

# Putting Theory Exploration to Work

Moa Johansson  
Chalmers University of Technology

WAIT Workshop, Amsterdam  
28 June 2018

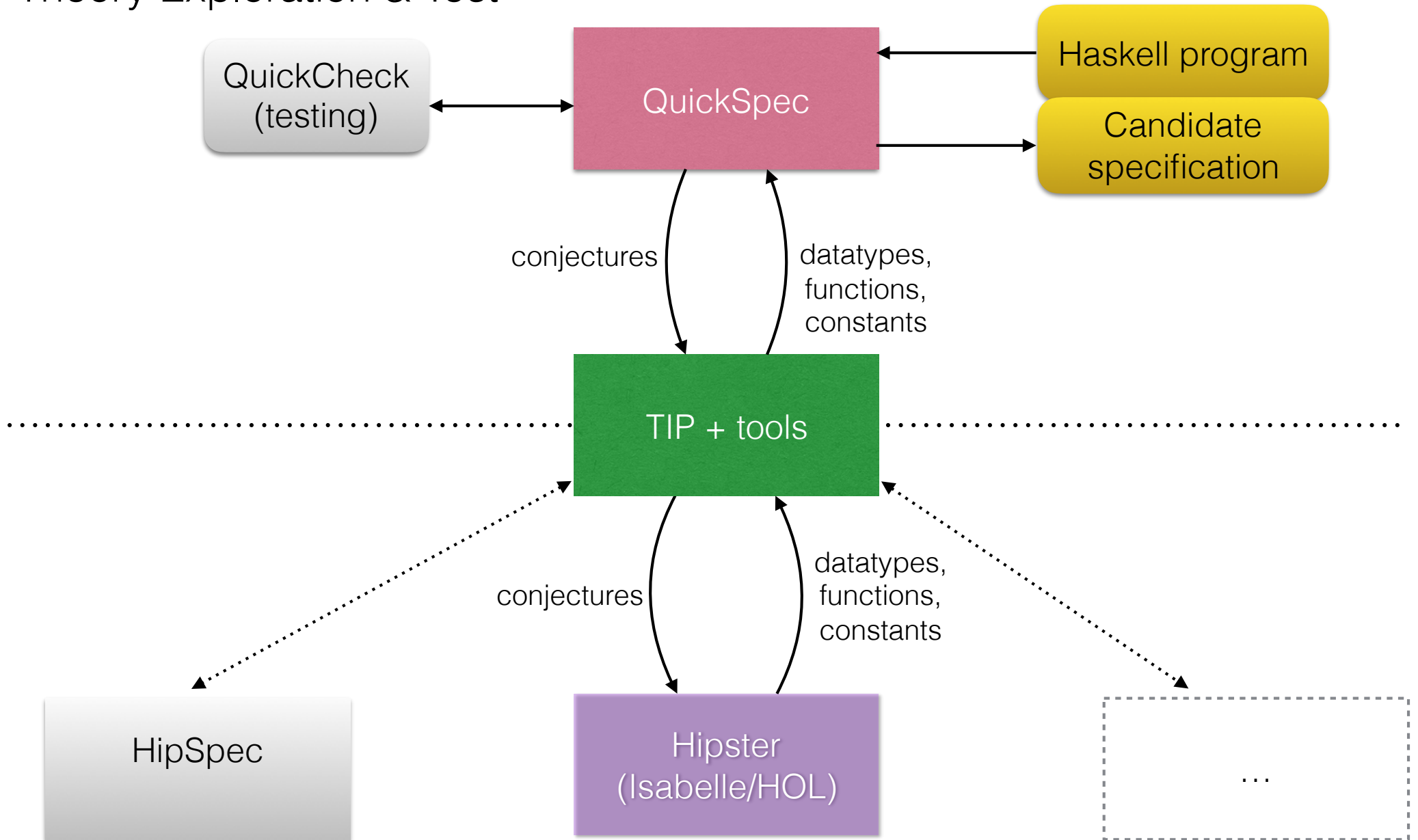
# Challenges in inductive theorem proving

Why are inductive proofs hard to automate?

- Might need **extra lemmas**, also needing induction to prove...
- Might need to **generalise**, and prove something stronger.
- **Theory Exploration** is a technique to help with this.

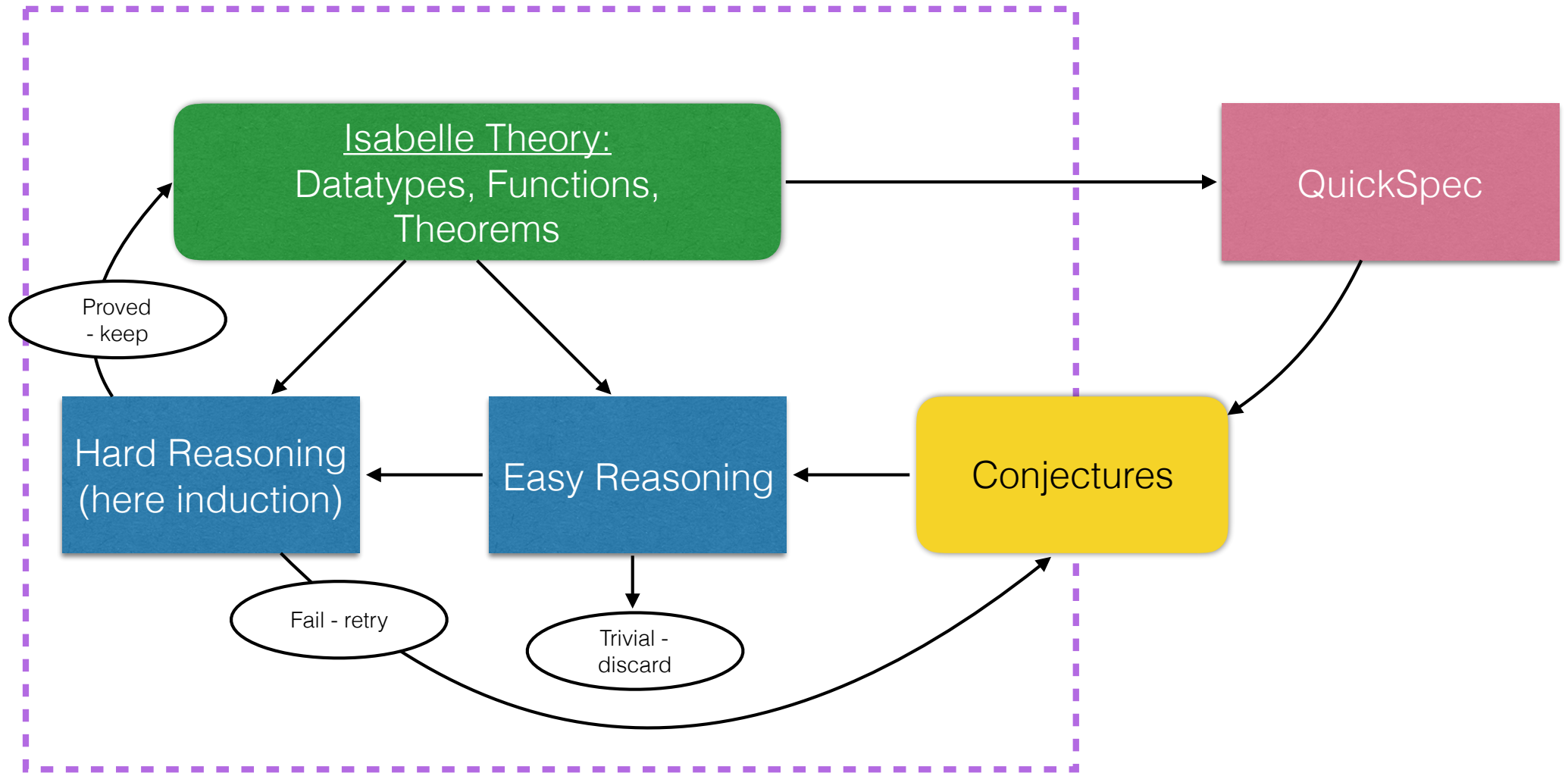
**Demo:** How to cheat on your discrete maths exercises using **Hipster**.

# Theory Exploration & Test



Provers

# How Hipster works



# Part 2:

## Towards a usable tool

# Conditionals, conditionals, conditionals...

- First go: User specified predicates (Lobo Valbuena MSc)
- Second go: Automatically extracted predicates (Algehed et. al)
- TurboSpec?
- Needs more work (better testing):

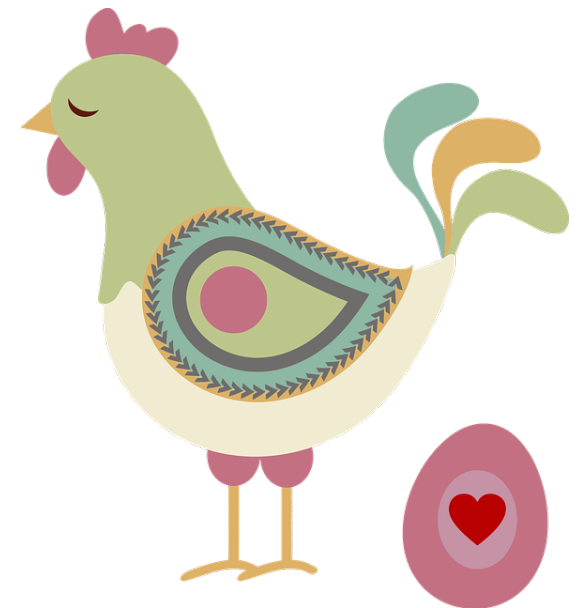
**Demo:** Insertion sort - when exploration go a bit wrong...

# Does it scale?

- Most interesting lemmas are not huge\*.
  - \*but can be very specific and have lots of pre-conditions...
- Exploration is not always the slowest part. Proofs can be.
  - Optimising proof-loop? Which tactics, when? Power vs speed.
- Limit exploration to subspaces? Schemas/templates?  
Lemmas relevant for failed proofs?

# More engineering issues

- Separation of exploration and proof: QuickSpec/Isabelle.
- Too much junk in some theories, slow proofs.
- Translations between Isabelle-TIP-Haskell.
- **Actual use-cases (and users!)**
  - Theory tips?
  - Distributing tools for Isabelle?





# Theory Exploration & Test

