

# Solutions to Assignment#3

Liana Yepremyan

## 1 Problem 9.1

*In the backtracking algorithm for SAT, suppose that we always choose a subproblem (CNF formula) that has a clause that is as small as possible; and we expand it along a variable that appears in this small clause. Show that this is a polynomial-time algorithm in the special case in which the input formula has only clauses with two literals (that is, it is an instance of 2-SAT).*

*Solution:* Let us first consider an example, say the following formula  $\phi$ .

$$(X \vee Y) \wedge (Z \vee \neg Y)(X)(\neg X \vee \neg Y)(W \vee Z)(\neg Z \vee W)(\neg U \vee \neg W)(\neg Z \vee U)$$

To understand why the proposed algorithm solves 2-SAT in polynomial time, let us see what happens when we choose  $X$  and expand  $\phi$  along  $X$ . Since  $\phi$  is satisfiable if and only if  $X$  is assigned true, we get a residual formula -

$$(Z \vee \neg Y)(\neg Y)(W \vee Z)(\neg Z \vee W)(\neg U \vee \neg W)(\neg Z \vee U)$$

Note that every clause that remains is either a unit clause, or was present in  $\phi$  originally. In general, this is also the case, if all the clauses of the same size, we chose just one variable randomly and then expand on it, and get that every clause that remains is either a unit clause or was present in  $\phi$  originally. When unit clauses are present, we expand on those variables, by the algorithm. In this example, we see that  $Y$  must be assigned false, and expanding we get:

$$(W \vee Z)(\neg Z \vee W)(\neg U \vee \neg W)(\neg Z \vee U)$$

Again, the remaining non-unit clauses were all present in  $\phi$ ; in fact, this must be the case since the algorithm modifies clauses by deleting literals. Crucially, because of this fact, the residual formula is satisfiable if and only if  $\phi$  was. Thus, there is no need to backtrack, and 2-SAT can be solved by a

linear sequence of linear expansions. In the original backtracking algorithm, this corresponds to skipping the second call to expand whenever the first call returns a non-empty result.  $\square$