

COMP 360A: Introduction to Automated Reasoning  
Problem Set 6  
Due: Tuesday, Oct 31

Submit, individually, a **pdf** file and the two edited **ml** files **solvers.ml** and **test.ml** to Classroom. OCaml functions must be accompanied by adequate tests to receive full credit. You may resubmit as much as you like before the end of the due date.

**Problem 1.** The goal of this problem is to implement a backtracking SAT solver based on the DPLL algorithm, which outputs a satisfying assignment if there is one.

1. (1 point) Write tests for the following functions: `apply_empty`, `apply`, `assignment_of_literals`, and `choose_var`.
2. (2 points) Finish designing `apply_clause` and `applySing`.
3. (2 points) Finish designing `naive_sat`. Use `choose_var` and `apply` as helper functions.
4. (4 points) Finish designing `find_unit_clauses`, `find_pure`, and `dpll`. Use the first two functions, `choose_var` and `apply` as helper functions for `dpll`.

**Problem 2.** (3 points) Prove the following proposition. Let  $F$  be a CNF formula. If  $F$  is unsatisfiable, then there exists a refutation  $R$  of  $F$  such that for all clauses  $C_i$  in  $R$ ,  $C_i$  is not a tautology. (Hint: Use the resolution restriction and the resolution expansion lemmas from the textbook.)

**Problem 3.** The goal of this problem is to implement an UNSAT solver, which outputs a refutation if there is one. Note that because we are implementing resolution traces with lists, the ordering of the traces is reversed: in particular, resolvents precede their parents.

1. (1 point) Write tests for the following functions: `proper_sublist` and `subset`.
2. (3 points) Finish designing `is_taut` and `resolve_on`. Use `proper_sublist` and `is_taut` as helpers for `resolve_on`.
3. (3 points) Finish designing `resolve`. Use `resolve_on` as a helper.
4. (3 points) Finish designing `resolve_pairwise`. Use `resolve` as a helper.
5. (3 points) Finish designing `naive_ref`. Use `subset` and `resolve_pairwise` as helpers.
6. (3 extra credit points) Improve `naive_ref` so that the output refutations do not contain repeated resolvents, and so that the resolvents in the output do not contain repeated literals.