Lógica en la Informática / Logic in Computer Science
January 14th, 2021. 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).

1) (4 points) Prove your answers to the following questions, using only the formal definitions of propositional logic.

1a) Given two propositional formulas F and G, is it true that $F \to G$ is a tautology iff $F \models G$?

Answer: This is true.

$F \to G$ is a tautology iff	(by def. of $\to$)
$\neg F \vee G$ is a tautology iff	(by def. of tautology)
for all I, $I \models \neg F \vee G$ iff	(by def. of $\models$)
for all I, $eval_I(\neg F \vee G) = 1$ iff	(by def of $eval_I(\vee)$)
for all I, $max(eval_I(\neg F), eval_I(G)) = 1$ iff	(by def of $eval_I(\neg)$)
for all I, $max(1 - eval_I(F), eval_I(G)) = 1$ iff	(by def of max)
for all I, $1 - eval_I(F) = 1$ or $eval_I(G) = 1$ iff	(by def of $-$)
for all I, $eval_I(F) = 0$ or $eval_I(G) = 1$ iff	(by def of $\models$)
for all I, $I \not\models F$ or $I \models G$ iff	(by def of logical consequence)
$F \models G$.	

1b) Let F and G be propositional formulas. Is it true that if $F \to G$ is satisfiable and F is satisfiable, then G is satisfiable?

Answer: This is false. Counterexample: $F = p$ and $G = p \wedge \neg p$. Then $F \to G$ is satisfiable (it has the model I where $I(p) = 0$) and F is satisfiable (model: $I(p) = 1$), but G is unsatisfiable.

2) (3 points) 2-SAT is the satisfiability problem for sets of clauses where each clause has at most 2 literals. Similarly 3-SAT is defined for at most 3 literals.

2a) Explain very briefly what the precise computational complexity of 2-SAT is, and why.

Answer: Linear. Build the directed graph G_S whose nodes are the literals and with two edges $\neg l \to l'$, and $\neg l' \to l$ per clause $l \vee l'$. Then S is unsatisfiable iff G_S has a cycle containing p and $\neg p$ for some $p \in \mathcal{P}$ (linear using the strongly connected components algorithm).

2b) Same question for 3-SAT. In particular, explain why 3-SAT is at least as hard as SAT for arbitrary formulas.

Answer: NP-complete. It is NP-hard because 3-SAT is at least as hard as SAT (for arbitrary formulas), since using Tseitin we can do SAT with 3-SAT. It is in NP because we can check a solution (a model) of 3-SAT in polynomial (in fact, linear) time.

3) (3 points) Let S be a satisfiable set of propositional Horn clauses. Answer the following two questions, explaining very, very, briefly why.

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?

Answer: **3a)** Use the linear-time Horn-SAT algorithm. If it finds a model, it is minimal, because each propagated positive unit *must* be true in *all* models of S.
3b) Any other model must *extend* the unique minimal model I with at least one more true symbol q with $I(q) = 0$. We can try each q, adding it to S as a new unit clause and solve the resulting Horn-SAT problem. This is quadratic, since we try at most $|\mathcal{P}|$ (linear) Horn-SAT problems.

4) (3 points) For 4a and 4b, just write the simplest and cleanest possible formula F. Use no more predicate or function symbols than just p. Give no explanations.

4a) Write a satisfiable first-order formula F, using only a *binary* predicate p, such that all models I of F have an infinite domain D_I.

4b) Write a satisfiable formula F of first-order logic with equality, using only a *unary* predicate p, such that F expresses that there is a single element satisfying p, that is, all models I of F have a single (unique) element e in its domain D_I such that $p_I(e) = 1$.

Answer:

4a: $\forall x\ \neg p(x,x) \quad \wedge \quad \forall x\ \exists y\ p(x,y) \quad \wedge \quad \forall x\ \forall y\ \forall z\ (\neg p(x,y) \vee \neg p(y,z) \vee p(x,z))$

4b: $\exists x\ (p(x) \wedge \forall y\ (\neg x = y \rightarrow \neg p(y)))$

5) (3 points) Let F be the first-order formula $\exists x \forall y \exists z\ (\ p(z,y) \wedge \neg p(x,y)\ )$.
5a) Give a model I of F with $D_I = \{a,b,c\}$.

Answer: Intuitively, we can build the model considering. e.g., that the x that exists is a. Then we need that $\neg p(x,y)$ for all y, that is, $p_I(a,a) = 0$, $p_I(a,b) = 0$, $p_I(a,c) = 0$. Furthermore, we need that for all y, there exists a z such that $p(z,y)$, which we can achieve by taking always the same z (this is not necessary, but here it works): $p_I(b,a) = 1$, $p_I(b,b) = 1$, $p_I(b,c) = 1$. This gives us a model independently of how we define the remaining three cases $p_I(c,a)$, $p_I(c,b)$, $p_I(c,c)$.

5b) Is it true that $F \models \forall x\ p(x,x)$?

Answer: No. The model of F given in 6a does not satisfy $\forall x\ p(x,x)$.

5c) Is there any model of F with a single-element domain?

Answer: No. Calling that single element a, i.e., $D_i = \{a\}$, we would need $p_I(a,a) = 1$ due to the subformula $p(z,y)$, but also $p_I(a,a) = 0$ due to the subformula $\neg p(x,y)$.

6) (4 points) Formalize and prove by resolution that sentence F is a logical consequence of the first five:

A: All people that have electric cars are ecologists.
B: If someone has a grandmother, then that someone has a mother whose mother is that grandmother.
C: A person is an ecologist if his/her mother is an ecologist.
D: Mary is John's grandmother.
E: Mary has an electric car.
F: John is an ecologist.

Answer: We use the following four predicat symbols:

$hasEcar(x)$	means	"x has an electric car"
$isEcologist(x)$	means	"x is an ecologist"
$mother(x, y)$	means	"y is the mother of x"
$grandma(x, y)$	means	"y is the grandmother of x"

We now formalize and prove that $A \wedge \ldots \wedge E \wedge \neg F$ is unsatisfiable:

A: $\forall x\ (hasEcar(x) \rightarrow isEcologist(x)\ )$
B: $\forall x \forall z\ (\ grandma(x, z) \rightarrow \exists y(mother(x, y) \wedge mother(y, z))\ )$
C: $\forall x \forall (y)\ (mother(x, y) \wedge isEcologist(y) \rightarrow isEcologist(x)\ )$
D: $grandma(john, mary)$
E: $hasEcar(mary)$
$\neg$F: $\neg isEcologist(john)$.

The following clauses are obtained:

A: $\neg hasEcar(x) \vee isEcologist(x)$
B gives:
$\forall x \forall z\ (\neg grandma(x, z) \vee \exists y(mother(x, y) \wedge mother(y, z))\ )$
$\forall x \forall z\ (\neg grandma(x, z) \vee (mother(x, f_y(x, z)) \wedge mother(f_y(x, z), z))\ )$
which gives two clauses:
B1: $\neg grandma(x, z) \vee mother(x, f_y(x, z))$
B2: $\neg grandma(x, z) \vee mother(f_y(x, z), z)$
C: $\neg mother(x, y) \vee \neg isEcologist(y) \vee isEcologist(x)$
D: $grandma(john, mary)$
E: $hasEcar(mary)$
$\neg$F: $\neg isEcologist(john)$

Doing resolution steps:

$num:$	$from:$	$mgu:$	$new\ clause:$
1.	$A + E$	$\{x = mary\}$	$isEcologist(mary)$
2.	$B1 + D$	$\{x = john, z = mary\}$	$mother(john, f_y(john, mary))$
3.	$B2 + D$	$\{x = john, z = mary\}$	$mother(f_y(john, mary), mary)$
4.	$2 + C$	$\{x = john, y = f_y(john, mary)\}$	$\neg isEcologist(f_y(john, mary)) \vee isEcologist(john)$
5.	$4 + \neg F$	$\{\}$	$\neg isEcologist(f_y(john, mary))$
6.	$3 + C$	$\{x = f_y(john, mary), y = mary\}$	$\neg isEcologist(mary) \vee isEcologist(f_y(john, mary))$
7.	$5 + 6$	$\{\}$	$\neg isEcologist(mary)$
8.	$1 + 7$	$\{\}$	$[]$