Proof Clustering for Proof Plans

Matt Humphrey

Working with:

Manuel Blum

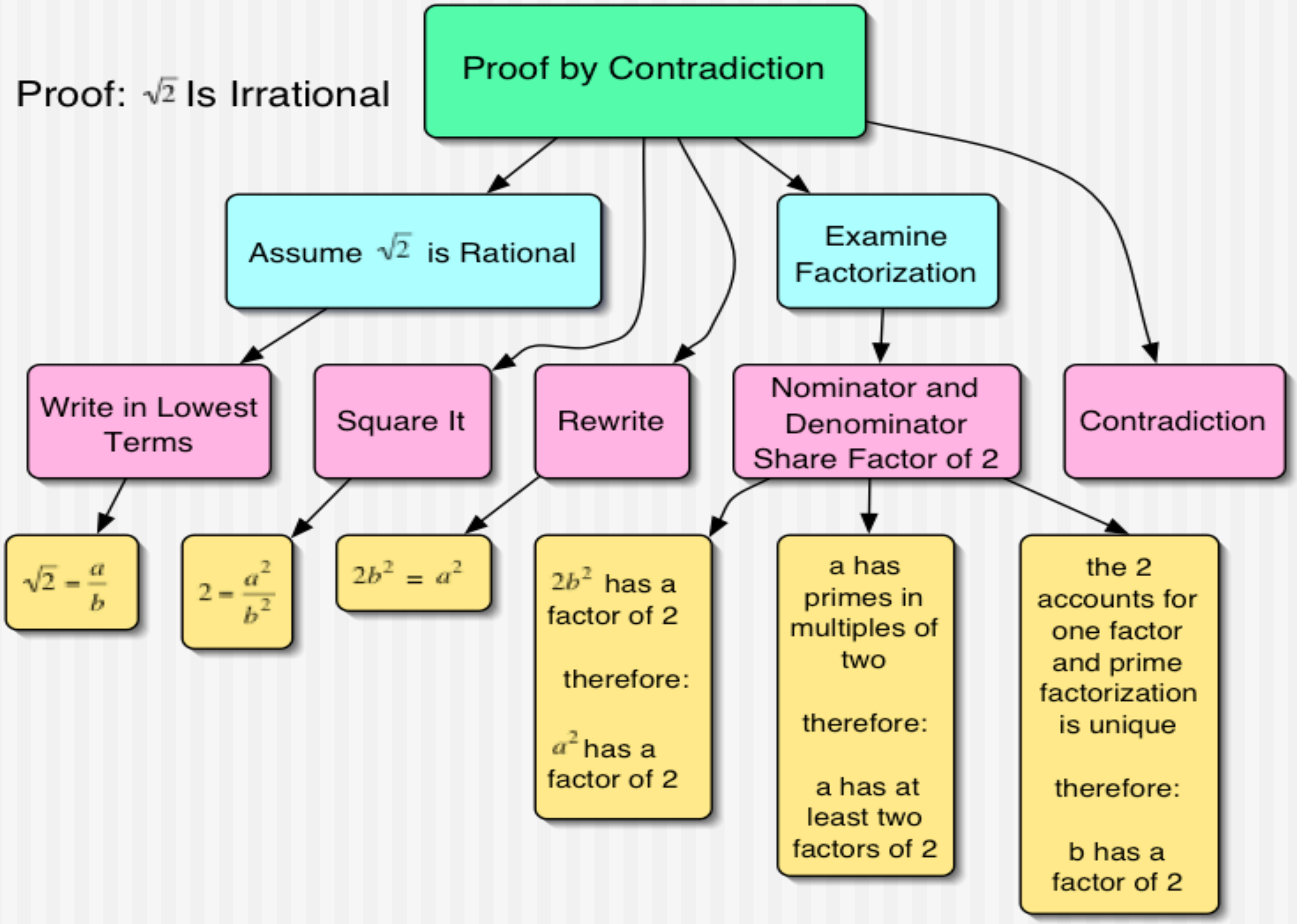
Brendan Juba

Ryan Williams

What is Proof Planning?

- Informal Definition
  - See a proof in terms of “ideas”
  - Different levels of abstraction
  - Represented as a graph, tree, DAG?
  - Tool for directing exhaustive proof search
- Formal Definition
  - No perfect all-encapsulating definition
  - Usually defined per theorem proving system
  - The concept itself is mostly informal

Example Proof Plan



Why Proof Planning?

- Cuts down proof search space
- Bridges gap between human/computer
- Proof = Guarantee + Explanation (Robinson 65)
- And…
  - It can be automated
  - It has been automated (to some extent)

Why Study This?

- Artificial Intelligence Perspective
  - What can/cannot be modeled by computer?
  - How to model something so informal
- Cognitive Psychology Perspective
  - Intuitions about human thought process
  - Reasoning about human ability to abstract
- Practical Perspective
  - Proving theorems automatically is useful

Learning by Example

- Previous proofs as hints
  - What information can be gained?
- What has been tried?
  - Analogical Reasoning
  - Strategies (higher level)
  - Methods, Tactics (lower level)
  - And in most cases a combination of these

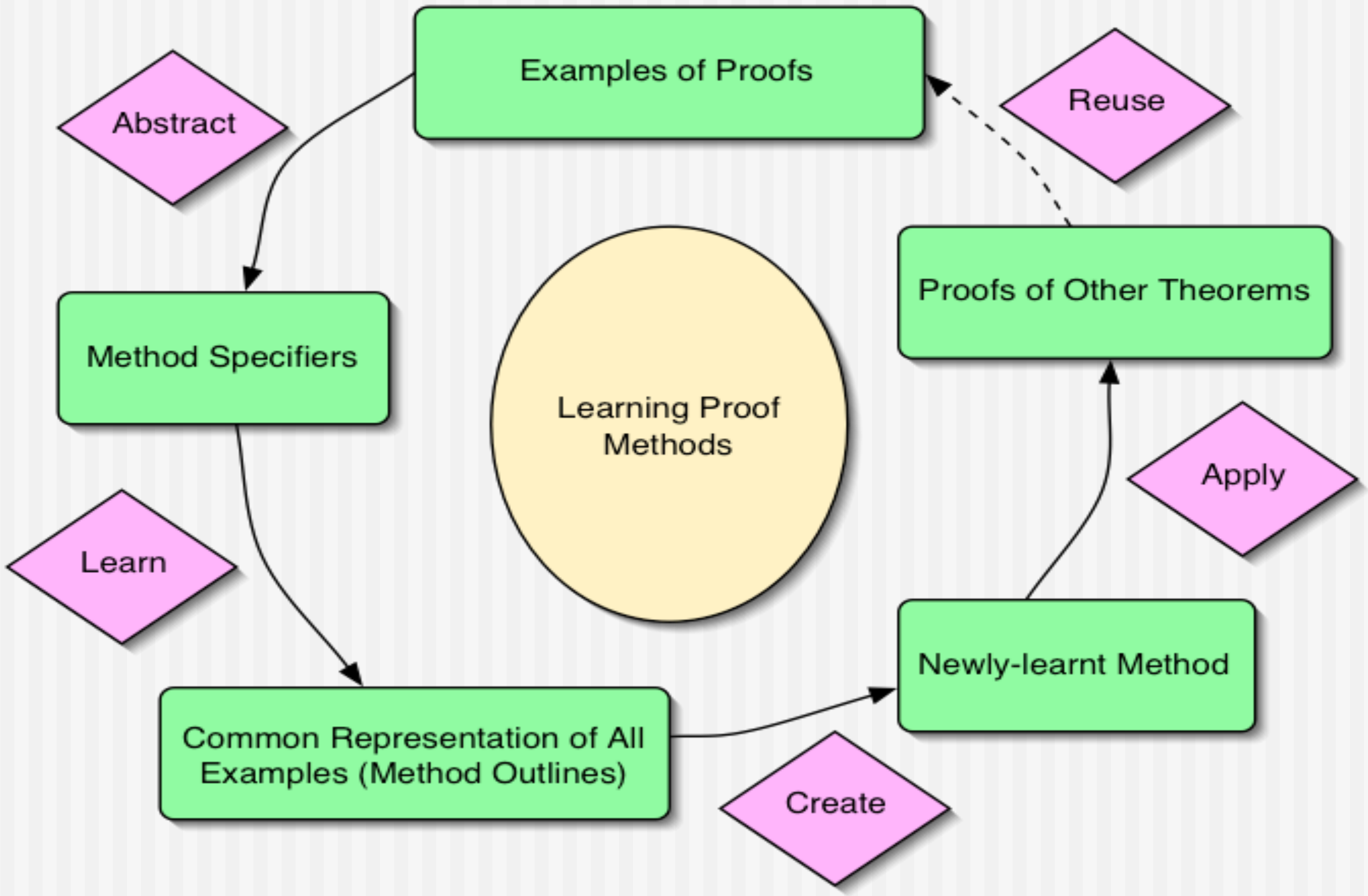
Proof Clustering

- Proof Planning can be aided by:
  - The ability to recognize similarity in proofs
  - The ability to extract information from proofs
- If we can cluster similar proofs, we can:
  - More easily generalize a strategy or tactic
  - Determine which proofs are useful examples
  - Build a proof component hierarchy
  - Automate the process of learning techniques

Ωmega Proof Planning System with LearnΩmatic

- Uses examples as tools in the proving process
- Heuristics guide the proof search
  - Uses learned proof techniques (methods)
  - Selects what it feels to be the most relevant methods
- LearnΩmatic
  - Learns new methods from sets of examples
  - Increases proving capability on the fly
  - Minimizes hard-coding of techniques
  - Increases applicability, no domain limitation

Learning Sequence



An Example of Learning…

- Extended Regular expression format
- A grouping of ‘similar’ proof techniques:

    assoc-l, assoc-l, inv-r, id-l

    assoc-l, inv-l, id-l

    assoc-l, assoc-l, assoc-l, inv-r, id-l

- …generalizes to the method:

    assoc-l*, [inv-r | inv-l], id-l

Problems with the LearnΩmatic

- Relies on ‘positive examples’ only
  - User must have knowledge about proofs
  - Hard to expand the system’s capabilities
  - Could produce bad methods for bad input
- Learning new methods is not automated!
  - Waits for the user to supply clusters

Specific Goal

- Enhance LearnΩmatic with fully automated proof clustering
  - First: be able to check a cluster for similarity
  - Second: be able to identify a ‘good’ cluster
- The results can be directly plugged into the learning algorithm for new methods
  - Proof cluster -> learning algorithm -> newly-learnt proof method -> application

Plan of Attack

- Determine what constitutes a good group
  - Maybe a simple heuristic (compression…)
  - Maybe a more detailed analysis is necessary
- Implement proof clustering on top of the LearnΩmatic system
- Collect results
  - Ideally: proving theorems on proof clustering
  - At least: empirical data from test cases

Some Questions We Have…

- Are regular expressions appropriate?
- Do Ωmega and the LearnΩmatic even represent the right approach (bottom-up)?
- How much will proof clustering aid theorem proving?
- Can we generalize proof clustering?

References

- S. Bhansali. Domain-Based program synthesis using planning and derivational analogy. University of Illinois at Urbana-Champaign, May, 1991.
- A. Bundy. A science of reasoning. In J.-L. Lassez and G. Plotkin, editors, Computational Logic: Essays in Honor of Alan Robinson, pages 178--198. MIT Press, 1991.
- A. Bundy. The use of explicit plans to guide inductive proofs. In Ninth Conference on Automated Deduction, Lecture Notes in Computer Science, Vol. 203, pages 111--120. SpringerVerlag, 1988.
- M. Jamnik, M. Kerber, M. Pollet, C. Benzmüller. Automatic learning of proof methods in proof planning. L. J. of the IGPL, Vol. 11 No. 6, pages 647--673. 2003.
- E. Melis and J. Whittle. Analogy in inductive theorem proving. Journal of Automated Reasoning, 20(3):--, 1998.