Lógica en la Informática / Logic in Computer Science

Thursday May 10th, 2018

Time: 1h30min. No books, lecture notes or formula sheets allowed.

1) (3 points)
1a) Let F, G, H be propositional formulas. Is it true that always $(F \land G) \land H \equiv F \land (G \land H)$? Prove it using only the definition of propositional logic.

Answer: Yes.

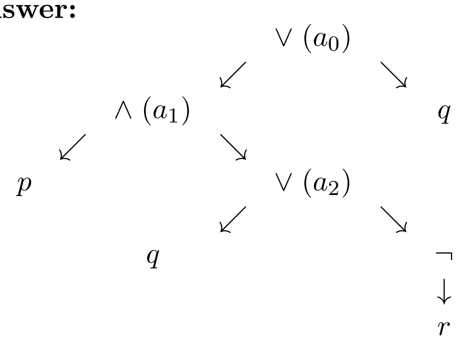
$(F \land G) \land H \equiv F \land (G \land H)$ iff — by definition of $\equiv$
$(F \land G) \land H$ and $F \land (G \land H)$ have the same models — iff, by definition of model
forall I, $I \models (F \land G) \land H)$ iff $I \models F \land (G \land H))$ — iff, by definition of $\models$
forall I, $eval_I((F \land G) \land H) = eval_I(F \land (G \land H))$ — iff, by definition of evaluation of $\land$
forall I, $min(eval_I(F \land G), eval_I(H)) = min(eval_I(F), eval_I(G \land H)$
— iff, by definition of evaluation of $\land$
forall I, $min(min(eval_I(F), eval_I(G)), eval_I(H)) = min(eval_I(F), min(eval_I(G), eval_I(H)))$
— iff, by definition of min
forall I, $min(eval_I(F), eval_I(G), eval_I(H)) = min(eval_I(F), eval_I(G), eval_I(H))$.

1b) Let F, G, H be propositional formulas. Is it true that always $F \land (G \lor H) \equiv F \lor (G \land H)$? Prove it using only the definition of propositional logic.

Answer: No. Counter example: Take $F = p$, $G = H = q$ and $I(p) = 1$ and $I(q) = 0$. Then $I \not\models p \land (q \lor q)$ but $I \models p \lor (q \land q)$.

2) (2 points) Write all clauses obtained by applying Tseitin's transformation to the formula $(p \land (q \lor \neg r)) \lor q$. Use auxiliary variables named $a_0, a_1, a_2, \ldots$ (where a_0 is for the root).

Answer:



Clauses:

one unit clause for the root: a_0
3 clauses for $a_0 \leftrightarrow a_1 \lor q$: $\neg a_1 \lor a_0$, $\neg q \lor a_0$, $\neg a_0 \lor a_1 \lor q$
3 clauses for $a_1 \leftrightarrow p \land a_2$: $\neg a_1 \lor p$, $\neg a_1 \lor a_2$, $a_1 \lor \neg p \lor \neg a_2$
3 clauses for $a_2 \leftrightarrow q \lor \neg r$: $\neg q \lor a_2$, $r \lor a_2$, $\neg a_2 \lor q \lor \neg r$

3) (4 points) John wants to buy a subset of Amazon's n products (and, as you know, with a very large n). But he has the following $1+p+q$ constraints, where all M, I_i, L_j, R_j denote subsets of $\{1 \dots n\}$:

- he *must* buy all products of M
- $I_1 \dots I_p$ are *incompatibility sets*: for each I_i, John cannot buy *all* products in I_i
- constraints $L_1 \rightarrow R_1 \ \dots \ L_q \rightarrow R_q$, where $L_i \rightarrow R_i$ means that if John buys *all* products of L_i, then he must also buy *all* products of R_i.

3a) Answer all three questions **very briefly**. What would you recommend John to do for *efficiently* finding a set S of products that he can buy without violating any of the constraints?

3b) Same question for finding a set S with minimal $|S|$.

3c) Is the minimal set S of 3b) unique or can there be several distinct minimal sets?

Answers for 3a,b,c: Express it by Horn SAT.
Variables: for each i in $\{1 \dots n\}$ a variable x_i meaning "John buys product i". (Horn) clauses:
-for each i in M, a unit clause: x_i
-for each I_i, a (purely negative) Horn clause: $\bigvee_{j \in I_i} \neg x_j$
-for each constraint $L_i \rightarrow R_i$ and for each k in this R_i, a Horn clause: $x_k \vee \bigvee_{j \in L_i} \neg x_j$
Apply the linear-time Horn SAT algorithm by positive unit propagation, which sets to true only those variables that *must* be true in any model and therefore finds the *unique minimal* model (and set S).

4) Consider the following problem, called *model counting*:
Input: a natural number k and a set of propositional clauses S over symbols $\mathcal{P}$.
Question: does S have at least k different models $I : \mathcal{P} \rightarrow \{0,1\}$?
We want to analyze the computational complexity of model counting, that is, determine if it is polynomial, NP-complete, or perhaps even harder, etc. Answer all four questions **very briefly** (max. 10 words per question).

4a) (1 point) Is model counting at least as hard as SAT? (that is, can we express SAT as a model counting problem?) Why?

Answer: Yes. A set of clauses S is SAT iff the model counting problem with input $k = 1$ and S answers "yes".

4b) (4b,c,d: 1 bonus point, if short and correct) What do you think, is SAT at least as hard as model counting? Why?

Answer: No. No way to do model counting by a polynomial number of calls to SAT is known. So SAT does not seem to be as hard. Model counting seems harder than SAT.

4c,4d) Same questions if S is a set of Horn clauses.

Answer: Same answers as before. In fact, no way to do *Horn* model counting by a polynomial number of calls to *arbitrary* SAT is known.