Lógica en la Informática / Logic in Computer Science
June 17th, 2019. 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) Let F and G be arbitrary propositional formulas. Prove your answers using only the definitions of propositional logic.

A) Is it true that if $F \models G$ and $F \models \neg G$ then F is unsatisfiable?

B) Is it true that if F is unsatisfiable then $(G \vee F) \to G$ is a tautology?

Answer:

A). yes. By contradiction. Assume F satisfiable. Then, by definition of satisfiable,
there exists an I such that $I \models F$ which, since $F \models G$ and $F \models \neg G$, implies
exists I such that $I \models G$ and $I \models \neg G$ which by definition of $\models$ implies
exists I such that $eval_I(G) = 1$ and $eval_I(\neg G) = 1$ which by definition of $eval_I(\neg)$ implies
exists I such that $eval_I(G) = 1$ and $1 - eval_I(G) = 1$ which by definition of $eval_I$ implies
exists I such that $eval_I(G) = 1$ and $eval_I(G) = 0$ which is a contradiction.

For B), also yes:
$(G \vee F) \to G$ is a tautology iff by definition of tautology
for all I, $eval_I((G \vee F) \to G) = 1$ iff by definition of $\to$
for all I, $eval_I(\neg(G \vee F) \vee G) = 1$ iff by definition of $eval_I(\vee)$
for all I, $max(eval_I(\neg(G \vee F)), eval_I(G)) = 1$ iff by definition of $eval_I(\neg)$
for all I, $max(1 - eval_I(G \vee F), eval_I(G)) = 1$ iff by definition of $eval_I(\vee)$
for all I, $max(1 - max(eval_I(G), eval_I(F)), eval_I(G)) = 1$ iff since F unsatisfiable
for all I, $max(1 - max(eval_I(G), 0), eval_I(G)) = 1$ iff by definition of $eval_I$
for all I, $max(1 - eval_I(G), eval_I(G)) = 1$, which holds if $eval_I(G) = 0$ and also if $eval_I(G) = 1$.

2) Using the Tseitin transformation, we can transform an arbitrary propositional formula F into a set of clauses $T(F)$ (a CNF with auxiliary variables) that is *equisatisfiable*: F is SAT iff $T(F)$ is SAT. Moreover, the size of $T(F)$ is linear in the size of F.

2A) Assuming $P \neq NP$, is there any transformation T' into an equisatisfiable linear-size DNF? If yes, which one? If not, why?

Answer: No (unless $P = NP$). If such a similar transformation existed, then we could solve an NP-complete problem (is F SAT?) by transforming F in linear time into the DNF $T'(F)$, and then deciding whether the DNF $T'(F)$ is satisfiable (which, as we know, can be done in linear time for DNFs).

2B) Is there any similar transformation T' into a linear-size DNF, such that F is a tautology iff $T'(F)$ is a tautology? If yes, which one? If not, why?

Answer: Yes. F is a tautology iff $\neg F$ is unsatisfiable iff the Tseitin transformation $T(\neg F)$ is unsatisfiable iff $\neg T(\neg F)$ is a tautology. And indeed $\neg T(\neg F)$ can be easily transformed into a DNF: $T(\neg F)$ is a conjunction of claues $C_1 \wedge \ldots \wedge C_n$. Its negation $\neg(C_1 \wedge \ldots \wedge C_n)$ is equivalent to $\neg C_1 \vee \ldots \vee \neg C_n$, and each $\neg C_i$ is of the form $\neg(l_1 \vee \ldots \vee l_m)$ which is equivalent to $\neg l_1 \wedge \ldots \wedge \neg l_m$. Note that, unlike what happened in the previous case, here we transform an NP-complete problem into another NP-complete problem.

3) A pseudo-Boolean constraint has the form $a_1x_1 + \ldots + a_nx_n \leq k$ (or the same with $\geq$), where the coefficients a_i and the k are natural numbers and the x_i are propositional variables. Which clauses are needed to encode the pseudo-Boolean constraint $2x + 3y + 4z + 6u + 8v \leq 10$ into SAT, if no auxiliary variables are used? Which clauses are needed in general, with no auxiliary variables, for a constraint $a_1x_1 + \ldots + a_nx_n \leq k$?

Answer: To encode $2x + 3y + 4z + 6u + 8v \leq 10$, for every (minimal) subset of variables such that the sum of its coefficients is more than 10, we forbid that all of them are true. In this case, it suffices to have five clauses: $\neg v \vee \neg y$, $\neg v \vee \neg z$, $\neg v \vee \neg u$, $\neg u \vee \neg z \vee \neg y$, $\neg u \vee \neg z \vee \neg x$ and $\neg u \vee \neg y \vee \neg x$.

Note that "minimal" here means that, for example, the clause $\neg v \vee \neg y \vee \neg x$ is not needed because it is subsumed by the stronger clause $\neg v \vee \neg y$.

In general, given a constraint $a_1x_1 + \ldots + a_nx_n \leq k$, we need one clause $\neg x_{i_1} \vee \ldots \vee \neg x_{i_k}$ for each subset $S = \{i_1 \ldots i_k\}$ of $\{1 \ldots n\}$ such that $a_{i_1} + \cdots + a_{i_k} > k$, and such that moreover S is minimal ($a_{i_1} + \cdots + a_{i_k} - a_{i_j} \leq k$ for every j with $1 \leq j \leq k$).

4) Formalize and prove by resolution that sentence D is a logical consequence of the other three. Use (among others) a binary predicate symbol $OwnsCar(x, y)$ meaning "x owns the car y".

A: Paul McCartney is rich.
B: All cars with diesel engines smell badly.
C: Rich people's cars never smell badly.
D: Paul McCartney owns no diesel car.

Answer: We prove that $A \wedge B \wedge C \wedge \neg D$ is unsatisfiable.

$A:$ $IsRich(paul)$
$B:$ $\forall x \ \ Diesel(x) \rightarrow Smells(x)$
$C:$ $\forall x \ \forall y \ \ (\ (OwnsCar(x, y) \wedge IsRich(x)) \rightarrow \neg Smells(y)\ )$
$\neg D:$ $\exists x \ \ OwnsCar(paul, x) \wedge Diesel(x)$

In clausal form:

$A:$ $IsRich(paul)$
$B:$ $\neg Diesel(x) \vee Smells(x)$
$C:$ $\neg OwnsCar(x, y) \vee \neg IsRich(x)) \vee \neg Smells(y)$
$\neg D_1:$ $OwnsCar(paul, c_x)$
$\neg D_2:$ $Diesel(c_x)$

Resolution:

$6:$ $Smells(c_x)$ (from $\neg D_2$ and B, where $\sigma = \{x = c_x\}$)
$7:$ $\neg OwnsCar(x, c_x) \vee \neg IsRich(x))$ (from C and 6, where $\sigma = \{y = c_x\}$)
$8:$ $\neg OwnsCar(paul, c_x)$ (from A and 7, where $\sigma = \{x = paul\}$)
$9:$ empty clause (from $\neg D_1$ and 8).

5A) Consider a binary function symbol s and the following first-order interpretations I and I':

I: where D_I is the set of natural numbers and where $s_I(n,m) = n+m$.

I': where $D_{I'}$ is the set of integer numbers and where $s_{I'}(n,m) = n+m$.

Write the simplest possible formula F in first-order logic with equality using only the function symbol s and the equality predicate $=$ (no other symbols), such that F is true in one of the interpretations and false in the other one. Do not give any explanations.

Answer: $F:\quad \forall x \forall y \exists z\ s(x,z) = y$ (that is, z, the difference $y - x$, is defined for all x, y)
Note: Of course it makes no sense to write anything like $\forall x \exists y\, s(x,y) = 0$, because 0 is not a symbol of the syntax of F; it is a domain element.

5B) Consider binary function symbols s and p and the first-order interpretations I and I' where D_I is the set of real numbers and I' where $D_{I'}$ is the set of complex numbers and where in both cases, s is interpreted as the sum (as before) and p is interpreted as the product. Same question as 5A: complete the formula F below, using only symbols s and p: $F:\quad \exists y \exists z\ (\ (\forall x\, p(x,y) = \ldots) \wedge p(z,z) = s(\ldots)\ )$

Answer: We express the existence of the square root z of a negative number, which does not hold in the real numbers, using the part $\exists z\, p(z,z) = s(\ldots)$. We make $s(\ldots)$ negative using $s(y,y)$ and expressing that y is -1 using the part $\exists y\, (\forall x\, p(x,y) = \ldots)$: make $xy = 2xy + x$, which implies $-(xy) = x$. Writing $xy = 2xy + x$ as $p(x,y) = s(s(p(x,y), p(x,y)), x)$, we get:

$F:\quad \exists y \exists z\ (\ (\forall x\, p(x,y) = s(s(p(x,y), p(x,y)), x)) \wedge p(z,z) = s(y,y)\ )$

Another answer: We force $y = 1$ and then $z^2 = y + 2z^2$, which implies $z^2 = -y$.

$F:\quad \exists y \exists z\ (\ (\forall x\, p(x,y) = x) \wedge p(z,z) = s(y, s(p(z,z), p(z,z)))\ )$

6A) Let F be the formula $\forall x\ p(c,x) \wedge \exists y\ (\ q(y) \vee \neg p(y,y)\ )$. Let G be the formula $\exists z\ (\ p(z,c) \vee q(z)\ )$. Do we have $F \models G$? Prove it.

Answer: Yes. We prove $F \wedge \neg G$ insat by resolution. F gives two clauses:

$F_1:\ p(c,x)$ and

$F_2:\ q(c_y) \vee \neg p(c_y, c_y)$.

The formula $\neg G$ is $\neg \exists z\ (\ p(z,c) \vee q(z)\ )$, which becomes $\forall z\ \neg p(z,c) \wedge \neg q(z)$, giving two clauses:

$G_1:\ \neg p(z,c)$ and

$G_2:\ \neg q(z)$.

By resolution between F_1 and G_1, where $\sigma = \{z = c, x = c\}$, we get the empty clause.

6B) Let F be the formula $\quad \forall x\ \ (\ p(x,x)\ \wedge\ \neg p(x, f(x))\ \wedge\ \neg p(x, g(x))\ \wedge\ \neg p(f(x), g(x))\ )$.
Is F satisfiable? If so, give a model with the smallest possible sized domain. If not, prove unsatisfiability.

Answer: Yes.

$D_I = \{0,1,2\}$

$p_I(n,m) =$ "$n = m$"

$f(n) = (n+1)\ mod\ 3$

$g(n) = (n+2)\ mod\ 3$