Princess

Theorem Proving in First-Order Logic modulo Linear Integer Arithmetic

October 2, 2012

Authors

- Philipp Rümmer (main developer) (Department of Information Technology, Uppsala University)
- Angelo Brillout (Eidgenössische Technische Hochschule Zürich)

Code-Language / License

written in Scala
licensed under GPL v3

Facts

- uses Presburger arithmetic
- can eliminate quantifiers
- handling of (partial and total) functions via an encoding into uninterpreted predicates
- input formates
  - native format
  - SMT-LIB 2
  - FOF/TFF/CNF dialects of TPTP.
- handling of the theory of arrays via user-specified axioms

Native Input Language

```
\exists int grandson_years, son_years, my_years,
            grandson_months, son_weeks, grandson_days; (
  grandson_days = grandson_months * 4 * 7 &
  grandson_months = grandson_years * 12 &
  son_weeks = son_years * 12 * 4 &
  grandson_days = son_weeks &
  grandson_months = my_years &
  grandson_years + son_years + my_years = 140
)

Formula is valid, satisfying assignment for the existential constants is:
true
```

Native Input Language

```
\existentialConstants {
  int my_years;
}

\problem {
  \exists int grandson_years, son_years,
              grandson_months, son_weeks, grandson_days; (
    grandson_days = grandson_months * 4 * 7 &
    grandson_months = grandson_years * 12 &
    son_weeks = son_years * 12 * 4 &
    grandson_days = son_weeks &
    grandson_months = my_years &
    grandson_years + son_years + my_years = 140
  )
}


Formula is valid, satisfying assignment for the existential constants is:
(my_years + -84 = 0)
```

Presburger arithmetic

- $\neg(0 = x + 1)$
- $x + 1 = y + 1 \to x = y$
- $x + 0 = x$
- $(x + y) + 1 = x + (y + 1)$
- $(P(0) \wedge \forall x(P(x) \to P(x + 1))) \to \forall y P(y)$.

Performance

- TPTP Performance (June 2012)
  - TFA: Winner, place 1/3
  - FOF: place 12/16
  - FOF@Turing: place 12/16
- AUFLIA Benchmarks (November 2011)
  benchmarks on containing arrays, uninterpreted functions, integer arithmetic, and quantifiers
  place 2