Lógica en la Informática / Logic in Computer Science

Monday June 15, 2015

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

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 propositional formulas over predicate symbols $\mathcal{P} = \{p_1, \ldots, p_n\}$. Let N_F denote the number of different models $I\colon \mathcal{P} \to \{0,1\}$ of the formula F, and similarly N_G for G. For each one of the following questions, give an answer that is precise as possible. Do **not** give any explanations. Just answer: "N_F=xx" or "At most xx. At least yy".

1a) If F is a tautology, what is N_F?
`Answer:` $N_F = 2^n$

1b) How many models does $F \wedge p_1$ have at most? and at least?
`Answer: At least 0. At most` $min(N_F, 2^{n-1})$.

1c) How many models does $F \vee p_1$ have at most? and at least?
`Answer: At least` $max(N_F, 2^{n-1})$ `At most` $min(2^n, N_F + 2^{n-1})$

1d) How many models does $F \vee G$ have at most? and at least?
`Answer: At least` $max\{N_F, N_G\}$. `At most` $min(2^n, N_F + N_G)$.

1e) How many models does $F \wedge G$ have at most? and at least?
`Answer: At least 0. At most` $min(N_F, N_G)$.

1f) If $F \models G$, how many models does $F \vee G$ have at most? and at least?
`Answer: At least` N_G. `At most` N_G.

`[Comment about the answers:` In general, F and G can easily contradict each other, so we can only say that y$F \wedge G$ has at least 0 models. $F \wedge G$ cannot have more models than F alone, or than G alone, so it has at most $min(N_F, N_G)$ models. For $F \vee G$: it has at least $max(N_F, N_G)$ models, and it could have up to $N_F + N_G$ models unless this number is larger than 2^n, so $F \vee G$ has at most $min(2^n, N_F + N_G)$ models. If $F \models G$, then all models of F are models of G so the models of $F \vee G$ are exactly the models of G (at least and at most N_G). Questions 1b and 1c are a particular case where G is p_1, and then we know that $N_G = 2^{n-1}$. `]`

2) Is it true that a formula F is a tautology if, and only if, its Tseitin transformation $Tseitin(F)$ is a tautology? Prove it. Your answer should be as short, clean and simple as possible, without any unnecessary explanations.

`Answer:` No. It is not true. Counterexample: let F be the tautology $p \vee \neg p$. $Tseitin(F)$ is a set of clauses with a new auxiliary root variable aux with the 1-literal clause aux (and clauses indicating that $aux \leftrightarrow p \vee \neg p$). Then if $I(aux) = 0$, we have $I \not\models Tseitin(F)$ and hence $Tseitin(F)$ is not a tautology.

`[Comment about the answers:` 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).

But when extending a model of F to a model of $Tseitin(F)$ not *all* interpretations of the auxiliary variables give a model of $Tseitin(F)$, and therefore, even if F is a tautology, as soon as a single auxiliary variable is needed, $Tseitin(F)$ is not a tautology.]

3) Please answer this question 3a and 3b on separate paper sheets. An old rich arab wants to distribute all his (many, many) goods among his (many, many) children. Since he loves all of them equally, he wants the give to each child goods of the same total value. The set of goods is $\mathcal{G} = \{g_1, \ldots, g_N\}$, each g_i with value v_i \$, and there are M children.

3a) Explain how would you solve this using SAT (which variables, clauses, and constraints). If you use any cardinality or pseudo-Boolean constraints, do not give their encoding into clauses.

`Answer:` we will use variables $x_{i,c}$ with $1 \leq i \leq N$ and $1 \leq c \leq M$, meaning that "good g_i is given to the c-th child". Using them, we will encode the following facts:

- Each good i, with $1 \leq i \leq N$, is given to at least one child:

  N clauses $x_{i,1} \vee x_{i,2} \vee \ldots \vee x_{i,M}$.

- Each good i, with $1 \leq i \leq N$, is given to at most one child:

  N cardinality constraints $x_{i,1} + x_{i,2} + \ldots + x_{i,M} \leq 1$.

- All children receive goods with the same value. This value is $V = \sum_{i=1}^{N} v_i/M$. Now each child c, with $1 \leq c \leq M$, gets goods of total value V:

  M pseudo-Boolean constraints $v_1 \cdot x_{1,c} + v_2 \cdot x_{2,c} + \ldots + v_N \cdot x_{N,c} = V$.

3b) If, in addition, the first C goods are cars, and the father only wants to give cars to his first (oldest) K children (zero or more cars to each one of these), how would you solve this using SAT?

`Answer:` We need to impose that the cars cannot be assigned to the last $M - K$ children. This is done with the unit clauses $\neg x_{i,c}$, for all $1 \leq i \leq C$ and $K < c \leq M$.

4) Write a Prolog predicate `shortest([I1,J1], [I2,J2])` that writes to the output the shortest path on a chess board a horse needs to go from square `[I1,J2]` to square`[I2,J2]`. Coordinates `I,J` are in 1..8. The path is the list of intermediate board squares. Your solution should be short, clean and simple, without any comments. `Answer:`

```
path( E,E, C,C ).
path( CurrentState, FinalState, PathUntilNow, TotalPath ):-
  oneStep( CurrentState, NextState ),
  \+member(NextState,PathUntilNow),
  path( NextState, FinalState, [NextState|PathUntilNow], TotalPath ).

oneStep([I1,J1],[I2,J2]):-   member( [A,B], [[1,2],[2,1]] ),
  member( SignA, [1,-1] ),   I2 is I1 + A*SignA,
  member( SignB, [1,-1] ),   J2 is J1 + B*SignB,   between(1,8,I2), between(1,8,J2).

shortest([I1,J1],[I2,J2]):-   between(1,64,N),
  path( [I1,J1], [I2,J2], [[I1,J1]], Path ), length( Path, N ),  write(path), nl.
```

5a) Consider the first-order predicate and function symbols $\mathcal{P} = \{p^2, q^1\}$ and $\mathcal{F} = \{f^2, g^1, a^0, b^0, c^0\}$. How many different atoms *without variables* can be constructed usng $\mathcal{P}$ and $\mathcal{F}$? Just write the amount, without giving any explanations.

`Answer: infinitely many.`

5b) Is the satisfiability of first-order formulas without variables decidable? Why? Your answer should be short, clean and simple as possible (no bla bla).

`Answer:` Yes. It is decidable. If F has no variables, then S, the clausal form of F, is finite and has no variables either. We know that F is unsatisfiable iff S is unsatisfiable iff $\Box \in ResFact(S)$ (the closure under first-order resolution and factoring of S).

But since S has no variables, resolution here is just propositional resolution, and factoring is just eliminating repeated literals in a clause. Therefore, one can consider each different atom in S as a propositional predicate symbol, getting a propositional clause set $prop(S)$ and $\Box \in ResFact(S)$ iff $\Box \in PropositionalRes(prop(S))$, which is decidable.

6) Consider the following sentences:

1. All red horses run fast or there is a horse that does not run fast.
2. There is at least one red horse.

6a) Write first-order formulas F_1 and F_2 formalizing 1. and 2. Do not write anything else here.

`Answer:`
$F_1 : (\forall x \ \ red(x) \to fast(x)) \lor (\exists y \ \ \neg fast(y))$
$F_2 : \exists x \ \ red(x)$

6b) Is F_1 a tautology? Prove it. Do not give any unnecessary explanations.

`Answer:`
Yes it is a tautology. We have F_1 taut iff $\neg F_1$ insat iff $\Box \in ResFact(clausalform(\neg F_1))$.
Turning $\neg F_1$ into clausal form:
$\neg((\forall x \ \ \neg red(x) \lor fast(x)) \lor (\exists y \ \ \neg fast(y)))$
$\neg(\forall x \ \ \neg red(x) \lor fast(x)) \land \neg(\exists y \ \ \neg fast(y))$
$(\exists x \ \ red(x) \land \neg fast(x)) \land (\forall y \ \ fast(y))$
$(red(c_x) \land \neg fast(c_x)) \land (\forall y \ \ fast(y))$
gives $\{red(c_x), \neg fast(c_x), fast(y)\}$ and by resolution between the last two clauses we get $\Box$.

6c) Do we have $F_1 \models F_2$? Prove it formally, as short and simple as you can (no bla bla).

`Answer:` No. We do not have $F_1 \models F_2$. Since we have seen that F_1 is a tautology, to prove $F_1 \not\models F_2$ we can simply give as a counterexample *any* interpretation I such that $I \not\models F_2$, that is, $I \not\models \exists x \ \ red(x)$. For this, define I such that $D_I = \{e\}$ and $red_I(e) = 0$.