J. Hoenicke

19.06.2012
submit until 26.06.2012, 14:15

Tutorials for Decision Procedures
Exercise sheet 8

Exercise 1: Deciding T_E
Apply the DAG-based decision procedure to decide satisfiability for the following Σ_E-formulae:

(a) $f(x) = x \ \wedge \ f(a) \neq a$

(b) $f(x) = x \ \wedge \ a = f(f(x)) \ \wedge \ f(a) \neq a$

(c) $f(g(x)) = g(f(x)) \ \wedge \ f(g(f(y))) = x \ \wedge \ f(y) = x \ \wedge \ g(f(x)) \neq x$

(d) $p(x) \ \wedge \ f(f(f(f(x)))) = x \ \wedge \ f(f(f(x))) = f(f(x)) \ \wedge \ \neg p(f(x))$

Exercise 2: Implementing Congruence Closure — Constructing the DAG
Implement the union-find data structure needed for the congruence closure algorithm. There is a stub in the file `CcSolver.java` on the lecture's website. Write some tests to see if it works correctly. Then implement the function `assertTerm(Term t)` in order to construct the DAG that the solver will work on. What is the time complexity of your code for building the DAG depending on the length of the formula?

Exercise 3: Implementing Congruence Closure
Using the union-find data structure of the previous exercise, write a solver for the conjunctive quantifier-free fragment of the theory of equality with uninterpreted functions (no uninterpreted predicates). You can compile and execute the stub on the website in the same manner as on the previous sheet. You need to change the corresponding line in `Main.java`. Give an upper bound for the time complexity of your algorithm depending on the number of symbols in the input formula. (Ideally this would be a polynomial; you receive all points for a less efficient implementation, though.)