Lógica en la Informática / Logic in Computer Science

Monday June 13, 2016

Time: 2h30min. No books, lecture notes or formula sheets allowed.

Note on evaluation:
eval(propositional logic) = max{ eval(Problems 1,2,3), eval(partial exam) }.
eval(first-order logic) = eval(Problems 4,5,6).

1a) Let F and G be propositional formulas such that F is a tautology. Is it true that $F \wedge G \equiv G$? Prove it using only the definitions of propositional logic.

1b) Let F and G be propositional formulas such that F is satisfiable and $F \rightarrow G$ is also satisfiable. Is it true that G is satisfiable? Prove it using only the definitions of propositional logic.

2) Let us remember the well-known graph coloring problem. **Input:** a natural number k, and an (undirected) graph with n vertices and m edges of the form $(u_1, v_1) \ldots (u_m, v_m)$, with all u_i and v_i in $\{1 \ldots n\}$, and **Question:** is there a way to "color" each vertex with a color (a number) in $1 \ldots k$ such that adyacent vertices get different colors?

We know that graph coloring is NP-complete in general. But what is its complexity if $k = 2$? Explain why using sat-based arguments.

3) Let S be a satisfiable set of propositional Horn clauses.
3a) What is the complexity of finding the *minimal* model of S, that is, the model I with the minimal number of symbols p such that $I(p) = 1$?
3b) What is the complexity of deciding whether S has only one model or more than one?
For both questions, explain very, very, briefly why.

4) We want to write a computer program that takes as input two arbitrary first-order formulas F and G and always terminates writing "yes" if $F \equiv G$, and "no" otherwise. Explain very shortly the steps you would follow to do this, or to get something as similar as possible.

5) Formalize and prove by resolution that sentence E is a logical consequence of the other four.
A: If a person likes logic, he does not like football.
B: Brothers of football players like football.
C: Messi is a football player and Ney is his brother.
D: Ney likes logic.
E: Our teacher is a nice guy who knows a lot about football and logic.

6) Complete the following graph coloring program (see problem 2). Do `makeConstraints` recursively, using `#\=` and the built-in predicate `nth1(I,L,X)` ("the `I`th element of the list `L` is `X`").

```
:- use_module(library(clpfd)).
numVertices(5).
edges([ 1-2, 1-3, 2-3, 2-4, 2-5, 3-5 ]).
numColors(3).

main:- numVertices(N),edges(Edges), listOfNPrologVars(N,Vars), ...
       Vars ins ...
       makeConstraints(Edges,Vars),
       ...
       write(Vars), nl.

makeConstraints(...

listOfNPrologVars(...
```