Interactive Theorem Proving

Week 11

Cezary Kaliszyk (VO)
Vincent van Oostrom (PS)

December 16, 2016

Summary

So far

Proof Assistants, HOL Light, $\lambda_\rightarrow$, λ_P, λ_2, paradoxes, set theory

- ZF(C) Axiomatization
- Bootstrapping Mathematics in set theory

Today

Mizar

Mizar History

Project started in 1973

- Attempt to reconstruct the mathematical vernacular
- In a computer environment

Today

- Formal language for writing proofs
  - Designed to be close to mathematical vernacular
  - Subset of standard English
  - Declarative style
  - Highly structured
    - rigorous and semantically unambiguous
- System for verifying the proofs
- Mizar Mathematical Library (MML)

MML

Systematic collection

- Started 1989
- Based on Tarski-Grothendieck axiomatization
- Incremental Revisions

Biggest library of formal mathematics

- 1177 articles
- 52775 theorems
- 10670 definitions
- 820 schemes
- 11333 registrations

Mizar Foundations

Logic

- classical first order logic
- second order free variables
  - for recursive definitions or induction
- natural deduction
  - usually forward reasoning

independent of the axioms of set theory

- First article: defining set and element
- Second article: axiomatization of set theory

Mizar article

- A `.miz` file (optionally `.voc` and `.abs`)
- Environment
- Theorems (local lemmas)
- Definitions
- Schemes
- Justifications
  - Simple justification
  - Proof
  - Schematization

Justifications and proofs

Local justification

```
A: statement_1;
   ...
   statement_2 by A;
```

External justification

```
   x in { x } by TARSKI:def 1;
```

Proof structure

```
statement
proof
   ...
   thus statement;
end;
```

Natural deduction (1/3)

Conjunction

```
A & B
proof
   ...
   thus A;
   ...
   thus B;
end;
```

Implication

```
A implies B
proof
   assume A;
   ...
   thus B;
end;
```

Natural deduction (2/3)

Disjunction

```
A or B
proof
   assume not A;
   ...
   thus B;
end;
```

Comment

```
:: this is a comment
```

Natural deduction (3/3)

Equivalence

```
A iff B
proof
   thus A implies B
   proof
      assume A;
      ...
      thus B;
   end;
   thus B implies A
   proof
      assume B;
      ...
      thus A;
   end;
end;
```

Predicate logic

Universal Quantification

```
for x being T holds P[x]
proof
   let x be T;
   ...
   thus P[x];
end;
```

Existential Quantification

```
ex x being T st P[x]
proof
   ...
   take x = expression;
   ...
   thus P[x];
end;
```

1.5-order logic: Schemes

```
scheme
  Ind { P[Nat] } : for k being Element of NAT holds P[k]
provided
  A1: P[0] and
  A2: for k being Element of NAT st P[k] holds P[k + 1]
proof
   ...
end;
```

1.5-order logic: Schematization

```
2 divides n * (n+1)
proof
   :: local predicate
   defpred P[Nat] means 2 divides $1 * ($1 + 1);
   a1: P[0];
   a2: for k being Nat st P[k] holds P[k + 1];
   :: refering to the scheme
   for k being Nat holds P[k] from NAT_1:sch 2(a1,a2);
   hence 2 divides n * (n + 1);
end;
```

Implementation

- Separate processes
- Parser
  - Environment
  - Operator syntax
- Analyzer
  - Disambiguation
  - Types! (adjectives)
- Checker
  - Disprover processes all disjuncts
  - Forms of a term
  - Congruence closure
- Post-processing
  - Relprem, Relinfer, ...

Mizar Syntax

27 special symbols

```
&
c=
```

110 reserved words

```
contradiction
not
or
implies
iff
for x holds a(x)
ex x st a(x)
```

Mizar Types

A type hierarchy

- `Function of X,Y`
- `PartFunc of X,Y`
- `Relation of X,Y`
- `Subset of [:X,Y:]`
- `set`

Adjectives

- Examples
  - `one-to-one Function of X,Y`
  - `finite non-empty proper Subset of X`
- Automatic deriving of type information using registrations
- Overloading of notations
- Types must be non-empty

Proof Checker

No set of inference rules

"obviousness w.r.t. an algorithm" by M. Davis

de Bruijn criterion is not preserved

- new computation mechanisms (CAS, DS)
- more automation in the equality calculus
- more general statements in an inference

Congruence closure

Is $x = y$ a consequence of $y = z$, $f(y) = z$, and $f(z) = x$?

- Monotonicity
- Transitivity
- Symmetry

Summary

Today

- Mizar project
- Foundations
- Natural deduction
- Checker and verifier

Next time

- Program extraction
- Logical frameworks