Lógica en la Informática / Logic in Computer Science
June 22nd, 2018. Time: 2h30min. No books or lecture notes.

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 be a formula. Is it true that F is satisfiable if, and only if, all logical consequences of F are satisfiable formulas? Prove it using only the definitions of propositional logic.

Answer: Yes, it is true.
Implication $\Longrightarrow$: Let G be any logical consequence of F. Then we have: F is satisfiable
implies for some I, $I \models F$ [definition of satisfiable]
implies $I \models G$ [definition of logical consequence, since $F \models G$]
implies G satisfiable [definition of satisfiable]
Implication $\Longleftarrow$: All logical consequences of F are satisfiable formulas
implies F satisfiable [since, by definition of logical consequence, $F \models F$]

1b) Is it true that a formula F is a tautology if, and only if, its Tseitin transformation $Tseitin(F)$ is a tautology? Prove it using only the definitions of propositional logic. Important note: all your answers should be as short, clean and simple as possible.

`Answer:` No. Counterexample: let F be the tautology $p \vee \neg p$. $Tseitin(F)$ is a set of four clauses: $\{\ a,\ \ \neg a \vee p \vee \neg p,\ \ a \vee p,\ \ a \vee \neg p\ \}$. $Tseitin(F)$ is not a tautology: $I \not\models Tseitin(F)$ if $I(a) = 0$.

[`Comment:` Every model of F can be extended to a model of $Tseitin(F)$ by interpreting *adequately* the new auxiliary variables in $Tseitin(F)$, and, conversely, every model of $Tseitin(F)$ can be converted into a model of F by "forgetting" about the auxiliary symbols (therefore both are *equisatisfiable*: if one of them is satisfiable, the other one also is).]

2a) Notation: we consider clauses C and sets S of clauses over a set of propositional symbols $\mathcal{P}$. We define $negateAll(C) = \{negate(lit) \mid lit \in C\}$, that is, the clause obtained by flipping (changing the sign) of *all* literals. For example, $negateAll(p \vee \neg q \vee \neg r)$ is $\neg p \vee q \vee r$. Similarly, we define $negateAll(S) = \{negateAll(C) \mid C \in S\}$, i.e, all literals in S are flipped. Explain in two lines: Is it true that S is satisfiable iff $negateAll(S)$ is satisfiable?

`Answer:` Yes. If I is an interpretation, define I' such that $I(p) \neq I'(p)$ for every $p \in \mathcal{P}$. Then I is a model of one of S or $negateAll(S)$ iff I' is a model of the other one.

2b) Now, for $N \subseteq \mathcal{P}$, $negate(N, C)$ negates the literals whose symbol is in N. For example, $negate(\ \{p, q\},\ p \vee \neg q \vee \neg r\ )$ is $\neg p \vee q \vee \neg r$. We extend this to $negate(N, S)$ as before. Explain in two lines: Is it true that S is satisfiable iff $negate(N, S)$ is satisfiable?

`Answer:` Yes. Similarly to 2a), define I' such that $I(p) \neq I'(p)$ iff $p \in N$. Then, I is a model of one of S or $negate(N, S)$ iff I' is a model of the other one.

2c) S is called *renamable Horn* if there is some $N \subseteq \mathcal{P}$ such that $negate(N, S)$ is Horn. Explain in two lines: Given S and N such that $negate(N, S)$ is Horn, what would you do to efficiently decide whether S is satisfiable?

`Answer:` S is satisfiable iff $negate(N, S)$ is satisfiable, and, for deciding the latter, we can use the linear-time HornSat algorithm.

2d) Assume you are given a renamable Horn S but you do not know the set N. Explain in two lines: Can you still decide the satisfiability of S with the same cost as in 2c)? We mean the same asymptotical cost, in $O(...)$-notation.

`Answer:` The linear-time HornSat algorithm works by positive unit propagation. Since we do not know N, we do not know which unit literals to propagate, but it works to do normal (positive and negative) unit propagation, which is still linear.

3) Write the clauses obtained by encoding $AtMostOne(x_0, x_1, x_2, x_3)$ using the logarithmic encoding (only write the clauses, give no explanations).

Answer:

$\neg x_0 \vee \neg a_0 \qquad \neg x_1 \vee a_0 \qquad \neg x_2 \vee \neg a_0 \qquad \neg x_3 \vee a_0$

$\neg x_0 \vee \neg a_1 \qquad \neg x_1 \vee \neg a_1 \qquad \neg x_2 \vee a_1 \qquad \neg x_3 \vee a_1.$

4a) Assume we have a binary predicate symbol P and two interpretations I_1 and I_2, where D_{I_1} is the natural numbers, D_{I_2} is the integers, and $P_{I_1}(n,m) = P_{I_2}(n,m) = \ n > m$. Write a formula F, using no other predicate symbols than P, such that exactly one of the two interpretations is a model of F and say which one. Give no explanations.

Answer: If F is $\forall x \exists y P(x,y)$, then $I_1 \not\models F$ and $I_2 \models F$.

4b) Same question if D_{I_1} is the integers, D_{I_2} is the rational numbers.

Answer: If F is $\forall x \forall y\ p(x,y) \to (\exists z\ P(x,z) \land P(z,y))$, then $I_1 \not\models F$ and $I_2 \models F$.

4c) Same question if D_{I_1} is the real numbers, D_{I_2} the complex numbers, with two binary symbols: a predicate symbol Eq interpreted as equality, and a function symbol p interpreted as the product.

Answer: If F is $\forall x \exists y\ Eq(x, p(y,y))$, then $I_1 \not\models F$ and $I_2 \models F$. (every number has a square root).

5) Assume we have a yes/no question Q, based on some input data. Explain **in a few words** each one of the following cases:

5a) What does it mean that Q is decidable?

Answer: That there exists a procedure that, given the input data, answers correctly whether the answer of Q is yes or no, and always terminates.

5b) What does it mean that Q is semi-decidable?

Answer: That there exists a procedure such that, given the input data, if it terminates it answers correctly whether the answer of Q is yes or no, and it always terminates if the answer is yes.

5c) What does it mean that Q is co-semi-decidable?

Answer: That there exists a procedure such that, given the input data, if it terminates it answers correctly whether the answer of Q is yes or no, and it always terminates if the answer is no.

5d) Is SAT in first-order logic decidable? semi-decidable? co-semi-decidable?

Answer: co-semi-decidable

5e) Same question for logical equivalence.

Answer: semi-decidable

5f) Give an (as simple as you can!) example of non-termination of resolution in first-order logic.

Answer: Two clauses: $\neg p(x) \lor p(f(x))$ and $p(a)$. This generates new clauses: $p(f(a))$, $p(f(f(a)))$, $p(f(f(f(a))))$, ...

6) Formalize and prove by resolution that sentence E is a logical consequence of the other four.

A: Cristiano is a real madrid player
B: Messi and Cristiano are world-class football players
C: To be a world-class football player, one has to be modest
D: Real madrid has no modest players
E: This year Germany will win the world cup

`Answer:` We prove that $A \land B \land C \land D$ is unsatisfiable and therefore $A \land B \land C \land D \models E$. Formalizing with unary predicates RMP, $WCFP$, $Modest$ and the constants $messi$ and $cristiano$, and expressing the sentences in clausal form, we get the clauses $A, B1, B2, C, D$:

$A)$ $RMP(cristiano)$
$B)$ $WCFP(messi) \land WCFP(cristiano)$
$B1)$ $WCFP(messi)$
$B2)$ $WCFP(cristiano)$
$C)$ $\forall x\ WCFP(x) \to Modest(x)$
$\neg WCFP(x) \lor Modest(x)$
$D)$ $\forall x\ RMP(x) \to \neg Modest(x)$
$\neg RMP(x) \lor \neg Modest(x)$

By resolution we obtain the empty clause as follows:

$num:$	$by:$	$mgu:$	$get:$
1)	$res(A,D)$	$x = cristiano$	$\neg Modest(cristiano)$
2)	$res(B2,C)$	$x = cristiano$	$Modest(cristiano)$
3)	$res(1,2)$		$emptyclause$