche formelle

- définition claire et simple de la notion de preuve . . .
- absence d'ambiguités
- vérifiable
- mécanisable (si la logique est décidable)

Preuves

axiomes “informels”	axiomes logiques
↓ preuve “mathématique”	⇓ preuve “formelle”
proposition “informelle”	proposition logique

Preuves

In real life, we are *not told to prove a theorem*.
Rather, we are faced with something that seems true
— a strategy for implementing a program for example —
and we need to *decide whether or not the "theorem" is true*.

Hopcroft, Motwani, Ullman: "Introduction to Automata Theory, Languages and Computation" (2001)

Où commencer?

- ☐ comprendre la proposition "informelle"
- ☐ **appliquer les définitions** (!)
  pour expliquer les entités de la proposition
- ☐ essayer d'extraire sa structure logique puis de la formaliser

($\Rightarrow$ se souvenir des exercices de Session 1 . . .)

Comment procéder?

Trouver une bonne stratégie pour conduire la preuve !

- ☐ Par déduction ?
  Exemples: "si A et $A \Rightarrow B$, alors B", "si A, alors $A \vee B$"
- ☐ Par construction ? ($\exists y : P(y)$)
- ☐ Par un contre-exemple ?
  À la place de $\neg(\forall x) : P(x)$ on prouve $(\exists x) : (\neg P(x))$.
- ☐ Par la contraposée (par négation) ?
  À la place de $(P \Rightarrow Q)$ on prouve $(\neg Q \Rightarrow \neg P)$,
  souvent si $Q \equiv (\neg \exists x : R(x))$ . . . (alors $\neg Q \equiv (\exists x : R(x))$).
- ☐ Par l'absurde ?
  A la place de P on prouve $(\neg P \Rightarrow \bot)$.
- ☐ Par le principe d'induction naturelle/structurelle?
  "Si $P(0)$ et $(P(n) \Rightarrow P(n{+}1))$, alors $\forall n \in \mathbb{N} : P(n)$.
- ☐ . . .

Degré de formalisation

Il arrive souvent que l'on quitte le domaine formel,
parce que l'on ne veut pas re-démontrer des faits "évidents"
que l'on a, par exemple, déjà appris à l'école.

Par exemple, l'application de "si A, alors $A \vee B$" est souvent considéré comme étant vrai de manière implicite.

En tout cas, il faut clairement indiquer chaque instance d'un fait que l'on assume d'être connu.

A observer dans les exemples et les exercices du cours . . .

Où s'arrêter?

- ☐ à quel niveau de détail ?
- ☐ quelles axiomes sont permis ?

Un des buts de faire des preuves est de montrer qu'une conclusion est valide sous certaines conditions.
Ces conditions doivent souvent être déterminées, et elles ne sont pas données dès le début ! On cherche plutôt les premisses les plus faibles qui sont suffisamment forts pour démontrer la conclusion.

Cette interprétation est très proche de ce qui se passe dans la recherche. Une fois terminé, on présente évidemment les théorèmes de manière complète, avec toutes les conditions et suppositions indispensables que l'on a trouvé (en faisant des preuves rigoureuses).

Exemple 1: $|\mathbb{N}| = |\mathbb{Q}|$ par construction

Théorème: Il existe une bijection $\mathbb{N} \to \mathbb{Q}$.

$\overset{\text{def}}{\Longleftrightarrow}$ $(\exists d : \mathbb{N} \to \mathbb{Q}) : d$ est surjective et injective

$$\overset{\text{def}}{\Longleftrightarrow} (\exists d : \mathbb{N} \to \mathbb{Q}) \; : \; \begin{array}{l} (\forall q \in \mathbb{Q} : \exists n \in \mathbb{N} : d(n) = q) \\ \wedge \;\; (\forall n_1, n_2 \in \mathbb{N} : n_1 \neq n_2 \Rightarrow d(n_1) \neq d(n_2)) \end{array}$$

Stratégie: construction de la fonction d,
suivi par la preuve qu'elle possède les propriété désirées.

Idée de la construction:
énumérer tous les nombres rationnels possibles.
Pour cela: se souvenir de la définition des nombres rationnel . . .

Attention dans la preuve de l'injection . . .

Exemple 2: Contradiction

Définition:
Un ensemble x est *fini* s'il existe un nombre naturel n ($\in \mathbb{N}_0$) égal au nombre d'éléments de x.
Dans ce cas, on écrit $\|x\| = n$.
Si x n'est pas fini, on dit qu'il est *infini*.

Soit x, y, z des ensembles. Soit $x \subseteq z$ et $y \subseteq z$.
On appelle y le *complément de x par rapport à z*,
si $x \cup y = z$ et si $x \cap y = \emptyset$.

Théorème:
Soit $x \subseteq z$, x fini et z infini.
Soit y le complément de x par rapport à z.
Alors, y est infini.

(A utiliser: $x \cap y = \emptyset \Rightarrow \|x \cup y\| = \|x\| + \|y\|$.)

Ensembles infinis

Comment démontrer les résultats suivants?

Conjecture:
$\forall n \in \mathbb{N} : n^2 + n + 41$ est un nombre premier.

Conjecture: (Euler)
$\forall a, b, c, d \in \mathbb{Z}^+ : a^4 + b^4 + c^4 \neq d^4$

Conjecture:
$\forall a, b, c \in \mathbb{Z}^+ : 313 * (a^3 + b^3) \neq c^3)$

Conjecture: (Goldbach)
$\forall a \in \mathbb{Z}^+ : (a > 2 \wedge 2|a \Rightarrow (\exists b, c \in \mathbb{Z}^+ : b, c$ premiers $\wedge\, a = b + c))$

Définition inductive

- ☐ Par une *grammaire*: $n \;\; ::= \;\; \text{zéro} \;\; | \;\; \text{suc}(n)$
- ☐ Par des *règles d'inférence*:

$$\textsc{zero}\ \frac{}{\text{zéro} \in \mathbb{N}} \qquad\qquad \textsc{suc}\ \frac{n \in \mathbb{N}}{\text{suc}(n) \in \mathbb{N}}$$

Normalement, on utilise 0 à la place de zéro,
et n à la place de $\underbrace{\text{suc}(\cdots \text{suc}(}_{n}\text{zéro})\cdots)$.

- ☐ Définition inductive des fonctions sur $\mathbb{N}$:

$$s(n) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{si } n = 0 \ (n = \text{zéro}) \\ n * s(n-1) & \text{si } n > 0 \ (n = \text{suc}(\hat{n})) \end{cases}$$

Induction Naturelle et Structurelle

- ☐ Soit $\forall n \in \mathbb{N} : P(n)$ un prédicat sur les nombres naturels.
- ☐ Le principe de l'induction naturelle est que
si l'on arrive à prouver
  - que $P(0)$ est vrai, et
  - que $P(n) \Rightarrow P(n+1)$ est vrai,

  alors $\forall n \in \mathbb{N} : P(n)$ est vrai.
- ☐ Ce principe se généralise à d'autres structures de données comme, par exemple, les listes.
Celles-ci peuvent être définies par

$$l \;\; ::= \;\; \text{nil} \;\; | \;\; \text{cons}(n, l)$$

où n représente un élément, dans ce cas: un nombre naturel.

Induction “sur des règles”

- ☐ Soit l'ensemble X défini
par un ensemble $\mathcal{R}$ d'axiomes A_i et de règles (propres) R_j:

$$A_i\ \frac{}{y_i \in X}\ \textsc{cond}(A_i) \qquad\qquad R_j\ \frac{x_1 \in X, \ldots, x_{n_j} \in X}{y_j \in X}\ \textsc{cond}(R_j)$$

(Il peut y avoir des conditions pour chaque élément de $\mathcal{R}$.)

- ☐ Le principe de preuve correspondant est alors:
On peut déduire que $(\forall x \in X)(P(x))$
  - si pour tout axiome A_i: $\text{cond}(A_i) \Rightarrow P(y_i)$, et
  - si pour toute règle R_j:
  $\text{cond}(R_j) \wedge (\forall 1 \leq k \leq n_j : P(x_k)) \Rightarrow P(y_j)$.

Ordres

Définition:
Soit X un ensemble. Soit $< \subseteq X \times X$ et $\leq \subseteq X \times X$.

- ☐ $(X, \leq)$ est un *pré-ordre*, si $\leq$ est *transitive*:

$$\forall x_1, x_2, x_3 \in X : (x_1 \leq x_2 \wedge x_2 \leq x_3) \Rightarrow x_1 \leq x_3$$

- ☐ Un pré-ordre $(X, \leq)$ est *réflexif*, si $\forall x \in X : x \leq x$.
- ☐ Un pré-ordre $(X, <)$ est *strict*, si $\forall x \in X : \neg(x < x)$.
- ☐ Un pré-ordre $(X, \leq)$ est un *ordre (partiel)*,
si $\leq$ est réflexive et *anti-symétrique*:

$$\forall x_1, x_2 \in X : (x_1 \leq x_2 \wedge x_2 \leq x_1) \Rightarrow x_2 = x_1$$

- ☐ Un pré-ordre $(X, \leq)$ est *total* si

$$\forall x_1, x_2 \in X : x_1 \leq x_2 \vee x_2 \leq x_1 \vee x_1 = x_2$$

Ordre lexicographique

Définition: Soit A un alphabet fini.
L'ensemble A^* des mots sur A est défini par

$$A^* \stackrel{\text{def}}{=} \{\epsilon\} \cup \{\, \omega \mid \omega = a_1 \cdot \ldots \cdot a_n \ \wedge \ 0 \neq n \in \mathbb{N} \ \wedge \ a_i \in A \,\}$$

où $\omega_1 \cdot \omega_2$ est la concaténation (associative) de deux mots
et les mots de longueur 1 sont représentés par leur symbole.
NB: $(A^*, \cdot, \epsilon)$ est un monoïde, donc $\omega \cdot \epsilon = \epsilon \cdot \omega = \omega$.

Définition: Soit $\prec$ un ordre total sur un alphabet A fini.
Soit $\omega_1, \omega_2 \in A^*$ avec $\omega_1 = a_{11} \cdot \ldots \cdot a_{1m_1}$ et $\omega_2 = a_{21} \cdot \ldots \cdot a_{2m_2}$.
ω_1 est lexicographiquement plus petit que ω_2 — $\omega_1 \lll_l \omega_2$ — si

- ☐ $m_1 < m_2$, ou
- ☐ $m_1 = m_2$ et il y a un $i \leq m_1$ tel que:
  - $a_{1j} = a_{2j}$ pour tout $j < i$, et
  - $a_{1i} \prec a_{2i}$.

Lemme

Soit $\omega_1 \neq \omega_2 \in A^*$.
Alors,

1. il existe $\omega \neq \epsilon$ tel que $\omega_2 = \omega_1 \cdot \omega$, ou
2. il existe $\omega \neq \epsilon$ tel que $\omega_1 = \omega_2 \cdot \omega$, ou
3. il existe $\omega, \omega_1', \omega_2' \in A^*$ et $a_1, a_2 \in A$

   tel que $\begin{array}{rcl} \omega_1 & = & \omega \cdot a_1 \cdot \omega_1' \\ \omega_2 & = & \omega \cdot a_2 \cdot \omega_2' \end{array}$ et

   (a) $a_1 \prec a_2$, ou
   (b) $a_2 \prec a_1$.

Ordre lexicographique

- ☐ Soit $(A, \prec)$ un alphabet ordonné dont l'ordre est strict.
- ☐ Nous redéfinissons l'ordre lexicographique $(A^*, \lll_l)$
par les règles suivantes (pour $\omega, \omega_1, \omega_2 \in A^*$ et $a_1, a_2 \in A$):

$$\text{L1}\ \frac{}{\epsilon \lll_l \omega}\ \omega \neq \epsilon \qquad\qquad \text{L2}\ \frac{}{\omega \cdot a_1 \lll_l \omega \cdot a_2}\ a_1 \prec a_2$$

$$\text{L3}\ \frac{\omega_1 \lll_l \omega_2}{\omega_1 \cdot a_1 \lll_l \omega_2 \cdot a_2}$$

- ☐ Exemple: $A = \{\, a \prec b \,\}$
  - Prouver que $\epsilon \not\lll_l \epsilon$.
  - Prouver que $a \lll_l a \cdot b$.
  - Prouver que $b \lll_l a \cdot b$.
  - Prouver que $a \cdot a \cdot b \not\lll_l a \cdot b$.

Ordre alphabétique

- ☐ Soit $(A, \prec)$ un alphabet strictement ordonné.
- ☐ Nous définissons l'ordre alphabétique $(A^*, \lll_d)$
par les règles suivantes (pour $\omega, \omega_1, \omega_2 \in A^*$ et $a_1, a_2 \in A$):

$$\text{D1}\ \frac{}{\epsilon \lll_d \omega}\ \omega \neq \epsilon \qquad\qquad \text{D2}\ \frac{}{a_1 \cdot \omega_1 \lll_d a_2 \cdot \omega_2}\ a_1 \prec a_2$$

$$\text{D3}\ \frac{\omega_1 \lll_d \omega_2}{a \cdot \omega_1 \lll_d a \cdot \omega_2}$$

- ☐ Exemple: $A = \{\, a \prec b \,\}$
  - Prouver que $\epsilon \not\lll_d \epsilon$.
  - Prouver que $a \lll_d a \cdot b$.
  - Prouver que $b \not\lll_d a \cdot b$.
  - Prouver que $a \cdot a \cdot b \lll_d a \cdot b$.

Fonctions et images

- Une *fonction* $f : A \to B$ est une relation $f \subseteq A \times B$ telle que
  - si $(x, y_1) \in f$ et $(x, y_2) \in f$ alors $y_1 = y_2$, et
  - pour tout $x \in A$ il existe un $y \in B$ tel que $(x, y) \in f$ (ceci est la propriété de totalité pour les fonctions).
- L'*image* (aussi: *portée*) d'une fonction totale $f : A \to B$ est l'ensemble noté $\mathrm{image}_{AB}(f)$ (ou simplement $\mathrm{image}(f)$) qui est défini par $\mathrm{image}_{AB}(f) = \{ f(x) \mid x \in A \}$ ($\stackrel{\text{def}}{=} f(A)$).
- Noter que si $f : A \to B$, alors $\mathrm{image}_{AB}(f) \subseteq B$ (ou $\mathrm{image}_{AB}(f) \in 2^B$).
- Par conséquent, image_{AB} est une fonction !

$$(A \times B) \times 2^B \qquad\qquad (A \to B) \to 2^B$$

Exemple 3: $|\mathcal{F}_{\mathbb{N}}| \geq |2^{\mathbb{N}}|$

- La proposition: Il existe une surjection "de gauche à droite":

$$(\exists g : \underbrace{(\mathbb{N} \to \mathbb{N})}_{\mathcal{F}_{\mathbb{N}}} \to 2^{\mathbb{N}})\ (\forall N \subseteq 2^{\mathbb{N}})\ (\exists f : \mathbb{N} \to \mathbb{N})\ (g(f) = N)$$

- La fonction $\begin{array}{rccl} c: & \mathcal{F}_{\mathbb{N}} & \to & 2^{\mathbb{N}} \\ & f & \mapsto & \mathrm{image}_{\mathbb{N}\mathbb{N}}(f) \end{array}$
  associe à toute fonction (totale) f sur les nombres naturels l'*image* de f. $(\forall f)(c(f) = \mathrm{image}_{\mathbb{N}\mathbb{N}}(f) = f(\mathbb{N}))$.

Exercice:

- La fonction c n'est pas une surjection. Pourquoi?
- Alors, essayer d'adapter la définition de la fonction c …

Conclusion: Rigueur!

- preuve "informelle" $\neq$ preuve non-rigoureuse
- Il faut tout de même réfléchir sur la sémantique des symboles avant de pouvoir enclencher la "machine déductive"