How TheoryMine Creates Novel and (Moderately) Interesting Theorems

Alan Bundy

(Joint work with Flaminia Cavallo, Lucas Dixon, Moa Johansson & Roy McCasland)

School of Informatics,
University of Edinburgh

C3GI 2012

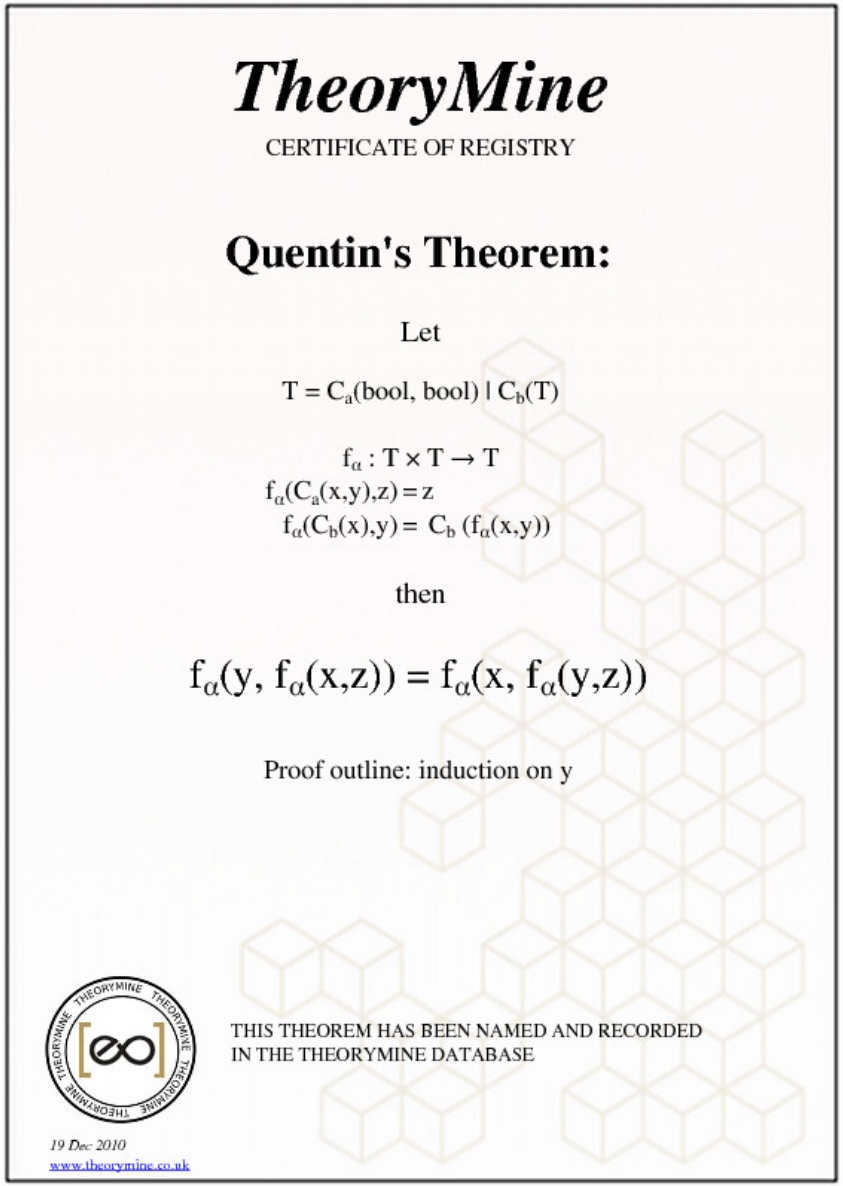
Outline

TheoryMine

theory[mine]

- TheoryMine is a spin-out company in the novelty gift market.
  - www.theorymine.co.uk
- Generates novel theorems for customers to name.
- Theorems are inductive consequences of recursively defined functions and types.
- Certificate summarises: theorem plus type and function definitions.
- Theorem purchase not new, *cf* l'Hospital's Rule.

Example Certificate



TheoryMine

CERTIFICATE OF REGISTRY

Quentin's Theorem:

Let

$$T = C_a(\text{bool}, \text{bool}) \mid C_b(T)$$

$$f_\alpha : T \times T \rightarrow T$$
$$f_\alpha(C_a(x,y),z) = z$$
$$f_\alpha(C_b(x),y) = C_b\,(f_\alpha(x,y))$$

then

$$f_\alpha(y, f_\alpha(x,z)) = f_\alpha(x, f_\alpha(y,z))$$

Proof outline: induction on y

THIS THEOREM HAS BEEN NAMED AND RECORDED IN THE THEORYMINE DATABASE

19 Dec 2010

www.theorymine.co.uk

Creating Novel Theorems

- Novel recursive mathematical objects are created, e.g.,
- Recursive mathematical functions on these objects are created, e.g.,
- Modestly interesting conjectures are generated from these functions, e.g.,
- Obviously false conjectures are filtered out.
- Inductive proof plans and critics drive interactive theorem prover to try to prove the survivors.

Creating Recursive Theories

- IsaWannaThm was UG project of Flaminia Cavallo.
- It generates a set of recursive types.
- It generates some recursive functions over these types.
- These two choices provide a recursive theory.
  - Note that theories are purely definitional, so consistent.
  - Type and function spaces are both infinite,
  - but size limits imposed to ensure tractability.
- It uses IsaCoSy to generate inductive conjectures in this theory.
  - Note that all functions must appear in each conjecture.
- IsaCoSy uses IsaPlanner to prove them.

Creating Recursive Types

- Example type definition (from certificate):

$$T \quad ::= \quad C_a(Bool, Bool) \mid C_b(T)$$

  Like four-coloured 'natural numbers': 0,1,2,..., 0,1,2,..., ...
- Grammar for generating novel recursive types.
  - Within resource limits, incrementally generate all possible types.
  - From initial set of types, e.g., *Bool* and $\mathbb{N}$.
  - Vary number of constructor functions.
  - Vary their arity and types of their arguments.
  - Filter out any already in Isabelle library.

Creating Recursive Functions

- Example function definition (from certificate):

$$\begin{aligned} f_\alpha &: T \times T \mapsto T \\ f_\alpha(C_a(x,y),z) &::= z \\ f_\alpha(C_b(x),y) &::= C_b(f_\alpha(x,y)) \end{aligned}$$

  A coloured version of addition: $f_\alpha(2,3) = 5$
  NB - number inherits colour of second argument.

- Grammar for generating simple structurally recursive functions.
  - Within resource limits, incrementally generate all possible functions.
  - Use IsaCoSy to generate function bodies.
  - Can use initial and previously defined functions.
  - Reject non-terminating functions using Isabelle's function package.

Creating Conjectures

- IsaCoSy was PhD project of Moa Johansson.
- Used by TheoryMine to generate inductive conjectures for any recursive theory, e.g.,

$$f_\alpha(x, f_\alpha(y, z)) \quad = \quad f_\alpha(y, f_\alpha(x, z))$$

  A contextual commutativity.
- TheoryMine conjectures are quantifier-free equations between irreducible terms.
- Within resource limits, incrementally generates all possible such conjectures.
- Non-theorems filtered out by Isabelle's Quickcheck counter-example finder, e.g.,

$$f_\alpha(x, y) = f_\alpha(y, x) \qquad f_\alpha(2, 3) = 5 \neq 5 = f_\alpha(3, 2)$$

- Survivors sent to IsaPlanner to be proved.

IsaCoSy's Irreducible Term Generation

- Irreducibility ensures that conjectures:
  - Are in normal form, so simplest expression of result.
  - Are not rewrite consequence of definitions and previous theorems.
  - That is, induction (or backwards rewriting) is required to prove them.
  - Therefore, have some intrinsic merit.
- Constraints ensure reducible terms are not generated.
  - Definitions and theorems are oriented as rewrite rules, e.g., $f(c(x)) \Rightarrow t$.
  - A new constraint is then imposed to ban generation of (sub-)terms of form $f(c(\ldots))$.
  - Constraint set grows with new definitions and theorems.
- IsaCoSy also generates bodies of function definitions.

Guiding Inductive Proofs

- IsaPlanner was originally PhD project of Lucas Dixon.
- Uses proof planning to guide Isabelle on inductive conjectures.
  - Uses rippling and proof critics.
  - Proofs entirely automatic.
  - High success rate on simple theorems.
- Example theorems for $T + f_\alpha$ theory:

  The Richard Scott Russell Theorem:

$$f_\alpha(x, C_b(y)) = C_b(f_\alpha(x, y))$$

  The Herdman Theorem:

$$f_\alpha(f_\alpha(x, y), z) = f_\alpha(x, f_\alpha(y, z))$$

  Quentin's Theorem

$$f_\alpha(y, f_\alpha(x, z)) = f_\alpha(x, f_\alpha(y, z))$$

Proving Inductive Theorems

- Isabelle is generic, interactive proof assistant from Cambridge and Munich.
- Classical, higher-order logic most popular theory,
  - which is what we use.
- LCF-style prover with small, trusted core of logical rules,
  - provides very high level assurance of correctness.
- Tactic-driven by IsaPlanner.
- Also includes Quickcheck counter-example finder.

Explanation of Design Decisions

The following TheoryMine features are crucial to its business model:

- Restriction to purely definitional theories ensures consistency.
- Isabelle's LCF-style architecture ensures correctness.
- IsaPlanner's proof planning provides automatic proof.
- IsaCoSy's irreducibility heuristic ensures intrinsic interest of theorems.
- IsaWannaThm's meta-grammars generate a huge number of novel theories.
  - And hence theorems. Estimated at 10^{16}.

Conclusion

- Successfully generates huge numbers of novel theorems of some intrinsic merit.
- Design decisions motivated by business model.
- Recursive types, functions and theorems are simple.
- Many open conjectures generated.
- TheoryMine slows as theories become more complex.