Lógica en la Informática / Logic in Computer Science

Nov 15th, 2012.

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

1) For each one of the following statements, indicate whether it is true or false, without giving any explanations why.

1. Let F, G, H be formulas. If $F \wedge G \models \neg H$ then $F \wedge G \wedge H$ is unsatisfiable.
2. Let F, G, H be formulas. If $F \vee G \models H$ then $F \wedge \neg H$ is unsatisfiable.
3. The formula $p \vee p$ is a logical consequence of the formula $(p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg r)$.
4. The formula $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee p)$ is unsatisfiable.
5. Assume $|\mathcal{P}| = n$. There are 2^n interpretations. Moreover there are exactly $k = 2^{2^n}$ formulas $F_1, \ldots, F_k$ such that for all i, j with $i \neq j$ in $1 \ldots k$, $F_i \not\equiv F_j$. Each one of these F_i represents a different Boolean function.
6. If F is unsatisfiable, then for every G we have $G \models F$.
7. If F is a tautology, then for every G we have $F \models G$.
8. If F es a tautology, then for every G we have $G \models F$.

2A) Let F and G be formulas. Can it happen that $F \models G$ and $F \models \neg G$? Prove it.

2B) Let F be a formula. Is it true that F is satisfiable if, and only if, all logical consequences of F are satisfiable formulas? Prove it.

3) What is the complexity of deciding the satisfiability of an input formula in DNF? Explain why.

4) After the general strike yesterday, the n workers of Iberia want to create a national comittee with k members. Each worker proposes a list of 10 names out of the list of n workers. The objective is that, for each worker, at least one of the names of his/her list is in the commitee. We want to decide whether this is possible or not.

4A Do you think that this problem is polynomial? How would you solve it using SAT?

4B Answer the same questions if each worker proposes only 2 names.

Lógica en la Informática / Logic in Computer Science

Thursday April 18th, 2013

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1A) Is it true that if F is unsatisfiable then $\neg F$ is a tautology? Prove it using only the formal definitions of propositional logic.

1B) Is it true that if F, G, H are formulas such that $F \wedge G \not\models H$ then $F \wedge G \wedge H$ is unsatisfiable? Prove it using only the formal definitions of propositional logic.

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) Is there any transformation T' into an equisatisfiable linear-size DNF? If yes, which one? I If not, why?

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?

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 $2x + 3y + 5z + 6u + 8v \leq 11$ into SAT, if no auxiliary variables are used? Which clauses are needed in general, with no auxiliary variables, for a constraint $a_1x_1 + \ldots + a_nx_n \leq k$?

4) For organizing the general elections in Ecuador, we need to decide where to locate the polling places (the places where people can vote). To do this, we have a long list $L = \{1 \ldots N\}$ of possible places (schools, town halls, etc), and for each inhabitant i of the m inhabitants of Ecuador ($i \in \{1 \ldots m\}$), a sublist L_i of L with those places that are close enough to i's home. To save costs, we would like to only open K polling places, but of course guaranteeing that every inhabitant i can vote at one of the places on its list L_i. Which (and how many) variables and clauses do we need to solve this problem using SAT?

5) We have a computing facility with N identical computers, and we need to handle T computing tasks, with $T > N$. Each task $i \in \{1 \ldots T\}$ has a duration of d_i seconds. Each computer can handle no more than one task at the same time. We want to determine whether it is possible to distribute the tasks over these computers in such a way that after less than K seconds all tasks are finished. Which (and how many) variables and clauses do we need to solve this problem using SAT? Note: if needed, you can leave arithmetical constraints (at-most-one, cardinality, pseudo-Boolean...) without encoding them into SAT.

Lógica en la Informática / Logic in Computer Science

Tuesday November 26th, 2013

Time: 1h55min. No books, lecture notes or formula sheets allowed.

1A) Given two propositional formulas F and G, is it true that $F \rightarrow G$ is a tautology iff $F \wedge \neg G$ is unsatisfiable? Prove it using only the formal definitions of propositional logic.

1B) Given two propositional formulas F and G, is it true that $F \models G$ iff $F \rightarrow G$ is satisfiable? Prove it using only the formal definitions of propositional logic.

1C) What is the complexity of Horn-SAT? What is the complexity of 2-SAT? What do you think is the complexity of HornOrTwo-SAT, that is, deciding the satisfiability of sets S of clauses such that every clause in S is either Horn or has at most two literals?

2) Consider the *at most one* constraint (AMO), saying that among n literals $l_1 \dots l_n$, at most one can be true, that is, $l_1 + \dots + l_n \leq 1$.
2A) Write the name of the encoding you know for this constraint that needs the fewest auxiliary variables. How many (as a function of n)?

2B) Write the name of the encoding for this constraint that needs the fewest clauses. How many?

2C) We call such an AMO encoding *arc-consistent for unit propagation* if, when one of $l_1 \dots l_n$ becomes true, the SAT solver's unit propagation mechanism will set the other literals to false. This is a good property. Does the encoding you gave as a answer for 2A fulfill it?

2D) Write all clauses needed to express the cardinality constraint $l_1 + \dots + l_5 \geq 2$ without using any auxiliary variables (do not write any unnecessary clauses).

2E) Write all clauses needed to express the Pseudo-Boolean constraint $7x+4y+5z+3u+8v+9w \leq 16$ without using any auxiliary variables (do not write any unnecessary clauses).

3) We need to plan the activities of a transportation company during a period of H hours. The company has T trucks, D drivers and there are N transportation tasks to be done, each one of which lasts one hour and needs one driver per truck.
Each task $i \in 1 \dots N$ needs K_i trucks, and has a list $L_i \subseteq \{1 \dots H\}$ of hours at which this task i can take place. For example, if $L_7 = \{3, 4, 8\}$ this means that task 7 can take place at hour 3, at hour 4 or at hour 8.
For each driver $d \in 1 \dots D$ there is a list of blockings $B_d \subseteq \{1 \dots H\}$ of hours at which driver D can *not* work.
3A) Explain how to use a SAT solver for planning this: for each task, when does it take place, and using which drivers. Clearly indicate which types of propositional variables you are using, and how many of each type, using the following format:

variables $t_{i,h}$ meaning "task i takes place at hour h"
for all tasks $i \in 1 \dots N$ and for all hours $h \in 1 \dots H$
Total: $N \cdot H$ variables.

Since H, D and N may be large, it is not allowed to use $O(H \cdot D \cdot N)$ variables (but using such a large number of clauses is fine). Hint: you many use several types of variables, for example one type with $N \cdot H$ variables and another one with $N \cdot D$.

Also clearly indicate which clauses you need, and how many of each type, and how many literals each type of clause has. If you use any AMO, cardinality or pseudo-Boolean constraints, it is not necessary to convert these into CNF.

3B) Extend your solution to take into account that no driver can work more than 50 hours in total during the whole period of H hours. Hint: this may require another type of variables.

Lógica en la Informática / Logic in Computer Science

Tuesday April 22nd, 2014

Time: 1h55min. No books, lecture notes or formula sheets allowed.

1A) Let F and G be two propositional formulas such that $F \models G$. Is it true that $F \equiv F \land G$? Prove it using only the formal definitions of propositional logic.
1B) Given two propositional formulas F and G, is it true that either $F \models G$ or $F \models \neg G$? Prove it using only the formal definitions of propositional logic.

2) If S is a set of clauses, let us denote by $UP(S)$ the set of all literals that can be obtained from S by zero or more steps of unit propagation. Imagine you have a C++ program P that does unit propagation in linear time, taking as input any set of clauses S and returning $UP(S)$. Explain your answers to the following questions:
2A): Is it true that $l \in UP(S)$ implies $S \models l$?

2B): Let l be any literal. Is it true that $S \models l$ implies $l \in UP(S)$?

2C): Can you use your program P to decide 2-SAT in polynomial time?

2D): Can you use your program P to decide Horn-SAT in polynomial time?

3A) Write all clauses needed to express the cardinality constraint $x_1 + \cdots + x_6 \leq 4$ without using any auxiliary variables (do not write any unnecessary clauses).

3B) Write all clauses needed to express the Pseudo-Boolean constraint $1x + 3y + 4z + 5u + 8v \geq 14$ without using any auxiliary variables (do not write any unnecessary clauses). Hint: write one clause for each (minimal) subset S of the variables such that not *all* variables of S can be false.

4) We want to use a SAT solver to do *factoring*: given a natural number n, find two natural numbers p and q with $p \geq 2$ and $q \geq 2$, such that $n = p \cdot q$. Of course, the SAT solver will return "unsatisfiable" if and only if n is a prime number. (Curiosity: if we could factor large n, we could break many cryptographic systems!).

4A) Let a and b be bits (propositional variables). Write the seven clauses meeded to express that the two-bit number $c\,d$ is the result of the sum $a+b$, that is, c is the "carry" ($c = a \land b$) and d means "exactly one of a, b is 1" (exclusive or: $c = xor(a, b)$).

4B) Here we will factor numbers n of four bits $n_3\, n_2\, n_1\, n_0$ only, so $n \leq 15$. This means that, since we want to find $p \geq 2$ and $q \geq 2$, we know that $p < 8$ and $q < 8$ so for p and for q three bits each are sufficient, which we will call $p_2\, p_1\, p_0$ and $q_2\, q_1\, q_0$. Graphically, we can express the multiplication as we would do it "by hand":

$$
\begin{array}{ccccc}
 & & p_2 & p_1 & p_0 \\
 & & q_2 & q_1 & q_0 \\
\cline{3-5}
 & & x_2 & x_1 & x_0 \\
 & y_2 & y_1 & y_0 & 0 \\
z_2 & z_1 & z_0 & 0 & 0 \\
\hline
0 & n_3 & n_2 & n_1 & n_0
\end{array}
$$

using 9 intermediate auxiliary variables (called x, y, z, with subindices), where in fact we already know that z_2 must be 0. Using these auxiliary variables, and a few other auxiliary variables expressing the "carries" (please call them c_*), write here the expressions, like $n_1 = xor(x_1, y_0)$, cardinality constraints, etc., needed to ensure that indeed $n = p \cdot q$. After that, write the concrete clauses needed for each expression.

Lógica en la Informática / Logic in Computer Science

Tuesday November 18th, 2014

Time: 1h345min. No books, lecture notes or formula sheets allowed.

1A) (1 pt) Let F and G be propositional formulas. Is it true that if $F \to G$ is a tautology and F is satisfiable, then G is a tautology? Prove it using only the definition of propositional logic.

1B) (1 pt) 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? Prove it using only the definition of propositional logic.

2) (2 pts) Let S be a set of propositional clauses and let C be a propositional clause $l_1 \vee \ldots \vee l_n$. We want to use a SAT solver in order to know whether $S \models C$. One Erasmus student says: "That's easy! It is true if the SAT solver finds a model for $S \cup \{\neg l_1, \ldots, \neg l_n\}$". Is he right? Explain in detail why.

3) (2.5 pts) The engineers at Intel have two very large digital circuits C_1 and C_2, built with binary **and** and **or** gates and unary **not** gates (inverters). Each one of them has n input wires `w1...wn` and 1 output wire `w-out`, and they want to know whether the two circuits are equivalent, that is, whether they give the same output bit for every combination of n input bits. Each circuit is given in a simple format with one line per gate and inner wires represented by numbers, for example:

```
w-out=and(w1,1)
    1=or(2,3)
    2=and(w2,3)
    3=and(4,w4)
    4=not(w3)
```

Note that circuits are not always trees (as formulas are). They can be directed acyclic graphs because subcircuits can be shared, as it happens with wire `3` in the example. Explain in the simplest possible way what you would do for using the picosat SAT solver to determine whether C_1 and C_2 are equivalent.

4) (3.5 pts) The Seat factory in Martorell has a single production line, on which every day N cars are produced in a certain sequence (a linear order: one car after the other). Each car needs zero or more special *features* (demanded by the buyer who has ordered the car: leather seats, sunroof, special music equipment, special paint, etc.). In total there are 40 possible features, that are installed in the car in 40 workshops located along the production line, and there are constraints saying things like: "at most 3 out of every 7 *consecutive* cars can get a sunroof". This is because these workshops only have a limited work capacity: more sunroofs close together would slow down the production line. So the information for a specific day, written in Prolog style, would look like this:

```
car(1,[3,5,8,10]).  % car 1 needs features 3,5,8,10
...
car(N,[2,6,40]).    % car N needs features 2,6,40

feature(1,5,10).    % at most 5 cars with feature 1 in each consecutive 10 cars
...
feature(40,3,7).    % at most 3 cars with feature 40 in each consecutive 7 cars
```

4A) Explain in detail how to use a SAT solver for deciding in which order we can produce the N cars. Clearly indicate which variables you use and their precise meaning, and which properties you impose using which clauses or which constraints (for cardinality or pseudo-Boolean constraints, it is not necessary to write their encodings into clauses).

4B) Imagine the problem becomes too difficult for the SAT solver. Explain how you would exploit the fact that there are groups of identical cars, i.e., cars needing the same subset of features.

Lógica en la Informática / Logic in Computer Science

Tuesday April 21st, 2015

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1) Which is the minimal known computational cost of deciding the satisfiability of the following types of propositional formulas (along this whole exam: just write the answer; do not explain why, unless you are asked to do so explicitly):

a: Arbitrary propositional formulas:

b: 3-SAT:

c: 2-SAT:

d: DNF;

e: CNF;

f: Horn-SAT:

2)
2a: What is the minimal known computational cost of deciding whether a given propositional formula F is a tautology?

2b: What is the minimal known computational cost of deciding whether a given propositional formula F in CNF is a tautology? Why?

2c: What is the minimal known computational cost of transforming a given propositional formula F into a logically equivalent formula G in CNF, without using any new auxiliary variables?

2d: Write the result of applying the Tseitin transformation to $p \vee (q \wedge r) \vee (\neg p \wedge r)$.

2e: Let G be the Tseitin transformation of a given propositional formula F. What is the size of G with respect to F? What are the properties of G with respect to F?

3) Let F and G be propositional formulas. For each of the following two statements, say whether it is right or wrong, and prove why:

3a: $F \models G$ implies $F \equiv F \wedge G$

3b: Always $F \models G$ or $F \models \neg G$.

4) A factory produces rolls of 100m of cable. The factory has n orders from customers. Each order i in $1..n$ is for buying k_i meters of cable, $k_i \leq 100$. All orders have to be served by cutting the pieces of 100m, without using more than K rolls.

4a: Explain in detail how to use a SAT solver for solving this. Clearly indicate which variables you use and their precise meaning, and which properties you impose using which clauses or which constraints (for cardinality or pseudo-Boolean constraints, it is not necessary to write their encodings into clauses).

4b: Answer the same questions, assuming that for each order i in $1..n$ we also know the price p_i (in euros) each customer will pay for its k_i meters of cable, and the factory wants to know whether it can obtain an income of at least 10.000 euros with its K rolls by serving *some of* the orders.

Lógica en la Informática / Logic in Computer Science

Wednesday November 11, 2015

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1a) Let F, G, H be formulas. Is it true that if $F \vee G \models H$ then $F \wedge \neg H$ is unsatisfiable? Prove it using only the definition of propositional logic.
1b: Let F, G be formulas. Is it true that always $F \models G$ or $F \models \neg G$? Prove it using only the definition of propositional logic.

2) Consider the following decision problem, called "minOnes":
Input:
a natural number k and a propositional formula F in CNF over propositional variables $\{x_1 \ldots, x_n\}$
Question:
Is there any model I of F with at most k ones, i.e., where $\mid \{x_i \mid 1 \leq i \leq n \text{ and } I(x_i) = 1\} \mid \leq k$?

2a) Do you think that minOnes is NP-complete? Why?

2b) How would you use a SAT solver to decide it?

2c) How would you use a SAT solver to solve the optimization version of minOnes, that is, given F, to find its model with the smallest possible number of ones?

3) The very large catalan supermarket CATSUP is open every day during 10 hours (from 10am to 8pm). It wants to schedule the working times of its N employees during a 30-day period.

For each hour h in this 300-hour period, CATSUP has made a prediction of the number N_h of employees needed (at least) during hour h.

Each employee i in $1..N$ has to work exactly 160 hours in this 30-day period, always at least 9 hours per working day, and no employee gets more than 5 consecutive working days in a row.

Explain in detail how to use a SAT solver for deciding exactly when each employee has to work. Clearly indicate which types of propositional variables you use and their precise meaning, and which properties you impose using which clauses or which constraints. For cardinality or pseudo-Boolean constraints, it is not necessary to give their encodings into clauses. Your solution should be as efficient and simple as possible.

Lógica en la Informática / Logic in Computer Science

Wednesday April 27th, 2016

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1) Given two propositional formulas F and G, is it true that $F \to G$ is a tautology iff $F \models G$? Prove it using only the formal definitions of propositional logic.

2) Let F be a formula. Is it true that F is satisfiable if, and only if, all logical consequences of F are satisfiable formulas? Prove it using only the formal definitions of propositional logic.

3) What is Horn-SAT? What is its computational complexity? Explain very briefly why.

4) Consider the following decision problem, called "MaxSAT":
Input: A natural number k and a set S of n propositional clauses over propositional symbols $\mathcal{P}$.
Question: Is there any interpretation $I : \mathcal{P} \to \{0, 1\}$ that satisfies at least k clauses of S?

4b) How would you use a SAT solver to decide it?

4c) How would you use a SAT solver to solve the optimization version of MaxSAT, that is, how to find the I that satisfies as many of the clauses of S as possible?

Lógica en la Informática / Logic in Computer Science

Tuesday November 22nd, 2016

Time: 1h45min. No books, lecture notes or formula sheets allowed.

1) Let S be a set of propositional clauses over a set of n predicate symbols, and let $Res(S)$ be its closure under resolution. For each one of the following cases, indicate whether $Res(S)$ is infinite or finite, and, if finite, of which size. Very briefly explain why.

1a) If clauses in S have at most two literals.

1b) S is a set of Horn clauses.

1c) Every clause in S has either two literals or is a Horn clause.

1d) S is an arbitrary set of propositional clauses.

2) Let C_1 and C_2 be propositional clauses, and let D be the conclusion by resolution of C_1 and C_2.

2a) Is D a logical consequence of $C_1 \vee C_2$? Prove it formally, using only the definitions of propositional logic.

2b) Is D a logical consequence of $C_1 \wedge C_2$? Prove it formally, using only the definitions of propositional logic.

2c) Let S be a set of propositional clauses and let $Res(S)$ be its closure under resolution. Is it true that S is satisfiable if, and only if, the empty clause is not in $Res(S)$? Very briefly explain why.

3) Consider the well-known NP-complete *vertex cover* problem: Given a natural number k and a graph with n vertices and m edges $\{(u_1, v_1), \ldots, (u_m, v_m)\}$ with $u_i, v_i \in \{1 \ldots n\}$, it asks whether the graph has a k-cover, that is, a subset of size k of the vertices such that for each edge (u_i, v_i) at least one of u_i and v_i is in the cover.

3a) How would you use a SAT solver to decide it?

3b) How would you use a SAT solver for the optimization version of vertex cover, that is, given only the graph, to find its smallest possible k-cover?

3c) In one exam last year, we considered the following decision problem, called *minOnes*: given a natural number k and a set S of clauses over variables $\{x_1, \ldots, x_n\}$, it asks whether S has any model I with at most k ones, that is, with $I(x_1) + \ldots + I(x_n) \leq k$. Its optimization version is, given only S, to find its model with the minimal number of ones. If the set of clauses S only has clauses with at most two literals, does the optimization version of minOnes become polynomial? Explain briefly why.

Last names: ... **1st name: ...** **DNI: ...**

Lgica en la Informtica / Logic in Computer Science

Permutation B. Tuesday April 18th, 2017
Time: 1h45min. No books, lecture notes or formula sheets allowed.

1) Let us remember the *Heule-3 encoding* for *at-most-one* (*amo*) that is, for expressing in CNF that at most one of the literals $x_1 \dots x_n$ is true, also written $x_1 + \dots + x_n \leq 1$. It uses the fact that $amo(x_1 \dots x_n)$ iff $amo(x_1, x_2, x_3, aux)$ AND $amo(\neg aux, x_4 \dots x_n)$. Then the part $amo(\neg aux, x_4 \dots x_n)$, which has $n-2$ variables, can be encoded recursively in the same way, and $amo(x_1, x_2, x_3, aux)$ can be expressed using the quadratic encoding with 6 clauses. In this way, for eliminating two variables we need one auxiliary variable end six clauses, so in total we need $n/2$ variables and $3n$ clauses.

1a We now want to extend the encoding for *at-most-two* (*amt*, also written $x_1 + \dots + x_n \leq 2$). Prove that $amt(x_1 \dots x_n)$ has a model I iff $amt(x_1, x_2, x_3, aux_1, aux_2) \wedge amt(\neg aux_1, \neg aux_2, x_4 \dots x_n)$ has a model I', with $I(x_i) = I'(x_i)$ for all i in $1 \dots n$.

1b Write all clauses for encoding $amt(x_1, x_2, x_3, aux_1, aux_2)$ with no more auxiliary variables.

1c How many clauses and auxiliary variables are needed in total for $amt(x_1 \dots x_n)$ in this way?

1d The Heule-3 encoding for $amo(x_1, \dots, x_n)$ has a good property: if one of the literals x_i becomes true, all other literals in $x_1, \dots, x_n$ are set to false **by unit propagation**. Does this *amt* encoding have such a property?, that is, if two of $x_1 \dots x_n$ become true, will unit propagation set the other variables to false? Explain why.

2) Every propositional formula F over n variables can also expressed by a Boolean circuit with n inputs and one output. In fact, sometimes the circuit can be much smaller than F because each subformula only needs to be represented once. For example, if F is

$$x_1 \wedge (x_3 \wedge x_4 \ \vee \ x_3 \wedge x_4) \quad \vee \quad x_2 \wedge (x_3 \wedge x_4 \ \vee \ x_3 \wedge x_4),$$

a circuit for F with only five gates, representing the output of each logical gate as a new variable (a natural number, and using 0 as the output), is:

```
0 = or(1,2)        1 = and(x1,3)        3 = or(4,4)
                   2 = and(x2,3)        4 = and(x3,x4)
```

Explain **very briefly** how you would use a standard SAT solver for CNFs to **efficiently** determine whether two circuits C_1 and C_2, represented like this, are logically equivalent.

3) For each one of the following statements, indicate here whether it is true or false without giving any explanations why.

1. If F is unsatisfiable, then for every G we have $G \models F$.
2. If F is unsatisfiable, then for every G we have $F \models G$.
3. Let F, G, H be formulas. If $F \vee G \models H$ then $F \wedge \neg H$ is unsatisfiable.
4. The formula $p \vee p$ is a logical consequence of the formula $(p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg r)$.
5. The formula $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee p)$ is unsatisfiable.
6. If F is a tautology, then for every G we have $F \models G$.
7. Let F, G, H be formulas. If $F \wedge G \not\models H$ then $F \wedge G \wedge H$ is unsatisfiable.
8. Let F, G, H be formulas. If $F \wedge G \models \neg H$ then $F \wedge G \wedge H$ is unsatisfiable.
9. If F es a tautology, then for every G we have $G \models F$.
10. Assume $|\mathcal{P}| = n$. There are 2^n interpretations. Moreover there are exactly $k = 2^{2^n}$ formulas $F_1, \dots, F_k$ such that for all i, j with $i \neq j$ in $1 \dots k$, $F_i \not\equiv F_j$. Each one of these F_i represents a different Boolean function.

Lógica en la Informática / Logic in Computer Science

Friday November 24, 2017

Permutation B. Time: 1h20min. No books, lecture notes or formula sheets allowed.

1) Below F, G, H denote arbitrary propositional formulas. Mark with an `X` the boxes of the true statements (give no explanations).

1. If $F \wedge G \not\models H$ then $F \wedge G \wedge H$ is unsatisfiable. ☐
2. If F es a tautology, then for every G we have $G \models F$. ☐
3. If F is unsatisfiable then $\neg F$ is a tautology. ☐
4. If $F \wedge G \models \neg H$ then $F \wedge G \wedge H$ is unsatisfiable. ☐
5. If $F \vee G \models H$ then $F \wedge \neg H$ is unsatisfiable. ☐
6. The formula $p \vee p$ is a logical consequence of $(p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg r)$. ☐
7. If F is unsatisfiable, then for every G we have $G \models F$. ☐
8. It can happen that $F \models G$ and $F \models \neg G$. ☐
9. The formula $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee p)$ is unsatisfiable. ☐
10. If F is a tautology, then for every G we have $F \models G$. ☐
11. If F is unsatisfiable then $\neg F \models F$. ☐
12. F is satisfiable if, and only if, all logical consequences of F are satisfiable formulas. ☐

2) Let C_1 and C_2 be propositional clauses, and let D be the conclusion by resolution of C_1 and C_2.

2a) Is D a logical consequence of $C_1 \wedge C_2$? Prove it formally, using only the definitions of propositional logic.

2b) Let S be a set of propositional clauses and let $Res(S)$ be its closure under resolution. Is it true that $S \equiv Res(S)$? Very briefly explain why.

3) Every propositional formula F over n variables can also expressed by a Boolean circuit with n inputs and one output. In fact, sometimes the circuit can be much smaller than F because each subformula only needs to be represented once. For example, if F is

$$x_1 \wedge (x_3 \wedge x_4 \;\vee\; x_3 \wedge x_4) \quad \vee \quad x_2 \wedge (x_3 \wedge x_4 \;\vee\; x_3 \wedge x_4),$$

a circuit C for F with only five gates exists. Representing the output of each logical gate as a new auxiliary variable a_i and using a_0 as the output of C, we can write C as:

```
a0 = or(a1,a2)     a1 = and(x1,a3)        a3 = or(a4,a4)
                   a2 = and(x2,a3)        a4 = and(x3,x4)
```

Explain **very briefly** how you would use a standard SAT solver for CNFs to **efficiently** determine whether two circuits C_1 and C_2, represented like this, are logically equivalent. Note: assume different names $b_0, b_1, b_2 \ldots$ are used for the auxiliary variables of C_2.