Lógica en la Informática / Logic in Computer Science
January 17th, 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 and G be propositional formulas. Is it true that always $F \models G$ or $F \models \neg G$? Prove it using only the definitions of propositional logic.
Answer: This is false. Let F be the formula p and let G be the formula q. Then for the interpretation I_1 with $I_1(p) = 1$ and $I_1(q) = 0$, we have $I_1 \models F$ and $I_1 \not\models G$, and hence $F \not\models G$. And for the interpretation I_2 with $I_2(p) = 1$ and $I_2(q) = 1$, we have $I_2 \models F$ and $I_2 \not\models \neg G$, and hence also $F \not\models \neg G$.

1b) Let F and G be propositional formulas. Is it true that $F \models G$ iff $F \wedge \neg G$ is unsatisfiable? Prove it using only the definitions of propositional logic.

Answer: $F \models G$

iff	for all I, $I \not\models F$ or $I \models G$	[definition of logical consequence]
iff	for all I, $eval_I(F) = 0$ or $eval_I(G) = 1$	[definition of satisfaction]
iff	for all I, $eval_I(F) = 0$ or $1 - eval_I(G) = 0$	[property of $-$]
iff	for all I, $eval_I(F) = 0$ or $eval_I(\neg G) = 0$	[definition of eval]
iff	for all I, $min(eval_I(F), eval_I(\neg G)) = 0$	[definition of min]
iff	for all I, $eval_I(F \wedge \neg G) = 0$	[definition of eval]
iff	for all I, $I \not\models F \wedge \neg G$	[definition of satisfaction]
iff	$F \wedge \neg G$ is unsatisfiable	[definition of satisfiable]

2) We are interested in the optimization problem, called *minOnes*: given a set S of clauses over variables $\{x_1, \ldots, x_n\}$, finding a *minimal* model I (if it exists), that is, a model of S with the minimal possible number of ones $I(x_1)+\ldots+I(x_n)$. Explain very briefly your answers to the following questions:
2a) Does every S have a unique minimal model, or can there be several minimal models?
2b) Given the set S and an arbitrary natural number k, what is the complexity of deciding whether S has any model I with at most k ones, that is, such that $I(x_1) + \ldots + I(x_n) \leq k$?
2c) Same question as 2a, if S is a set of Horn Clauses.
2d) Same question as 2b, if S is a set of Horn Clauses.

Answer:
2A: There can be several minimal models. If S is $\{p \vee q\}$, then I_1 where $I_1(p) = 1$ and $I_1(q) = 0$ is minimal, and I_2 where $I_2(p) = 0$ and $I_2(q) = 1$ is also minimal.
2B: NP-complete: a) any SAT problem over n variables can be expressed as a *minOnes* problem setting $k = n$; and, reversely, b) any *minOnes* problem can be expressed as a SAT problem adding a (polynomial-size) encoding of the cardinality constraint $x_1 + \ldots + x_n \leq k$.
2C: Every satisfiable set of Horn clauses has a unique minimal model: the one found by the standard polynomial algorithm for Horn SAT doing unit propagation of the positive true literals, where we only set to true those variables that must be true, in *any* model.
2D: Polynomial (in fact linear): just do as in 2C and check whether the resulting model (if any) has less than k ones or not.

3) We want to encode pseudo-Boolean constraints into SAT with the minimal set of clauses, and using no auxiliary variables. For $2x + 3y + 5z + 6u + 8v \leq 11$, the clauses are:
$\neg v \vee \neg u$ $\neg v \vee \neg z$ $\neg v \vee \neg x \vee \neg y$ $\neg u \vee \neg z \vee \neg y$ $\neg u \vee \neg z \vee \neg x$
Write the minimal set of clauses for $2x + 3y + 5z + 6u + 8v \geq 11$ (give no explanations).

Answer: $u \vee v$ $y \vee z \vee v$ $y \vee z \vee u$ $x \vee z \vee v$

4) For each one of the following cases, write a formula F of first-order logic without equality such that F fulfils the requirement. Keep F *as simple as you can* and give no explanations.
4a) F is unsatisfiable.
4b) F is a tautology.
4c) F is satisfiable and has no model I with $|D_I| < 3$.
4d) F is satisfiable but has no model with finite domain.
4e) F is satisfiable and all models I of F have $|D_I| = 2$.
4f) Same question as 4e, but for first-order logic with equality.

Answer:
4a: $p \wedge \neg p$
4b: $p \vee \neg p$
4c: $(\forall x\ p(x,x))\ \wedge\ \exists u\ \exists v\ \exists w\ (\neg p(u,v) \wedge \neg p(u,w) \wedge \neg p(v,w))$
4d: $(\forall x\ \neg p(x,x))\quad \wedge\quad (\forall x\ \forall y\ \forall z\ \ p(x,y) \wedge p(y,z) \rightarrow p(x,z))\quad \wedge\quad (\forall x\ \exists y\ p(x,y))$
4e: not expressible in first-order logic without equality
4f: $\exists x\ \exists y\ (\ \neg(x{=}y)\ \wedge (\forall z\ \ z{=}x \vee z{=}y)\ ).$ Alternative answer: $\neg(a{=}b)\ \wedge (\forall z\ \ z{=}a \vee z{=}b).$

5a) Explain in a few words how to formally prove $F \not\models G$ for given first-order formulas F and G.
5b) Same question for $F \models G$.
5c) 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.
5d) 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.

Answer:

5a: Giving a counter example, an interpretation I such that $I \models F$ but $I \not\models G$.

5b: By proving that $F \wedge \neg G$ is unsatisfiable, turning it into clausal form S, and obtaining the empty clause from S by resolution and factoring.

5c: Yes. $F \models G$. We prove it as in 5b. Here F gives two clauses: $1:\ p(a,x)$ and $2:\ \neg q(c_y)$ (here c_y is the Skolem constant introduced for y).
$\neg G$ is $\neg(\exists v \exists w\ \neg q(w) \wedge p(v,a))$ which becomes $\forall v \forall w\ \neg(\neg q(w) \wedge p(v,a))$ which becomes
$\forall v \forall w\ q(w) \vee \neg p(v,a)$ which becomes the clause $3:\ q(w) \vee \neg p(v,a)$.
By one step of resolution between 1 and 3 with mgu $\{x = a, v = a\}$ we get clause $4 : q(w)$, and with one more step of resolution between 2 and 4 with mgu $\{w = c_y\}$ we get the empty clause.

5d: No. $F \not\models G$. We prove it as in 5a. Consider the interpretation I where $D_I = \{a, b\}$ and p_I is interpreted as equality on this domain. Then $I \models F$ but $I \not\models G$.

6) Consider the following Prolog program and its well-known behaviour:

```
animals([dog,lion,elephant]).
bigger(lion,cat).
faster(lion,cat).
better(X,Y):- animals(L), member(X,L), bigger(X,Y), faster(X,Y).

      ?- better(U,V).
      U = lion
      V = cat
```

In Prolog, a list like `[dog,lion,elephant]` is in fact represented as a term
`f(dog,f(lion,f(elephant,emptylist)))`.
Therefore, we assume that the program also contains the standard clauses for `member` like this:

```
member( E, f(E,_) ).
member( E, f(_,L) ):- member(E,L).
```

Express the program as a set of first-order clauses P and prove that $\exists u\, \exists v\; better(u,v)$ is a logical consequence of P. Which values did the variables u and v get (by unification) in your proof? **Only write the steps and values. No explanations.**

Answer: We have to prove that $P \wedge \neg(\exists u\, \exists v\; better(u,v))$ is unsatisfiable.
Note that `better(X,Y):- animals(L), member(X,L), bigger(X,Y), faster(X,Y)` is the formula
$\forall X\; \forall Y\; better(X,Y) \leftarrow \exists L\; (\; animals(L) \wedge member(X,L) \wedge bigger(X,Y) \wedge faster(X,Y)\;)$
which is
$\forall X\; \forall Y\; better(X,Y) \vee\; \neg\; \exists L\; (\; animals(L) \wedge member(X,L) \wedge bigger(X,Y) \wedge faster(X,Y)\;)$
which is
$\forall X\; \forall Y\; better(X,Y) \vee\; \forall L\; \neg(\; animals(L) \wedge member(X,L) \wedge bigger(X,Y) \wedge faster(X,Y)\;)$
which is
$\forall X\; \forall Y\; better(X,Y) \vee \forall L\; (\; \neg animals(L) \vee \neg member(X,L) \vee \neg bigger(X,Y) \vee \neg faster(X,Y)\;)$
which is
$\forall X\; \forall Y\; \forall L \quad better(X,Y) \;\vee\; \neg animals(L) \;\vee\; \neg member(X,L) \;\vee\; \neg bigger(X,Y) \;\vee\; \neg faster(X,Y).$

Furthermore, the negation of $\exists u\, \exists v\; better(u,v)$ is: $\forall u\, \forall v \quad \neg better(u,v)$.

Altogether, from the program (with, as said, the `member` clauses) and $\forall u\, \forall v \quad \neg better(u,v)$ we get the following 8 clauses:

1. $animals(f(dog, f(lion, f(elephant, emptylist))))$
2. $bigger(lion, cat)$
3. $faster(lion, cat)$
5. $member(E, f(E, L))$
6. $member(E, f(X, L)) \vee \neg member(E, L)$
7. $better(X,Y) \vee \neg animals(L) \vee \neg member(X,L) \vee \neg bigger(X,Y) \vee \neg faster(X,Y).$
8. $\neg better(u,v)$

We now prove by resolution that this set of 8 clauses is unsatisfiable. We get:

9. $\neg animals(L) \vee \neg member(X,L) \vee \neg bigger(X,Y) \vee \neg faster(X,Y)$
   by resolution between 7 and 8, $\sigma = \{u = X, v = Y\}$
10. $\neg animals(L) \vee \neg member(lion, L) \vee \neg faster(lion, cat)$
    by resolution between 9 and 2, $\sigma = \{X = lion, Y = cat\}$
11. $\neg animals(L) \vee \neg member(lion, L)$
    by resolution between 10 and 3, $\sigma = \{\ \}$
12. $\neg member(lion, f(dog, f(lion, f(elephant, emptylist))))$
    by resolution between 11 and 1, $\sigma = \{L = f(dog, f(lion, f(elephant, emptylist)))\}$
13. $\neg member(lion, f(lion, f(elephant, emptylist)))$
    by resolution between 12 and 6, $\sigma = \{E = lion, X = dog, L = f(lion, f(elephant, emptylist))\}$
14. the empty clause
    by resolution between 13 and 5, $\sigma = \{E = lion, L = f(elephant, emptylist))\}$

Here u and v finally got the values $u = X = lion$ and $v = Y = cat$.