A Hiproof Interface for Viewing and Constructing Proofs

Alan Bundy

School of Informatics
University of Edinburgh
Outline

- What are hiproofs?
- Viewing large proofs.
- Top-down proof construction.
• What are Hiproofs?

- **Higraphs**: graphs whose nodes are higraphs.
- **Hiproofs**: use of higraphs to represent proofs.
- **Nodes** represent tactics; **arcs** represent proof structure.
Two kinds of hierarchy

Induction Strategy

Step Case

Base Case

Ripple

Tactic Tree

Proof Tree

Induction

Wave

Wave

Fertilize
Viewing Proof at Different Levels of Detail

Zoom in and out of proof

24.08.09
Motivation

• Need to construct very large proofs:
  • four colour theorem, Kepler conjecture, classification of finite simple groups, Fermat’s last theorem.

• Need to view such proofs in varying levels of detail.

• Need to construct such views in modular fashion.

• Graphics is congenial vehicle for understanding.
History

• Bundy & Ireland used box diagrams informally to describe proof plans.
• Denney, Power and Tourlas developed hiproof theory.
• Various hiproof viewers developed for Dixon’s IsaPlanner.
• Aspinall, Denney & Lüth developed Hitac language and semantics.
• Plan to use hiproof proof constructor.
Hiproof Construction

- Sketch out high-level structure of as top-level hiproof.
- Incrementally unpack boxes.
  - Using existing tactics or rules.
  - Postulating new proof-specific tactics.
- Use partial hiproof as record of proof state.
- Prove theorem.
  - Run existing tactics on subgoals.
  - Manual proof of outstanding subgoals.
- Appeal to 3rd party theorems as ‘axioms’.
Fantasy: Fermat’s Last Theorem

Reductio ad absurdum

~Fermat’s Last Theorem

Epsilon Conjecture (Fry)

Elliptic Curves ≠ Modular Forms

Resolution

False

Fermat’s Last Theorem

Shimura-Taniyama Conjecture (Wiles)

Convert to Galois Representation

Iwasawa Theory

Elliptic Curves = Modular Forms
Wiles Own Graphic

Note different semantics of arcs:

Containment vs implication
Conclusion

• Hiproof graphic can assist with viewing, understanding and constructing proofs.
• Compatible with tactic-based provers.
• Implement hiproof viewer in Proof General.
• Extend to top-down hiproof constructor.
• Evaluate on large proofs.
• Connected to PhD of Iain Whiteside.