VALIDATING MATHEMATICAL THEORIES AND ALGORITHMS WITH RISCAL

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria

JOHANNES KEPLER UNIVERSITY LINZ

The RISC Algorithm Language (RISCAL)

A language and software system for investigating finite mathematical models.

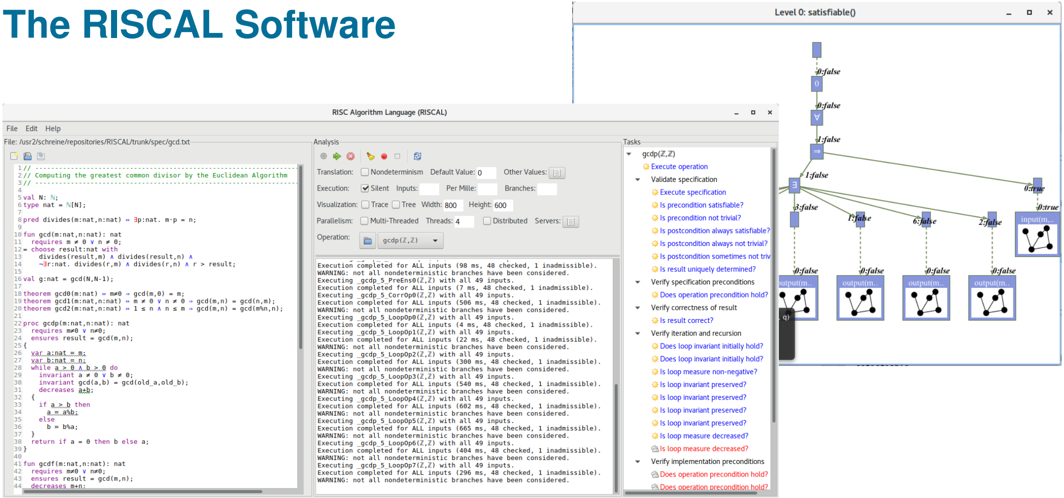
- Formulation of mathematical theories and theorems.
- Formulation and specification of (also non-deterministic) algorithms.
- Rooted in strongly typed first order logic and set theory.
- All types are finite (with sizes determined by model parameters).
- All formulas are automatically decidable by model checking.
- Correctness of all algorithms is automatically decidable by model checking.

Checking in some model of fixed size before proving in models of arbitrary size.

RISCAL Specifications

```
val n: ℕ;
type LiteralBase = ℤ[-n,n];
type Literal = LiteralBase with value ≠ 0;
...
pred satisfiable(f:Formula) ⇔ ∃v:Valuation. satisfies(v,f);
pred valid(f:Formula) ⇔ ∀v:Valuation. satisfies(v,f);
fun not(f: Formula):Formula = { c | c:Clause with ∀d∈f. ∃l∈d. -l∈c };
...
theorem notValid(f:Formula) ⇔ valid(f) ⇔ ¬satisfiable(not(f));
...
multiple pred DPLL(f:Formula)
  ensures result ⇔ satisfiable(f);
  decreases |literals(f)|;
⇔ if f = ∅[Clause] then
    ⊤
  else if ∅[Literal] ∈ f then
    ⊥
  else choose l∈literals(f) in
    DPLL(substitute(f,l)) ∨ DPLL(substitute(f,-l));
```

The RISCAL Software



Automatic checking of theorems, algorithms, and verification conditions; visual explanation of formula values.