

Formalisations préliminaires en théorie des groupes finis

Assia Mahboubi

en collaboration avec

G. Gonthier, S. Ould Biha, L. Rideau, E. Tassi, L. Théry

Centre de recherche commun INRIA-Microsoft Research

GdR IM 25 janvier 2007, Montpellier

CENTRE DE RECHERCHE
COMMUN



INRIA
MICROSOFT RESEARCH

Une preuve remarquable

Theorem (Feit-Thompson, 1963)

Tout groupe d'ordre impair est résoluble.

Une preuve remarquable

Theorem (Feit-Thompson,1963)

Tout groupe d'ordre impair est résoluble.

Caractéristiques de la preuve :

- ▶ Preuve originale : un numéro entier du *Pacific Journal of Mathematics*
- ▶ Un travail collectif pour simplifier la preuve (mais pas la raccourcir)
- ▶ Wikipedia (23/01/07) : “It takes a professional group theorist about a year of hard work to understand the proof completely”

Une preuve formellement vérifiée en Coq

Point de départ : matériau de la preuve formelle en Coq du **théorème des 4 couleurs** (G. Gonthier, aidé de B. Werner, 2004) :

- ▶ Une extension langage de **tactiques**
- ▶ Des **bibliothèques** génériques :
 - ▶ Égalités décidables
 - ▶ Ensembles finis
 - ▶ Fonctions, relations, listes, ...
- ▶ Une boîte à outils pour exploiter la **réflexion à petite échelle**

Tactiques

Se donner les outils adaptés pour construire les preuves formelles.

- ▶ Une **syntaxe unifiée** autour de trois types de manipulations primitives
 - ▶ chaînage avant/chaînage arrière
 - ▶ déplacement des faits entre le contexte d'hypothèses et le but
 - ▶ réécritures, chaînes de réécritures
- ▶ Un **contrôle** pointu des occurrences
- ▶ Un langage de scripts **concis**



Statut du calcul dans la logique de Coq

Exemple : $2 + 2 = 4$

▶ Preuve par **réécriture** :

- ▶ Réécrire successivement les axiomes de la théorie des entiers naturels

$$(1 + 1) + (1 + 1) = ((1 + 1) + 1) + 1$$

▶ Preuve par **calcul** :

- ▶ Définir l'opération d'addition sur les entiers
- ▶ Utiliser l'égalité modulo calcul du système (conversion)

$$\frac{\Gamma \vdash t : T \quad T \equiv U}{\Gamma \vdash t : U}$$



Réflexion à petite échelle (SSReflect)

Exploiter la puissance du calcul pour tirer parti des différentes représentations d'un même objet

Tirer profit de la **coercion** du **type de donnée** `bool` des propriétés décidables vers la **sorte** `Prop` des énoncés de Coq

```
Coercion is_true := b = true
```

Le mécanisme de vues

En une seule opération :

- ▶ interpréter un booléen en un fait (Prop) et
- ▶ décomposer ce dernier

Compose en une commande :

- ▶ Une coercion
- ▶ Une décomposition
- ▶ Une opération primitive

Difficultés dans la formalisation

- ▶ Ne pas calquer les définitions en théorie des types sur celle de la théorie des ensembles
- ▶ Construire une hiérarchie maîtrisée de coercions
- ▶ Choisir avec soin les types de données utilisés dans les définitions



Conclusion

- ▶ État des lieux :
 - ▶ Actions, permutations, théorèmes de Sylow
 - ▶ Morphismes, quotients, théorèmes d'isomorphismes
 - ▶ Théorie de la représentation, théorème de Maschke
- ▶ Originalité de la formalisation :
 - ▶ Cohérence du développement
 - ▶ Concision des preuves
 - ▶ Bénéfice des bibliothèques adaptées aux ensembles finis

