Computational Logic

# Interactive Theorem Proving

## Week 6

Cezary Kaliszyk


April 21, 2015

# Summary

## So far

Proof Assistants, HOL Light, $\lambda_{\rightarrow}$, Gentzen-style, Tactics

- Properties of $\lambda_{\rightarrow}$
- BHK interpretation and $\lambda$-cube again
- Dependent types
- $\lambda_P$

## Today

- HOL Light more advanced tactics
- Natural numbers, Quotients, the library

# Arithmetic

## Definition of Natural Numbers

From the axiom of infinity

## Rewriting

- `REWRITE_TAC [ARITH]`
- What rules are being used?
- Is it complete?

## Other domains

- Real, Complex, Integer, $\mathbb{R}^n$ (vectors)
- Bootstrapped decision procedures

# MESON

- Model-Elimination
  - Loveland 1968
- How it works
  - Given helper theorems (possibly polymorphic) assume them with appropriate types
  - Try to remove occurrences of the Hilbert operator
  - Eliminate trivial assumptions
  - Beta-reduce
  - Eliminate remaining abstractions (using $\lambda$-lifting)
  - Replace `if..then..else` expressions using Disjunctions
  - For quantification expressions over booleans, consider all cases
  - Transform to NNF and Skolemize
  - Make all applications first-order
  - Translate to FOL and execute model elimination

# MESON export — monomorphisation

- Simple, but effective procedure
  - Find all polymorphic constants in the goal and the first assumption
  - For every occurrence of a constant in the goal and in the assumptions find a type instantiation
  - Apply the instantiation to the assumption and include its new instantiated constants in the goal constants
  - Repeat for all other assumptions
- May produce very big goals for set constants
- Considering all constants repeatedly can be very slow

# MESON export — first order

- Given a term like:

  ```
  MAP f [a] = [f a]
  ```

  we have the symbol `f` sometimes applied to zero sometimes one argument
- Can be encoded in FO logic like:

  ```
  MAP f [a] = [I f a]
  ```

  If we assume that identity `I` is always applied to two arguments
- For every constant or free variable we find the minimum number of arguments it is applied to
- An application of a function `F` that needs two arguments to 4 arguments is now encoded as:

  ```
  I (I (F(a1, a2), a3), a4)
  ```

# Looking at the code

- Unit type
- Quotient Package
- Pairs
- Natural numbers
- Inductive types
- Arithmetic
- Lists
- Reals
- Integers
- Sets

# Summary

## Today

- Natural numbers, datatypes
- Quotients, reals
- HOL Light library

## Next time

- How hard is $\lambda_P$
- Second order logic
- Order of variables
- $\lambda_2$