De l'universalité de la vérité mathématique à l'interopérabilité des systèmes de vérification des démonstrations

Gilles Dowek

Comment ajouter deux nombres entiers exprimés en base dix ?

Ajouter les unités avec les unités, puis les dizaines avec les dizaines, etc., en propageant la retenue

Dans un langage, plutôt que dans une langue

```
let rec add n m c =
  if n == [] && m == [] && c == Zero
  then []
  else let (c',s) = add_digits (unit n) (unit m) c
       in s::(add (rest n) (rest m) c')
```

Dans un langage, plutôt que dans une langue

```
let add p q =
  let n = ref p
  in let m = ref q
  in let c = ref Zero
  in let res = ref []
  in while not (!n == [] && !m == [] && !c == Zero) do
     let (c',s) = add_digits (unit !n) (unit !m) !c
     in n := rest !n;m := rest !m; c:= c';res := !res@[s]
   done;
  !res
```

Pourquoi ?

Les mots de la langue « unité », « dizaine », « retenue », « etc. », etc. n'ont pas d'équivalents dans le langage : il faut les expliquer avec les mots du langage : des choix

Ici « Ajouter les unités avec les unités, puis les dizaines avec les dizaines, etc., en propageant la retenue » : récursivité (construction de point fixe) ou boucle (répétition d'une même opération)

Plusieurs programmes, mais aussi plusieurs langages

```
def add(n,m):
  c = '0'
  res = []
  while not ((n == []) & (m == []) & (c == '0')):
     (c1,s) = add_digits(unit(n),unit (m),c)
     n = rest(n)
     m = rest(m)
     c = c1
     res = res + [s]
  return res
```

Est-ce embêtant ?

Peut-être l'algorithme de l'addition peut-il s'exprimer dans le langage X, mais non dans le langage Y...

Dans le cas des langages de programmation : ce n'est pas le cas

Traductions entre langages

Mais ce qui nous intéresse aujourd'hui

Les langages d'expression de démonstrations mathématiques

Il existe x tel que $x + 3 = 5$.
$\exists x\ x + S(S(S(0))) = S(S(S(S(S(0)))))$

On sait que $2 + 3 = 5$, donc $x = 2$ convient.

$\langle S(S(0)), \mathit{refl}\ S(S(S(S(S(0))))) \rangle$

D'où vient cette idée bizarre ?

L'indécidabilité de la logique de prédicat (Church-Turing, 1936)

> Il n'y a pas d'algorithme qui prend en argument une proposition A et retourne 1 ou 0, selon qu'il existe une démonstration π de A ou non

Mais

> Il y a un algorithme qui prend en arguments une proposition A et une démonstration π et retourne 1 ou 0, selon que π est une démonstration de A ou non

Dès que nous avons eu des ordinateurs assez puissants (1967) : « faisons-le »
(De Bruijn, Milner)

Et comme pour les algorithmes...

Exprimer les démonstrations mathématiques dans un langage (plutôt que dans une langue)

Pourquoi pas logique des prédicats + théorie des ensembles ?
Pas de termes autres que les variables : $\emptyset$, $\mathbb{N}$, 3, $\mathcal{P}(\mathbb{N})$, 2×3, $\{x \in \mathbb{N} \mid \exists y\ x = 2 \times y\}$...

$$\emptyset \in \{\emptyset\}$$

$$\forall x \forall y\ ((\forall z\ (\neg z \in x)) \Rightarrow (\forall v\ (v \in y \Rightarrow (\forall w\ (\neg w \in v)))) \Rightarrow x \in y)$$

Pas conçue pour cela : exprimer les mathématiques en principe vs. en fait

Un grande créativité

- Une variante de la théorie des ensembles (avec des termes) : `Mizar`, `Isabelle/ZF`...
- Une variante d'une concurrente de la théorie des ensembles : la théorie de types simples : `HOL`, `Isabelle/HOL`, `HOL Light`, `PVS`...
- Des théories plus innovantes : `Automath`, `Coq`, `Agda`, `Lean`...
- Des théories plus spécialisées : `LCF`, `ACL`...

Et comme pour les programmes

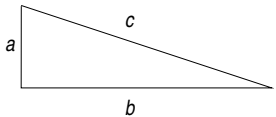
Une démonstration du petit théorème de Fermat
Une démonstration `Coq` du petit théorème de Fermat
Une démonstration `PVS` du petit théorème de Fermat

Une démonstration `Coq` du théorème des quatre couleurs (Gonthier-Werner)
Et une démonstration `PVS` ?

Est-ce embêtant ?

Cette fois oui : des propositions qui ont une démonstration en `Coq` mais pas en `HOL Light`

Mais... l'universalité est constitutive de la vérité mathématique



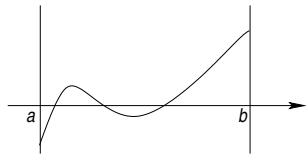
pour les Sumériens $a^2 + b^2 = c^2$, pour les Chinois $a^2 + b^2 = c^2$

Et si plusieurs mathématiques, laquelle utiliser quand nous fabriquons un avion ou un vaccin ?

À chacun ses mathématiques : impossible

Revenir à la langue ?

Impossible : un progrès irréversible dans l'histoire de la rigueur (après Euclide, Russell et Whitehead, Bourbaki)

- 

$$(\forall x\,(a \leq x \leq b \Rightarrow \text{continue}(f,x)) \wedge f(a) < 0 \wedge f(b) > 0) \Rightarrow \exists x\,(a \leq x \leq b \wedge f(x) = 0)$$

- Des démonstrations longues car collectives : classification des groupes simples finis (500 articles, plus de 100 auteurs, sur 30 ans)
- Des démonstrations longues car instrumentées : théorème de Hales (Kepler)
- Des démonstrations non relues car spécialisées et rébarbatives : Sats, seL4 (Heiser et al.), CompCert (Leroy et al.)

Est-ce la première crise de l'universalité de la vérité mathématique ?

- l'incomensurabilité de la diagonale et du côté du carré : *Il existe un nombre dont le carré est égal à* 2
- les complexes : *Il existe un nombre dont le carré est égal à* -1
- les sommes infinies : $\sum_n \frac{1}{2^n} = 2$, $\sum_n (-1)^n = 0$
- les infinitésimaux : *Il existe des nombres infiniment petits*
- les géométries non euclidiennes : *La somme des angles d'un triangle est égale à l'angle plat*
- l'axiome du choix : *Tous les espaces vectoriels ont une base*
- la constructivité : *Si* $A \cup B$ *est infini, alors* A *est infini ou* B *est infini*

Comment ces crises précédentes ont-elles été résolues ?

Une grande variété de solutions

Parmi lesquelles : des théories axiomatiques : E et H
Tomber d'accord sur le fait que

$$E \vdash \textit{la somme des angles d'un triangle est égale à l'angle plat}$$

mais pas

$$H \vdash \textit{la somme des angles d'un triangle est égale à l'angle plat}$$

La notion de cadre logique

- A vraie $\longrightarrow \Gamma \vdash A$ vrai
- Conditions de vérité : pour les énoncés de la géométrie $\longrightarrow$ pour des séquents arbitraires
- Séparation entre la définition des conditions de vérité d'un séquent : le cadre logique et la définition de diverses géométries comme théories
- Un cadre logique : la logique des prédicats (1928)
- Les différentes géométries (et l'arithmétique, et la théorie des ensembles...) définies dans ce cadre logique

Un projet possible

Exprimer les théories de `HOL Light`, `PVS`, `Coq`, `Lean`... dans la logique des prédicats

- (avec un peu de chance) beaucoup d'axiomes communs et quelques-uns distinguant les théories
- (avec un peu de chance) des axiomes compatibles
- une possibilité d'analyse des axiomes utilisés dans chaque démonstration (à rebours, axiome du choix...)
- une possibilité d'améliorer les démonstrations en utilisant moins d'axiomes (à rebours, constructivisation...)

Marche à peu près pour la théorie de `HOL Light` (Davis, Henkin (entre les lignes))

Un projet possible

Exprimer les théories de `HOL Light`, `PVS`, `Coq`, `Lean`... dans un cadre logique

- (avec un peu de chance) beaucoup d'axiomes communs et quelques-uns distinguant les théories
- (avec un peu de chance) des axiomes compatibles
- une possibilité d'analyse des axiomes utilisés dans chaque démonstration (à rebours, axiome du choix...)
- une possibilité d'améliorer les démonstrations en utilisant moins d'axiomes (à rebours, constructivisation...)

Au delà de la logique des prédicats

En un siècle : quelques limitations

D'autres cadres logiques : λ-Prolog, Isabelle, *the Edinburgh logical framework*, les systèmes de types purs, la déduction modulo théorie, la logique œcuménique, `Dedukti`

Dans `Dedukti`

- Les symboles de fonction peuvent lier des variables (comme en λ-Prolog, Isabelle, *the Edinburgh logical framework*)
- Les démonstrations sont des termes (comme dans *the Edinburgh logical framework*)
- La déduction et le calcul sont distingués (comme en déduction modulo théorie)
- Une bonne notion de coupure (comme en déduction modulo théorie)
- Les démonstrations constructives et classiques peuvent être exprimées (comme en logique œcuménique)

La théorie des types simples en Dedukti

$$
\begin{array}{rcl}
Set & : & Type \\
El & : & Set \to Type \\
o & : & Set \\
\iota & : & Set \\
\rightsquigarrow & : & Set \to Set \to Set \\
Prf & : & (El\ o) \to Type \\
\Rightarrow & : & (El\ o) \to (El\ o) \to (El\ o) \\
\forall & : & \Pi x : Set\ (((El\ x) \to (El\ o)) \to (El\ o))
\end{array}
$$

$$
\begin{array}{rcl}
(El\ (\rightsquigarrow\ x\ y)) & \longrightarrow & (El\ x) \to (El\ y) \\
(Prf\ (\Rightarrow\ x\ y)) & \longrightarrow & (Prf\ x) \to (Prf\ y) \\
(Prf\ (\forall\ x\ y)) & \longrightarrow & \Pi z : (El\ x)\ (Prf\ (y\ z))
\end{array}
$$

Le Calcul des constructions en Dedukti

$$
\begin{array}{rcl}
Set & : & Type \\
El & : & Set \to Type \\
o & : & Set \\
\iota & : & Set \\
\rightsquigarrow & : & \Pi x : Set\ ((El\ x) \to Set) \to Set) \\
Prf & : & (El\ o) \to Type \\
\Rightarrow & : & \Pi x : (El\ o)\ (((Prf\ x) \to (El\ o)) \to (El\ o)) \\
\forall & : & \Pi x : Set\ (((El\ x) \to (El\ o)) \to (El\ o)) \\
\pi & : & \Pi x : (El\ o)\ (((Prf\ x) \to Set) \to Set)
\end{array}
$$

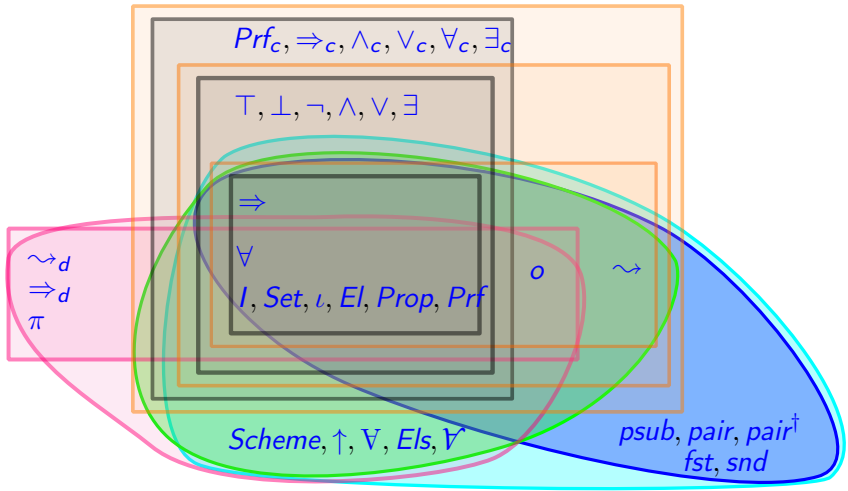
$$
\begin{array}{rcl}
(El\ (\rightsquigarrow\ x\ y)) & \longrightarrow & \Pi z : (El\ x)\ (El\ (y\ z)) \\
(Prf\ (\Rightarrow\ x\ y)) & \longrightarrow & \Pi z : (Prf\ x)\ (Prf\ (y\ z)) \\
(Prf\ (\forall\ x\ y)) & \longrightarrow & \Pi z : (El\ x)\ (Prf\ (y\ z)) \\
(El\ (\pi\ x\ y)) & \longrightarrow & \Pi z : (Prf\ x)\ (El\ (y\ z))
\end{array}
$$

Mathématiques à rebours en `Dedukti`

Trois fonctionnalités en plus dans le Calcul des constructions

Toutes les démonstrations de la théorie des types simples peuvent être traduites dans le Calcul des constructions

Les démonstrations du Calcul des constructions qui n'utilisent pas ces fonctionnalités peuvent être traduites dans la théorie des types simples



Plus d'axiomes : univers, le polymorphisme d'univers, la prédicativité, les types inductifs, la théorie cubique des types (Barras), la théorie des ensembles (Traversié)

- À rebours

  Le premier livre des Éléments d'Euclide en `Coq` $\longrightarrow$ dans la logique des prédicats (Géran)

  Le petit théorème de Fermat en `Matita` $\longrightarrow$ dans la théorie des types simples (Thiré)

  Le « postulat » de Bertrand en `Matita` $\longrightarrow$ dans la théorie prédicative des types (Felicíssimo)

- Interopérabilité

  Le premier livre des Éléments d'Euclide en `Isabelle/HOL`, `TSTP`...

  Le petit théorème de Fermat en `Isabelle/HOL`, `HOL Light`, `Coq`, `Lean`, `PVS`...

  Le « postulat » de Bertrand en `Agda`

À l'horizon

Des démonstrations qui utilisent des axiomes parfois différents
Mais indépendantes des logiciels utilisés pour les construire (l'esprit `Algol`)

Rigueur ET universalité