Lógica en la Informática / Logic in Computer Science

Thursday January 12th, 2017

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

Note on evaluation:
eval(propositional logic) = max{ eval(Problems 1,2,3,4), eval(partial exam) }.
eval(first-order logic) = eval(Problems 5,6,7).

1) Let F and G be arbitrary propositional formulas.

1a) Can it happen that $F \models G$ and $F \models \neg G$? Prove it from the definitions of propositional logic.

1b) Assume $F \models G$. Prove, from the definitions of propositional logic, that then $F \equiv F \wedge G$.
Hint: prove $F \equiv F \wedge G$ distinguishing the two cases: $I \models F$ and $I \not\models F$ (and use $F \models G$ in one case).

2) The Tseitin transformation T transforms an arbitrary propositional formula F into a CNF (a set of clauses with auxiliary variables) $T(F)$ that is *equisatisfiable*: F is SAT iff $T(F)$ is SAT. Moreover, the size of $T(F)$ is linear in the size of F. Answer **very briefly:** Is there any known transformation T' into an equisatisfiable linear-size DNF? If yes, which one? If not, why?

3) Answer **very briefly:** What is 2-SAT? Is it polynomial? Why?

4) Answer **very briefly:** Which clauses are needed to encode the pseudo-Boolean constraint $2x + 3y + 5z + 6u + 8v \leq 11$ into SAT, if no auxiliary variables are used? Which clauses are needed in general, with no auxiliary variables, for a constraint $a_1x_1 + \ldots + a_nx_n \leq k$?

5) Is the following first-order formula satisfiable? If not, explain why. If yes, give a model (and no explanations).

$\forall x \;\; \neg p(x, x) \quad \wedge$
$\forall x\, \forall y\, \forall z \;\; (\;\; p(x, y) \wedge p(y, z) \rightarrow p(x, z) \;\;) \quad \wedge$
$\forall x\, \exists y \;\; p(x, y) \quad \wedge$
$\forall x\, \forall y \;\; (\;\; p(x, y) \rightarrow \exists z\, (p(x, z) \wedge p(z, y)) \;\;)$

6)
6a) Explain in a few words how you would formally prove, given two first-order formulas F and G, that $F \not\models G$.
6b) Same question for $F \models G$.
6c) F is $\forall x\; p(a, x) \wedge \exists y\; \neg q(y)$ and G is $\exists v \exists w\; (\neg q(w) \wedge p(v, a))$. Do we have $F \models G$? Prove it.
6d) F is $\forall x \exists y\; p(x, y)$ and G is $\exists y \forall x\; p(x, y)$. Do we have $F \models G$? Prove it.

7) Write a program `return(L,A)` in swi prolog (using `library(clpfd)` or not, feel free) that outputs (writes) the *minimal* number of coins needed for returning the amount `A` if (infinitely many) coins of each value of the list `L` are available. Two examples:

`?-return([1,5,6],10).` writes 2 (since 2*5 = 10).
`?-return([1,2,5,13,17,35,157],361).` writes 5 (since 1*13 + 2*17 + 2*157 = 361).