Alain Giorgetti

Formaliser la démonstration.

Thématique de recherche : Vérification déductive de programmes

↳ sur papier / sur machine
la preuve papier préexiste à la preuve machine.

Q : Comment publier de nouveaux théorèmes ?

Cadre : calcul symbolique / résultats exacts / dans une théorie.

Domaine : math / physique th / informatique th.

Y a-t-il des erreurs ? – Dans la démo ...
– Dans la publication ...

La 1re étape : publier pour vérifier.

Exemple : Microsoft a le plus grand # de testeurs du monde !

Puis : mode d'évaluation par les pairs.

Usage : de calcul formel.

: preuve automatique.

Démonstration longue / démonstration longue.

Ex personnel : Ei n'a pas été défini ... la formule ne fonctionne pas si $q=0$ et $r=?$.
(article J. Comb Th B 1999)

T. Webb a trouvé cette erreur ; il se convainc de la correction en le programmant !

Il reste la Q : comment publier une démonstration longue ?

Qu'est-ce qu'une démonstration

Démontrer = raisonner + calculer.

> axiomes + règles de réduction.

(Dowek, qui pense que cela remonte aux années 30)

raisonner : indécidable — activité humaine — notion
calculer : décidable — machine — ignorée

Cela est-il quantifiable ?

La machine fait la partie calculatoire de la preuve.

Poincaré : tout travail math doit contenir suffisamment d'idées pour que l'intuition math puisse se développer.

Mathématiciens : Système de 7 éq à 7 inconnues : que des calculs. (pivot de Gauss).

1% raisonnement (100-1)% calcul

Mécaniser le calcul, c'est réduire 1.

Problème : la décidabilité peut n'être que théorique : il faut tenir compte de la complexité algorithmique.

Approfondissons chaque notion : le système formel = axiomes + règles de déduction.

ex de système : logique dynamique avec substitutions explicites

[La th des ensembles est un système formel sans interprétation.

C'est une grande ≠ avec les systèmes formels de l'informatique].

→ 5 règles : a la puissance de machine de Turing.

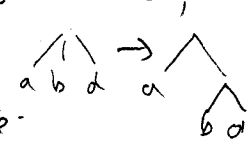
[Le "while" ne peut être implémenté dans toute sa généralité]

Calcul = règles + stratégies

Une règle. ex : $a \cdot b + a \cdot c \longrightarrow a \cdot (b+c)$

Il y a 2 filtrages : - syntaxique
- par associativité + commutativité

On peut aussi normaliser : ordonner les lettres ;

mettre des parenthèses 

Il y a des axiomes qui impliquent la commutativité.

... au 3e3, bcp plus difficile.

ex : Associativité + $(ab)^3 = a^3 b^3$ ⇒ Commutativité.

+ $(ab)^2 = a^2 b^2$ ⇒

On voudra - confluence : l'ordre des règles ne tient pas compte de l'infini

- terminaison : à un moment, plus aucune règle ne s'applique.

Pb : les règles ne sont pas modulaires : en rajoutant une règle, on doit tout redémontrer.

Pb de systèmes formels dans lesquels les règles sont tellement en force que...

Aucun Σ f ne montre comment il combine les règles de réécriture.

Calcul formel = manipulation symbolique.
= calcul numérique certifié (ex d'intervalles)

Objectif d'un calcul : - simplifier ?
- prouver que A = B ?
- paramétrisation ? Ce n'est pas la panacée !
- il y a un pb d'abstraction des symboles : m est traité autrement que si on le remplace par 2, 3, ...

réf : Jean-Paul Delahaye 1997 : les preuves par machine : "ce n'est encore que du calcul".

ssreflect au-dessus de Coq pour appliquer des règles "modulo" des "vues"
↳ née pour th. 4 couleurs et Feit-Thompson.
↳ a implémenté énormément de maths de base.

Pb : où publier des démos "Coq".
/ concept de tactique de preuve.