
Logic in Computer Science, November 2nd, 2022. Time: 1h30min. No books or lecture notes allowed.

-Insert your answers on the dotted lines ... below, and only there.
-Do NOT modify the problems or the @nota lines.
-When finished, upload this file with the same name: exam.txt

```
-Use the text symbols:     &     v    -       ->          |=        A       E
   for                    AND   OR   NOT   IMPLIES   "SATISFIES"  FORALL  EXISTS  etc.,
like in:
   I  |=  p & (q v -r)     (the interpretation I satisfies the formula p & (q v -r)  ).
   You can write subindices using "_". For example write x_i to denote x-sub-i.
```

Problem 1. (3 points).   @n@nota1:

1a) Given two propositional formulas F and G, is it true that F -> G is tautology and
    F satisfiable, then G is satisfiable?
    Prove it using only the formal definitions of propositional logic.

...

1b) Give an example of formulas F1, F2, and F3 such that F1 & F2 & F3 is unsatisfiable
    and any conjuction of two of then is satisfiable.

...

Problem 2. (2.5 points).   @n@nota2:

We define the problem NEG-SAT as follows:
given a propositional formula F, to determine whether there exists I such that I |= -F.

a) Describe a linear-time algorithm for NEG-SAT when the input formula
   is in CNF. Justify its correctness and its cost.

   Hint: you can assume that, given a clause C, detecting if
         C contains contradictory literals, i.e., p and -p for
         some variable p, can be done in linear time.

...

b) Let us call CNF-NEG-SAT the linear-time algorithm of the previous
   exercise for NEG-SAT when the input formula is in CNF:

```
     Algorithm CNF-NEG-SAT
     ---------------------

     Input: propositional formula F in CNF

     Output:
       YES if there exists I such that I |= -F,
       NO otherwise
```

   Consider now the following algorithm for solving the SAT problem for
   arbitrary formulas:

```
     Algorithm MY-SAT
     ----------------

     Input: propositional formula F
```

```
     Output:
       YES if there exists I such that I |= F,
       NO otherwise

     Step 1. G := Tseitin_transformation_of(-F)
     Step 2. return CNF-NEG-SAT(G)
```

The algorithm MY-SAT is NOT correct. Prove it giving a counterexample.

...

Problem 3. (2.5 points). @n@nota3:

3) Given S a set of clauses (CNF) over n propositional symbols,
and Resolution the deductive rule:

```
                 p v C          -p v D
                 ---------------------      for some symbol p
                         C v D
```

3a) Given n propositional symbols, how many different clauses are there (seen as sets of literals)?

...

3c) Is Resolution a correct deductive rule: (p v C) & (-p v D) |= C v D for any p, C, D? Prove it.

...

3c) Can Resolution be used to decide SAT? Briefly explain why o why not?

...

Problem 4. (2 points). @n@nota4:

4) Consider the cardinality constraint x1 + x2 + x3 + x4 + x5 >= 2 (expressing that at least 2 of the propositional symbols {x1, x2, x3, x4, x5} are true).

4a) Write the clauses needed to encode this constraint using no auxiliary variables.

...

4b) In general, in terms of n and k, how many clauses are needed to encode a cardinality constraint x1 + . . . + xn >= k using no auxiliary variables? (give no explanations here).

...

4c) Write at least two names of any other encoding you know for cardinality constraints, encodings that do use auxiliary variables.

...