Translating Mizar for First Order Theorem Provers

Josef Urban

Dept. of Theoretical Computer Science
Charles University
Malostranske nam. 25, Praha, Czech Republic

SUMMARY

MML was exported to untyped first order syntax suitable for current efficient first order provers

More than 500 000 theorem proving problems are generated directly and generation of millions of new problems of almost arbitrary difficulty is possible.

All kinds of theorem proving experiments possible, especially

WHAT IS MIZAR?

Mizar Proof Language (Proof Style) - natural deduction (Jaskowski) - quite successful - Mizar modes for other proof assistants (Isar, HOL Light)

Mizar Mathematical Library - 763 articles, all based on common axiomatics

Mizar implementation: proof checker, other command line tools

Additional tools and presentations: MML Query (searching), Journal of Formalized Mathematics (HTML representation), Mizar mode for Emacs (authoring), etc.

WHAT IS THE AXIOMATICS OF MIZAR?

explicit: Tarski-Groethendijck set theory

implicit (implementation): Global Choice, type rules and other hard wired rules (dependent type algebras?)

no official formalizations of the Mizar logic

MOTIVATION

Largest database of formalized mathematics.

TRAIN and TEST current first order theorem provers on MML

implement THEOREM DISCOVERY systems

CURRENT STATUS OF THEOREM PROVING

several EFFICIENT untyped first order theorem provers (Otter(Eqp), Vampire, E-Setheo, SPASS) - ordered resolution, unfailing completion and superposition, tableau proving, semantic blocking

1997 Robbins theorem - proved by EQP after about 50 years of human trials

TPTP (Thousands Problems for Theorem Proving) - used for prover competitions

PREVIOUS WORK - ILF

MML in the ILF system (1997,1998) - Ingo Dahn, Czeslaw Bylinski

export of MML to Prolog format,
postprocessing by Perl and Prolog,
inserting into a SQL (Postgres) database,
Prolog library for export into various prover formats, mail server, web interface

unfortunately unfinished and hard for me to revive completely, several years ahead of other similar projects

MAIN TRANSLATION PROBLEMS

Mizar is typed, theorem provers are not; choice of the proper type translation

Mizar has additional presentation layer, users do not see directly the content (constructor) layer

Mizar contains second order reasonings

Discovering the axioms built-in into the Mizar checker

MIZAR TYPES

usually defined by formulas, so in fact predicates treated in a special way by Miizar

possibly term-dependent: "Function of NAT, REAL"

widening hierarchy, largest type "set"

Adjectives: "non empty one-to-one Function of NAT, NAT"

Relativization: simplest method of elimination

MIZAR SYNTACTIC LAYERS

Large-scale formalization causes notation conflicts

User friendliness requires simple notation

Solution in Mizar: pattern(format) layer and constructor layer, mapping is done according to Environment Directives of each article, nontrivial

Translation: only the constructor layer is semantically relevant, so the articles are translated after the constructors are resolved

SECOND ORDER PARTS

Hard to translate

Fraenkel (setof) terms: "{n*n where n is Nat; n < 100}"

Schemes:

```
scheme Separation {A()->set, P[set]}:
ex X st for x holds x in X iff x in A() & P[x];
```

BUILT-IN AXIOMS

Requirements: includable pieces of code in the implementation - properties of basic set and arithmetical operations

Numbers(rational): mapped to Pascal implementation, so e.g. fast evaluation of 27*56/9. Hard to get in theorem provers.

Equality reasoning in the Mizar checker

anything else?

WHICH THEOREM PROVERS?

Various input formats: TPTP, Otter, LOP, DFG. Usually Prolog-based

Main proving problems: explosion of type inferences along the widening paths of the type hierarchy:
" x is Nat" -> "x is Integer" -> "x is Real"

SPASS prover: automated approximation of the type theory, semantic blocking of ill typed inferences and inference results

STRUCTURE OF THE TRANSLATED LIBRARY

First order exporter (fo_tool): command line tool based on the Mizar Pascal implementation, creates DFG version of the Mizar library.

External references for all theorem proofs are collected and exported too - "hard problems" - about 30000

Checker problems (Mizar Simple Justifications) are collected and exported - "easy problems" - about 500 000

EXAMPLE OF THE TRANSLATION

```
theorem Th9:
 z<>0 & x*z=y*z implies x=y
proof
  assume  z<>0;
    then consider z' being real number such that
      A1: z * z' = 1 by AXIOMS:20;
    assume x * z = y * z;
      then x * (z * z') = y * z * z' by AXIOMS:16;
      then x = y * 1 by AXIOMS:16,A1;
      hence thesis;
end;
```

```
checker_problem( by_36_24_real_1, [
formula( not( forall([B1],
  implies(
    v1_arytm(B1),     not( equal(k2_arytm(c1,B1),1))))),
t1_r_by_36_24_real_1),
formula( not( equal(c1,0)),
t2_r_by_36_24_real_1),
formula( forall([B1],
implies(   v1_arytm(B1),
  not(     and(
    not( equal(B1,0)),
      forall([B2],
      implies(         v1_arytm(B2),
        not( equal(k2_arytm(B1,B2),1)))))))),
t3_r_by_36_24_real_1),
formula(v1_arytm(c1),
```