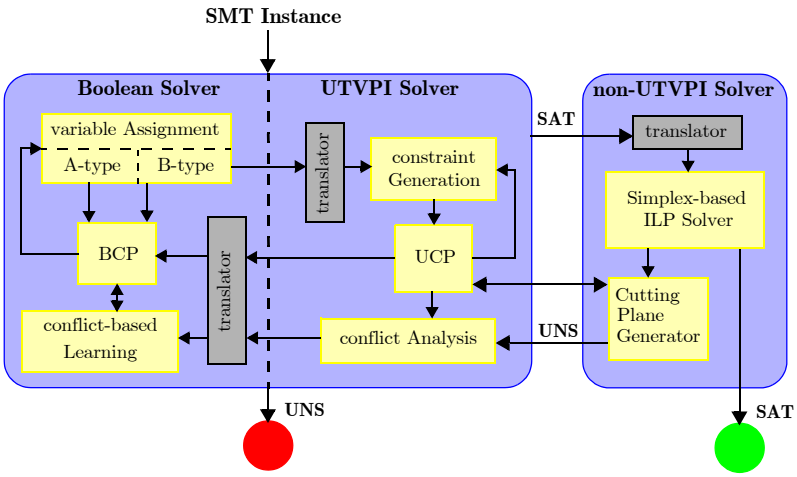
ARIO: A Scalable Satisfiability Modulo Theories (SMT) Solver

Hossein M. Sheini
Advisor: Karem A. Sakallah

Electrical Engineering and Computer Science Department
University of Michigan
1301 Beal Ave., Ann Arbor, MI 48109-2122, USA
hsheini@eecs.umich.edu

ARIO is a SAT-based solver capable to solve satisfiability problems involving Boolean, integer and real variables. The solver replaces integer and linear constraints in the logical representation of the problem with fresh Boolean *indicator* variables. The unique feature of the ARIO is its online-offline combination of strategies for solving different theories.



Main Features

1. **DPLL based SAT Solver -** The core of ARIO is a DPLL-based Chaff-style SAT solver adopting VSIDS and random decision heuristics, two-watched-literal Boolean Constraint Propagation strategy, conflict-induced learning, non-chronological backtracking, clause removal and random restarts.
2. **Online UTVPI Solver -** An incremental transitive-closure based solver for solving Unit-Two-Variable-Per-Inequality constraints. This solver is tightly coupled with the SAT solver, meaning that as soon as the indicator variable associated with a UTVPI constraint is assigned, the set of activated UTVPI constraints in updated. The transitive-closure property of the UTVPI constraints in maintained

throughout the search. Conflicts are communicated with the SAT solver via indicator variables and is further analyzed by the SAT solver.

3. **Online/Offline non-UTVPI Solver -** The feasibility of activated non-UTVPI constraints together with activated UTVPI constraints is checked using an offline Simplex/Branch and Bound based MIP solver. In case of infeasibility, the infeasible core is analyzed using cutting plane theory to further prune the search space by generating fresh UTVPI constraints. The feasibility of non-UTVPI constraints is also checked online while the search proceeds to capture conflicts early.
4. **Ackermann Method for Eliminating Uninterpreted Functions**
5. **Lazy Consistency Checking of Linear Constraints -** By ensuring that the problem is positively unate with regards to indicator variables, ARIO eliminates consistency checking when assigning indicator variables to false. This would limit the calls to the relatively slow UTVPI and non-UTVPI solvers to only cases when the SAT solver implies those variables to true to maintain the satisfiability of the problem.

Implementation

ARIO is implemented in C++ and compiled with gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5). The executable and more information/demo is publicly available at ARIO web site: http://www.eecs.umich.edu/~ario/

Supported Theories

- QF_IDL (Integer Difference Logic)
- QF_RDL (Real Difference Logic)
- QF_LIA (Linear Integer Arithmetic)
- QF_LRA (Linear Real Arithmetic)
- QF_UF (Uninterpreted Functions)
- QF_UFIDL (Integer Difference Logic with Uninterpreted Functions)
- QF_UFLIA (Linear Integer Arithmetic with Uninterpreted Functions)
- QF_UFLRA (Linear Real Arithmetic with Uninterpreted Functions)

Publications

[1] Hossein M Sheini, and Karem A. Sakallah, "**A SAT-based Decision Procedure for Mixed Logical/Integer Linear Problems**", International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'05) - LNCS 3524, pp 320-335, 2005.

[2] Hossein M Sheini, and Karem A. Sakallah, "**A Scalable Method for Solving Satisfiability of Integer Linear Arithmetic Logic**", Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT'05) - LNCS 3569, pp 241-256, 2005.