A Fast SAT Solver for Isabelle in Standard ML

Armin Heller

July 27, 2015

Abstract

This contribution contains a fast SAT solver for Isabelle written in Standard ML. By loading the theory DPT_SAT_Solver, the SAT solver installs itself (under the name “dptsat”) and certain Isabelle tools like Refute will start using it automatically. This is a port of the DPT (Decision Procedure Toolkit) SAT Solver written in OCaml. Theory DPT_SAT_Tests tests the solver on a few hundred problems.

Contents

theory DPT_SAT_Solver
imports SAT
begin

ML_file "dpt_sat_solver.ML"

end

theory DPT_SAT_Tests
imports DPT_SAT_Solver
begin

ML {*
val path = File.tmp_path (Path.explode "sat.out")
val max_secs = 60

(*
val _ = File.write path ""
fun write_out s = (tracing s; File.append path (s ^ "\n"))
*)
val write_out = tracing

fun test name =
  let
    val name = "cnf/" ^ name

1
val timer1 = Timer.startRealTimer ()
val formula = 
  SAT_Solver.read_dimacs_cnf_file
     (Path.append (Resources.master_directory @{theory}) (Path.explode name))
val timer2 = Timer.startRealTimer ()
val res = SAT_Solver.invoke_solver solver formula
val code = case res of
  SAT_Solver.SATISFIABLE _ => "SAT"
  | SAT_Solver.UNSATISFIABLE _ => "UNSAT"
  | SAT_Solver.UNKNOWN => "UNKNOWN"
fun show_time timer = 
  signed_string_of_int (Time.toMilliseconds (Timer.checkRealTimer timer1)) ^ "ms"
in
  write_out (solver ^ ":" ^ name ^ ": " ^ code ^ " " ^ show_time timer2); code
end
handle TimeLimit.TimeOut => (write_out (solver ^ ":" ^ name ^ ": TIMEOUT")); "UNKNOWN")
in
TimeLimit.timeLimit (Time.fromSeconds max_secs) aux ()
handle TimeLimit.TimeOut => (write_out (solver ^ ":" ^ name ^ ": TIMEOUT")); "UNKNOWN")
end

fun sat name = (test name = "SAT" orelse error "Expected SAT")
fun unsat name = (test name = "UNSAT" orelse error "Expected UNSAT")

ML_val {* unsat "np.core.398356.cnf" *}
ML_val {* sat "np.core.398568.cnf" *}
ML_val {* unsat "np.core.398723.cnf" *}
ML_val {* unsat "np.core.398761.cnf" *}
ML_val {* unsat "np.core.398773.cnf" *}
ML_val {* unsat "np.core.398787.cnf" *}
ML_val {* unsat "np.core.398823.cnf" *}
ML_val {* unsat "np.core.398855.cnf" *}
ML_val {* unsat "np.core.398863.cnf" *}
ML_val {* unsat "np.core.398889.cnf" *}
ML_val {* unsat "np.core.398907.cnf" *}
ML_val {* sat "np.core.399306.cnf" *}
ML_val {* sat "np.core.399317.cnf" *}
ML_val {* sat "np.core.399458.cnf" *}
ML_val {* sat "np.core.399645.cnf" *}
ML_val {* unsat "np.core.399856.cnf" *}
ML_val {* unsat "np.core.399874.cnf" *}
ML_val {* unsat "np.core.399904.cnf" *}
ML_val {* unsat "np.core.399960.cnf" *}
ML_val {* unsat "np.core.400034.cnf" *}
ML_val {* unsat "np.core.400046.cnf" *}
ML_val {* unsat "np.core.400209.cnf" *}
ML_val {* unsat "np.core.400219.cnf" *}
ML_val {* unsat "np.core.400351.cnf" *}
ML_val {* unsat "np.core.400353.cnf" *}
ML_val {* unsat "np.core.400474.cnf" *}

ML_val { * unsat "np.core.400496.cnf" *}
ML_val { * sat "np.core.400660.cnf" *}
ML_val { * sat "np.core.400683.cnf" *}
ML_val { * unsat "np.core.400719.cnf" *}
ML_val { * unsat "np.core.400745.cnf" *}
ML_val { * unsat "np.core.400795.cnf" *}
ML_val { * unsat "np.core.401023.cnf" *}
ML_val { * sat "np.core.401292.cnf" *}
ML_val { * unsat "np.core.401685.cnf" *}
ML_val { * unsat "np.core.401784.cnf" *}
ML_val { * sat "np.core.402032.cnf" *}
ML_val { * unsat "np.core.402136.cnf" *}
ML_val { * unsat "np.core.402512.cnf" *}
ML_val { * sat "np.core.402547.cnf" *}
ML_val { * unsat "np.core.402722.cnf" *}
ML_val { * unsat "np.core.402730.cnf" *}
ML_val { * unsat "np.core.402742.cnf" *}
ML_val { * unsat "np.core.402772.cnf" *}
ML_val { * unsat "np.core.402774.cnf" *}
ML_val { * unsat "np.core.402778.cnf" *}
ML_val { * unsat "np.core.402794.cnf" *}
ML_val { * unsat "np.core.403005.cnf" *}
ML_val { * unsat "np.core.403015.cnf" *}
ML_val { * unsat "np.core.403051.cnf" *}
ML_val { * unsat "np.core.403079.cnf" *}
ML_val { * unsat "np.core.403559.cnf" *}
ML_val { * unsat "np.core.403586.cnf" *}
ML_val { * unsat "np.core.403624.cnf" *}
ML_val { * unsat "np.core.403642.cnf" *}
ML_val { * unsat "np.core.403836.cnf" *}
ML_val { * unsat "np.core.403838.cnf" *}
ML_val { * unsat "np.core.403862.cnf" *}
ML_val { * unsat "np.core.404160.cnf" *}
ML_val { * unsat "np.core.404182.cnf" *}
ML_val { * unsat "np.core.404186.cnf" *}
ML_val { * unsat "np.core.404196.cnf" *}
ML_val { * unsat "np.core.404200.cnf" *}
ML_val { * unsat "np.core.404234.cnf" *}
ML_val { * unsat "np.core.404238.cnf" *}
ML_val { * unsat "np.core.404246.cnf" *}
ML_val { * unsat "np.core.404266.cnf" *}
ML_val { * unsat "np.core.404318.cnf" *}
ML_val { * unsat "np.core.404326.cnf" *}
ML_val { * unsat "np.core.404334.cnf" *}
ML_val { * unsat "np.core.404344.cnf" *}
ML_val { * unsat "np.core.404368.cnf" *}
ML_val { * unsat "np.core.404388.cnf" *}
ML_val { * unsat "np.core.404394.cnf" *}
ML_val { * unsat "np.core.404414.cnf" *}
ML_val { * unsat "np.core.404460.cnf" *}
ML_val { * unsat "np.core.404506.cnf" *}
ML_val { * unsat "np.core.404510.cnf" *}
ML_val { * unsat "np.core.404534.cnf" *}
ML_val { unsat "np.core.404592.cnf" * }
ML_val { unsat "np.core.404596.cnf" * }
ML_val { unsat "np.core.404866.cnf" * }
ML_val { unsat "np.core.404876.cnf" * }
ML_val { unsat "np.core.405031.cnf" * }
ML_val { unsat "np.core.405035.cnf" * }
ML_val { unsat "np.core.405052.cnf" * }
ML_val { unsat "np.core.405056.cnf" * }
ML_val { unsat "np.core.405095.cnf" * }
ML_val { unsat "np.core.405097.cnf" * }
ML_val { unsat "np.core.405100.cnf" * }
ML_val { unsat "np.core.405125.cnf" * }
ML_val { unsat "np.core.405155.cnf" * }
ML_val { unsat "np.core.405184.cnf" * }
ML_val { unsat "np.core.405205.cnf" * }
ML_val { unsat "np.core.405217.cnf" * }
ML_val { unsat "np.core.405254.cnf" * }
ML_val { unsat "np.core.405286.cnf" * }
ML_val { unsat "np.core.405296.cnf" * }
ML_val { unsat "np.core.405314.cnf" * }
ML_val { unsat "np.core.405343.cnf" * }
ML_val { unsat "np.core.405362.cnf" * }
ML_val { unsat "np.core.405372.cnf" * }
ML_val { unsat "np.core.405391.cnf" * }
ML_val { unsat "np.core.405443.cnf" * }
ML_val { unsat "np.core.405445.cnf" * }
ML_val { unsat "np.core.405455.cnf" * }
ML_val { unsat "np.core.405464.cnf" * }
ML_val { unsat "np.core.405536.cnf" * }
ML_val { unsat "np.core.405571.cnf" * }
ML_val { unsat "np.core.405579.cnf" * }
ML_val { unsat "np.core.405588.cnf" * }
ML_val { unsat "np.core.405647.cnf" * }
ML_val { unsat "np.core.405649.cnf" * }
ML_val { unsat "np.core.405657.cnf" * }
ML_val { unsat "np.core.405687.cnf" * }
ML_val { unsat "np.core.405701.cnf" * }
ML_val { unsat "np.core.405703.cnf" * }
ML_val { unsat "np.core.405721.cnf" * }
ML_val { unsat "np.core.405740.cnf" * }
ML_val { unsat "np.core.405811.cnf" * }
ML_val { unsat "np.core.405817.cnf" * }
ML_val { unsat "np.core.405823.cnf" * }
ML_val { unsat "np.core.405869.cnf" * }
ML_val { unsat "np.core.406136.cnf" * }
ML_val { unsat "np.core.406138.cnf" * }
ML_val { sat "np.core.406192.cnf" * }
ML_val { sat "np.core.406216.cnf" * }
ML_val { unsat "np.core.406290.cnf" * }
ML_val { unsat "np.core.406294.cnf" * }
ML_val { unsat "np.core.406310.cnf" * }
ML_val { unsat "np.core.406355.cnf" * }
ML_val { unsat "np.core.406411.cnf" * }
ML_val {* unsat "np.core.406413.cnf" *}
ML_val {* sat "np.core.406457.cnf" *}
ML_val {* sat "np.core.406541.cnf" *}
ML_val {* unsat "np.core.406599.cnf" *}
ML_val {* unsat "np.core.406601.cnf" *}
ML_val {* unsat "np.core.406609.cnf" *}
ML_val {* sat "np.core.406679.cnf" *}
ML_val {* unsat "np.core.406857.cnf" *}
ML_val {* unsat "np.core.406866.cnf" *}
ML_val {* unsat "np.core.406927.cnf" *}
ML_val {* unsat "np.core.406994.cnf" *}
ML_val {* unsat "np.core.407020.cnf" *}
ML_val {* unsat "np.core.407028.cnf" *}
ML_val {* sat "np.core.407044.cnf" *}
ML_val {* sat "np.core.407130.cnf" *}
ML_val {* unsat "np.core.407514.cnf" *}
ML_val {* unsat "np.core.407526.cnf" *}
ML_val {* unsat "np.huff.402973.cnf" *}
ML_val {* unsat "np.huff.403048.cnf" *}
ML_val {* unsat "np.huff.403214.cnf" *}
ML_val {* unsat "np.huff.403497.cnf" *}
ML_val {* unsat "np.huff.405095.cnf" *}
ML_val {* unsat "np.huff.405186.cnf" *}

end