Lógica en la Informática / Logic in Computer Science

January 23, 2012. Results published: Wed Jan 25. Review: Thu Jan 26, 10h, Omega 139.

Time: 2h30min. 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 $x_1 \dots x_n$ is true. 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.

Also remember: an encoding for *amo* is *arc-consistent for unit propagation* if, when one of $x_1 \dots x_n$ becomes true, the SAT solver's unit propagation mechanism will set the other variables to false.

1a Is this Heule-3 encoding for *amo* arc-consistent for unit propagation? Prove it.

1b We now want to extend the encoding for *at-most-two* (*amt*) constraints. 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'.

1c Explain how to encode the part $amt(x_1, x_2, x_3, aux_1, aux_2)$ with no more auxiliary variables.

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

1e Is this *amt* encoding arc-consistent for unit propagation? (That is, if two of $x_1 \dots x_n$ become true, will unit propagation set the other variables to false?) Prove it.

2) Facebook Catalunya has all the information about its N registered Catalan users and for each user, the list of her friends. Now they want to find a subset of 200 Catalan users that are all friends of each other (every two users in the subset are friends). Explain in detail how they can do this using the Barcelogic SAT Solver.

3) Now we want to solve problem 2) in Prolog. Assume there are 500,000 users and 500,000 clauses like: `friends(3454,[3,7,11,23,37854]).` meaning that (all) the friends of user `3454` are `3,7,11,23` and `37854`.

3a) Define a predicate `list200(L)` that can generate in L under backtracking all lists of 200 different users (200 numbers in $1 \dots 500{,}000$).

3b) Define a predicate `friendsforever` that writes a list of 200 friends of each other, if it exists.

3c) This predicate is called `friendsforever` because it may run for a long time (almost forever). Modify your program to make it faster.

4) Formalize and prove by resolution that sentence D is a logical consequence of the other three:
A: All politicians sometime lie.
B: Friends of football players never lie.
C: Messi is a football player.
D: Messi has no friends that are politicians.

Lógica en la Informática / Logic in Computer Science

Jan 16th, 2013.

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

Note: Questions 1-4 correspond to the propositional logic part.

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 $F \wedge G \not\models H$ then $F \wedge G \wedge H$ is unsatisfiable.
2. If F is unsatisfiable then $\neg F$ is a tautology.
3. The formula $\forall x\ p(x) \vee \forall x\ q(x)$ is a logical consequence of the formula $\forall x\ (p(x) \vee q(x))$.
4. The formula $\forall x\ (p(x) \vee q(x))$ is a logical consequence of the formula $\forall x\ p(x) \vee \forall x\ q(x)$.
5. The formula $\forall x\ p(x) \vee \forall y\ \neg p(y)$ is a tautology.
6. If F is unsatisfiable, then for every G we have $F \models G$.

2A) Let F be a formula. Is it true that if F is unsatisfiable then $\neg F \models F$? Prove it.

2B) If $F \to G$ is a tautology and F is a tautology, is it true that then G is a tautology? Prove it.

3) Consider the particular restriction of propositional resolution where at least one of the two premises must be a unit clause (a clause with just one literal L):

$$\frac{L \qquad L' \vee C}{C}$$ where L is some predicate symbol p and L' is $\neg p$, or vice versa

Is this deduction rule refutationally complete? Prove it.

Identify a class of clauses, the more general the better, for which this rule is refutationally complete. Explain why.

4) After consulting a dietician, we have a list of N nutrients that one must eat at least once a week. We also know, for each one of the P products of a grocery store, which of those nutrients it contains. Given an integer K, we want to know whether it is possible to buy at most K products so that all N nutrients are included in at least one of the products. Write a propositional formula that, with the help of a SAT solver, will allow you to solve this problem.

5) Consider the following four sentences:

A: Jack is Rob's brother.

B: Mike is Rob's brother.

C: If one person is someone's brother, then this second person also is a brother of the first one.

D: Jack is Mike's brother.

Translate each one of these sentences into first-order logic. Prove (by reasoning about interpretations) that $A \wedge B \wedge C \not\models D$. Write in first-order logic an additional clause E that defines a general property of brothers, such that $A \wedge B \wedge C \wedge E \models D$, and prove $A \wedge B \wedge C \wedge E \models D$ by resolution.

6) John, Paul and Ringo are rich. Here we will count their money in natural numbers that represent millions of Euros. They all have more than zero and at most 10. John has at least twice the amount that Ringo has. Paul has at least 3 more than John. Ringo has at least 3.

A: Write all solutions.

B: Write a Gnu Prolog program with finite domain constraint propagation to find all solutions.

C: What are the domains of each variable after fully propagating the constraints (before the labeling)?

7) In order to have a nice studying place at home, we decide to build a table. After some investigation, we know that we need the following tools:

```
E = [screws,hammer,saw,screwdriver,wood,paint,brush].
```

Once we go to the appropriate shop, we realize that in order to save money we must buy packs of tools. The following packs, all with the same price, are available:

```
P = [[screws,screwdriver],
     [wood,brush,paint],
     [hammer,wood],
     [saw,screws],
     [paint,brush],
     [hammer,saw,screwdriver] ].
```

Construct a Prolog predicate `buy(E,P)` that, given a set of tools `E` and a pack list `P`, writes the minimum number of packs that one needs to buy in order to have all necessary tools.

In this case, a possible output would be:

```
Number of packs: 3
[[screws,screwdriver],[wood,brush,paint],[hammer,saw,screwdriver]]
```

Lógica en la Informática / Logic in Computer Science

Friday June 7th, 2013

Time: 2h45min. No books, lecture notes or formula sheets allowed.
The part about propositional logic comprises the first three questions.

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

2) Consider a set of propositional clauses F, a clause $C \vee l_1 \in F$ and a binary clause $l_1 \vee l_2 \in F$. Prove, by reasoning about interpretations, that $F \setminus \{C \vee l_1\} \cup \{C \vee l_1 \vee \neg l_2\} \equiv F$.

Imagine we want to check the satisfiability of a set of clauses $\{p \vee q,\ q \vee r,\ q \vee \neg r\} \cup G$. Explain why, by applying the previous result, one can remove the clause $p \vee q$ and only check the satisfiability of $\{q \vee r,\ q \vee \neg r\} \cup G$.

3) We have a list of n professional programmers $\{p_1 \dots p_n\}$ and a set of m programming tasks $\{t_1 \dots t_m\}$. We know, for each programmer p_i, the amount of euros e_i (s)he charges; that is, p_i receives e_i euros in total iff (s)he does at least one task (*not* e_i per task, but in total). For each task t_i, we also know which subset $S_i \subseteq \{p_1 \dots p_n\}$ of programmers have the right skills to do the task t_i.

We want to know whether we can handle all tasks with less than K euros. Which (and how many) variables and clauses do we need to do this using SAT? Note: if needed, you can leave arithmetical constraints (at-most-one, cardinality, pseudo-Boolean...) without encoding them into SAT.

4) Suppose we describe undirected graphs in Prolog using predicates as in the following example:

```
vertices([1,2,3,4]).
edge(1,2).
edge(1,4),
edge(2,3),
edge(3,4),
nonedge(1,3).
nonedge(2,4).
```

Program in Prolog a new predicate `tree(T)` which means that `T` is a subset of the vertices that forms a tree: all vertices in `T` are connected but there are no cycles within `T`. Also give all auxiliary predicates.

5) Using the CLP module for finite domains (clpfd) of SWI prolog, we can force labelings of the variables to *maximize* a given expression. For example, the following code writes `[5,5]`:

```
[X,Y] ins 1..10,  X+Y #= 10, labeling([max(X*Y)],[X,Y]),  write([X,Y]).
```

Now you are flying back from China, and you should write such a program to compute how many units of each one of six products you shoould take in your suitcase with capacity 80Kg, if you want to maximize the total value, and the products have the following weights (Kg) and values (Euros):

	p_1	p_2	p_3	p_4	p_5	p_6
weight:	1	2	3	5	6	7
value:	1	4	7	11	14	15

6) Formalize in first-order logic and prove by resolution that $A \wedge B \wedge C \wedge D \models E$:

A) All owners of crazy dogs are stupid
B) Everything has some owner
C) There are no stupid people
D) Pluto is a crazy dog
E) Mourinho is very well educated

Solution: We introduce n propositional variables x_i meaning "programmer i works" (i.e., "programmer i does one or more tasks"). For each task t_i, we need one clause of length $|S_i|$, saying that at least one programmer of the set S_i works: $\bigvee_{p_j \in S_i} x_j$. We also need one pseudo-Boolean constraint saying that the programmers that work cost less than K euros in total: $e_1 x_1 + \cdots + e_n x_n < K$. The number of clauses needed for this pseudo-Boolean constraint depends on the encoding used.

```
tree(T):- vertices(V), subset(V,T), connected(T), \+hasCycle(T).

connected([]):-!.
connected([X|V]):- con([X],V).  % here [X] is the already connected part
con(_,[]).  % Below, find vertex Y to add to the already connected part C:
con(C,V):- memberRest(Y,V,Rest), member(X,C), edge1(X,Y), !, con( [Y|C], Rest ).

hasCycle(T):- subset(T,C), permutation(C,P), P=[First|_], isCycle(First,P).
isCycle(First,[Last] ):- edge1(Last,First),!.
isCycle(First,[X,Y|L]):- edge1(X,Y), isCycle( First, [Y|L] ), !.

%%%%% well-known auxiliary predicates:
subset([],[]).
subset([X|L],[X|S]):-subset(L,S).
subset([X|L],   S ):-subset(L,S).
permutation( [X|L], Perm ):- permutation(L,P), append(P1,P2,P), append(P1,[X|P2],Perm).
memberWithRest(X,Set,Rest):- append(A,[X|B],Set),  append(A,B,Rest).
edge1(X,Y):- edge(X,Y). % edge1: edges in both directions
edge1(X,Y):- edge(Y,X).
```

Solution:

```
china:-  L = [A,B,C,D,E,F],
         L ins 0 .. 80,   1*A + 2*B + 3*C +  5*D +  6*E +  7*F #=< 80,
         labeling( [ max( 1*A + 4*B + 7*C + 11*D + 14*E + 15*F ) ], L ), write(L), nl,!.
```

Solution: We need to prove that $A \wedge B \wedge C \wedge D \wedge \neg E$ is unsatisfiable. In fact, here $A \wedge B \wedge C \wedge D$ is already unsatisfiable (and E has nothing to do with them). Note that, intuitively, Pluto is a crazy dog (by D) with some owner (by B) that is stupid (by A), and this contradicts C, so one needs all four these sentences.

A) All owners of crazy dogs are stupid (here $O(x, y)$ means "x owns y"):
$\forall x(\exists y O(x, y) \wedge CD(y)) \to S(x)$
$\forall x \neg(\exists y O(x, y) \wedge CD(y)) \vee S(x)$
$\forall x \forall y \neg O(x, y) \vee \neg CD(y) \vee S(x)$

B) Everything has some owner:
$\forall x \exists y O(y, x)$
$\forall x O(f_y(x), x)$

C) There are no stupid people:
$\forall x \neg S(x)$

D) Pluto is a crazy dog:
$CD(pluto)$

So we get four clauses:
A) $\neg O(x, y) \vee \neg CD(y) \vee S(x)$
B) $O(f_y(z), z)$
C) $\neg S(v)$
D) $CD(pluto)$.

By resolution we get:
1) $\neg O(x, pluto) \vee S(x)$ (from A and D)
2) $S(f_y(pluto))$ (from 1 and B)
3) [] (from 2 and C)

Lógica en la Informática / Logic in Computer Science

Thursday January 9th, 2014

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

Questions 1,2,3 are the part of propositional logic, and 4,5,6 the part of first-order logic.

1a) Is it true that if F and G are propositional formulas, then $F \models G$ or $F \models \neg G$? Prove it using only the formal definitions of propositional logic.
1b) Let F and G be propositional formulas such that $F \rightarrow G$ is satisfiable and F is satisfiable. Is it true that then G is satisfiable? Prove it using only the formal definitions of propositional logic.

2) Given a natural number n with $n > 1$, let F_n denote the propositional formula
$(p_{11} \wedge \ldots \wedge p_{1n}) \quad \vee \quad (p_{21} \wedge \ldots \wedge p_{2n}) \quad \vee \quad \ldots \quad \vee \quad (p_{n1} \wedge \ldots \wedge p_{nn})$.
2a) Write F_2 and write an equivalent formula in CNF without using any auxiliary variables. Do the same for F_3. Express how many clauses are needed in general for F_n, as a function of n.
2b) Write the Tseitin transformation of F_2. Is it logically equivalent to F_2? How many clauses are there in the Tseitin transformation of F_n, as a function of n?

3) Recently we got a visit from N students from the Ecole Normale Superieure de Cachan (Paris). First we had a session where each one of 9 research groups of our LSI department gave a short talk. After each that, each student $i \in 1..N$ selected a subset $\{i_1, i_2, i_3\} \subset \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with 3 of the 9 groups to receive a long talk from those 3 groups, for which there were 3 slots (named A, B and C) for the remainder of the day. Note that if a certain student chooses, for example, talks 3, 5 and 6, then these three long talks must be given in three different slots, but not necessarily in that order. Obviously, one possibility is to give all the long talks in all 3 the slots (27 talks in total), but here at most K talks, with $K < 27$ are allowed.
3a) Explain how to use a SAT solver for scheduling the talks. If you use any AMO, cardinality or pseudo-Boolean constraints, it is not necessary to convert these into CNF.
Hint: note that if a student chooses, for example, talks 3, 5 and 6, then it suffices to state that talks 3 and 5 are given in (at least) two different slots, and also 3 and 6 in different slots, and also 5 and 6. Furthermore, to state that two talks are given in (at least) two different slots out of three slots, it suffices to force that in any pair of slots at least one of the two talks is given.

3b) Express that no talk is given more than twice.

3c) How could you use the SAT solver to find the solution with the minimal total number K of talks?

4a) Is there any procedure that takes as input a formula F of first-order logic, and that always terminates saying "yes" if F is satisfiable, and that always terminates saying "no" if F is unsatisfiable? If so, briefly explain how it works.

4b) Is there any procedure that takes as input two formulas F and G of first-order logic, and that always terminates saying "yes" if $F \models G$, and that always terminates saying "no" or does not terminate if $F \not\models G$? If so, briefly explain how it works.

4c) Is there any procedure that takes as input a formula F of first-order logic, and that always terminates saying "yes" if F is satisfiable, and that always terminates saying "no" or does not terminate if F is unsatisfiable? If so, briefly explain how it works.

5a) Consider the first-order interpretation I where:
$D_I = \{0, 1, 2\}$, $a_I = 2$, and $P_I(n, m) = 1$ if and only if $m = (n + 1)$ *modulo* 3.
Let F be the formula $\forall x \, \exists y \, \exists z \, P(a, y) \wedge (P(y, z) \vee P(z, x))$. Do we have $I \models F$? Prove it.

5b) Let F be the first-order formula $\forall x \, \exists y \, \forall z \, P(x, y, z) \wedge Q(y)$, and let G be $\exists y \, \forall x \, \forall z \, Q(x) \wedge P(x, y, z)$. Are they logically equivalent? Is any one of the two a logical consequence of the other one? Prove it.

6) We have three dice (a die in Spanish is "dado", and the plural of die is dice). They are *fair* (each one of their six sides has the same probability of coming up) and their sides have numbers between 1 and 9 (not between 1 and 6!). Now suppose we play a game (many times): I pick a die; after that, you pick another die, we roll both dice, and the player who gets the highest number receives one Euro from the other player. Can you design the dice (putting the numbers on them) in such a way that you can become rich, that is, so that you can always pick a die that is *better* than mine (here *better* means that it wins with probability $p > 0.5$)? Write a Prolog program that checks whether this is possible or not. Include all non-predefined predicates you use.

Lógica en la Informática / Logic in Computer Science

Monday June 16th, 2014

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

1) We want to encode the at-most-one (AMO) constraint $x_1 + \ldots + x_{10} \leq 1$ using 24 clauses and 3 auxiliary variables a, b, and c. Write the smaller constraints into which the constraint is split and write the clauses for each part.

2) Consider an (undirected) graph: a set of vertices V and a set of edges E. We want to *color* it with K colors: each vertex of V gets exactly one of the K colors such that, for each edge (i, i'), vertices i and i' get different colors. Explain very briefly what is the best way to do this using SAT for different values of K, and why.

3A) Consider a number of N items, each item $i \in 1..N$ having a weight of w_i kg and a price of p_i euros. We also have a number of M trucks, each one of capacity 1000kg. Explain how to use a SAT solver to select a *subset* (not necessarily all) of the items that will be carried, such that the total price of all items carried is at least P euros. and to assign each selected item to some truck (respecting truck capacities). 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).

3B) How would you use a SAT solver to maximize the total price?

4) Prove using first-order logic that sentence 5 is a logical consequence of the first four:
1. All dogs make noise.
2. Anyone who has a cat has no mice.
3. Neurotic people do not have any noisy animals.
4. John has either a cat or a dog.
5. If John is neurotic he has no mice.

5) Consider an $n \times n$ chessboard (for any natural number $n > 3$, not necessarily 8), where n is defined by a Prolog clause `boardSize(`n`).` (for example, `boardSize(14)` if if $n = 14$). Define a Prolog predicate `horse(I1,J1,I2,J2)` that writes the shortest possible sequence of positions that a horse of chess traverses to go from initial position `I1,J1` to final position `I2,J2` on the board (positions are (row,column), each one in $1..n$). It must write the sequence in the right order, the first position being `[I1,J1]`, and write "no solution" if no such a sequence exists.

Lógica en la Informática / Logic in Computer Science

Tuesday January 13th, 2015

Time: 2h30min. No books, lecture notes or formula sheets allowed.
The propositional logic part comprises questions 1 to 4.

1) Let F be a propositional formula. Is it true that F is a tautology if and only if $\neg F$ is unsatisfiable? Prove it using only the definition of propositional logic.

2) How would you efficiently decide whether a given propositional formula in DNF is satisfiable?

3) We want to solve a certain problem with a SAT solver. It contains a number of at-most-one constraints of the form $x_1 + \ldots + x_n \leq 1$, each one of which has to be encoded (expressed as a set of clauses). A good property of such an encoding is that, as soon as one of the variables x_i becomes true, the unit propagation mechanism of the SAT solver will set to false all the other variables $x_1 \ldots x_{i-1}, x_{i+1} \ldots x_n$. Does the Heule-3 encoding have this property? Briefly explain why.

4) The Seat car manufacturer has a website where one can order a car, *configuring* all its (hundreds of) *properties*: which type of engine you want, which color, type of music equipment, etc. The configuration software has to take into account that there exist constraints relating pairs of properties, e.g., "this type of air conditioning cannot go together with that kind of diesel engine". In general such constraints have the form: "property P_i cannot go together with property P_j", or "if property P_i is true then also property P_j must be true", or "at least one of property P_i and property P_j must be true". The software should give a warning if the user tries to make a configuration that is forbidden with respect to these constraints.

4a) Explain how you would do that using propositional logic (which variables and clauses you would introduce, etc.).
4b) How would you handle constraints relating more than two properties, such as "the user must choose at least one of the three properties P_i, P_j or P_k"?

5) Consider two groups of 10 people each. In the first group, as expected, the percentage of people with lung cancer among smokers is higher than among non-smokers. In the second group, the same is the case. But if we consider the 20 people of the two groups together, then the situation is the opposite: the proportion of people with lung cancer is higher among non-smokers than among smokers! Can this be true? Write a little Prolog program to find it out.

6) Let F be the first-order formula $\exists x \forall y \exists z\ p(z, y) \wedge \neg p(x, y)$.
6a) Give a model I with $D_I = \{a, b, c\}$.
6b) Is it true that $F \models \forall x\ p(x, x)$?
6c) Is there any model of F with a single-element domain?

7) In a certain village, the barber shaves all men that do not shave themselves, and only these men. Formalize this sentence in first-order logic and prove by resolution that the barber is a woman (i.e., not a man).

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 $I: \mathcal{P} \to \{0, 1\}$ 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?

1b) How many models does $F \wedge p_1$ have at most? and at least?

1c) How many models does $F \vee p_1$ have at most? and at least?

1d) How many models does $F \vee G$ have at most? and at least?

1e) How many models does $F \wedge G$ have at most? and at least?

1f) If $F \models G$, how many models does $F \vee G$ have at most? and at least?

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.

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

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?

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.

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.

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).

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.

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

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

Lógica en la Informática / Logic in Computer Science

Tuesday January 12, 2016

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) Prove using only the definition of propositional logic, that the $\vee$ connective is associative, that is, if F, G, H are formulas, then $(F \vee G) \vee H \equiv F \vee (G \vee H)$.

2) Consider a *car configuration problem* where a customer who buys a car can choose to install a subset C of a set $S = \{1..n\}$ of *features* (type of engine, type of seats, color, etc., etc.). But there are many constraints, of two types. A constraint i of the first type has the form: "if all features of this subset S_i of S are installed, then also this additional feature f_i of S must be installed". A constraint i of the second type says that "not all features of the subset S_i of S can be installed at the same time". Do you see any efficient algorithm based on SAT for deciding, given all constraints and the set C, whether the customer can choose its given subset C of features? If so, which one?

3) A large group of N people will attend a conference center during one day from 10:00 to 22:00h. During the day, M meetings have to be organized among different subsets of the N people. Each meeting i in $1..M$ is given by the subset $S_i \subseteq \{1..N\}$ and its duration of d_i consecutive hours. There are K_1 meeting rooms of size 25 available, and K_2 of size 50. No room can host more than one meeting at the same time, and of course no person can attend more than one meeting at the same time. There is also a list of *blockings* of the form (j, h), indicating that person j, with j in $1..N$, is not available for any meeting at hour h, with h in 10..21.

Explain in detail how to use a SAT solver for deciding, if possible, exactly when each meeting takes place and in which meeting room. 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 eficient and simple as possible.

4) The following prolog program solves exercise 3. Finish it implementing the predicates `room`, `attendantsOverlap`, `roomsOverlap`, and `blockingProblem`.

```
initialHour(10).
finalHour(22).
meeting( 1, [7,28,180,235], 3). % meeting 1: these four  people, during 3 hours
meeting( 2, [6,7,8],        2). % meeting 2: these three people, during 2 hours
...                             % more clauses for meetings
blocking(28,17).                % person 28 cannot attend meetings at 17 o'clock
...                             % more clauses for blockings
numSmallRooms(8).               % rooms 1-8
numLargeRooms(5).               % rooms 9-13

solution:- findall([N,S,D], meeting(N,S,D), L), schedule(L,Sol), write(Sol), nl.

schedule( [], [] ).
schedule( [[N,S,D]|L], [[N,S,D,Hour,Room]|Sched] ):-
    schedule(L,Sched),
    initialHour(IH), finalHour(FH),  FH1 is FH-D,  between(IH,FH1,Hour),
    \+blockingProblem(Hour,S,D),          % no blocking problem
    length(S,Num), room(Num,Room),        % Room is a adequate for Num people
    \+roomsOverlap(D,Hour,Room, Sched),   % no room overlapping problem
    \+attendantsOverlap(S,D,Hour, Sched). % no attendants overlapping problem

% Sched contains some overlapping meeting in the same room:
roomsOverlap( D,Hour,Room, Sched ):- member( [_,_,D2,Hour2,Room], Sched ), ...

% some attendant is blocked during this period:
blockingProblem(Hour,S,D):- ...

% some attendant has another meeting overlapping with this one
attendantsOverlap(S,D,Hour, Sched):- ...
```

5) Let F be a (closed) first-order formula and let I be a given first-order interpretation for the symbols occurring in F. For each one of the following cases, is it decidable whether $I \models F$?
5a) When D_I is a finite set.
5b) When D_I is the integers, and the symbols of F are interpreted in I as well-known operations on the integers, such as functions $+$ or $*$, and predicates $>$ and $=$.
5c) As in 5b) but where moreover F is a set of Horn clauses.

6)
6a) Explain in a few words how you would formally prove, given two first-order formulas F and G, that $F \not\models G$.
6b) Same question for $F \models G$.
6c) 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.
6d) 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.

Lógica en la Informática / Logic in Computer Science

Monday June 13, 2016

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).

1a) Let F and G be propositional formulas such that F is a tautology. Is it true that $F \wedge G \equiv G$? Prove it using only the definitions of propositional logic.
1b) Let F and G be propositional formulas such that F is satisfiable and $F \rightarrow G$ is also satisfiable. Is it true that G is satisfiable? Prove it using only the definitions of propositional logic.

2) Let us remember the well-known graph coloring problem. **Input:** a natural number k, and an (undirected) graph with n vertices and m edges of the form $(u_1, v_1) \ldots (u_m, v_m)$, with all u_i and v_i in $\{1 \ldots n\}$, and **Question:** is there a way to "color" each vertex with a color (a number) in $1 \ldots k$ such that adyacent vertices get different colors?

We know that graph coloring is NP-complete in general. But what is its complexity if $k = 2$? Explain why using sat-based arguments.

3) Let S be a satisfiable set of propositional Horn clauses.
3a) What is the complexity of finding the *minimal* model of S, that is, the model I with the minimal number of symbols p such that $I(p) = 1$?
3b) What is the complexity of deciding whether S has only one model or more than one?
For both questions, explain very, very, briefly why.

4) We want to write a computer program that takes as input two arbitrary first-order formulas F and G and always terminates writing "yes" if $F \equiv G$, and "no" otherwise. Explain very shortly the steps you would follow to do this, or to get something as similar as possible.

5) Formalize and prove by resolution that sentence E is a logical consequence of the other four.
A: If a person likes logic, he does not like football.
B: Brothers of football players like football.
C: Messi is a football player and Ney is his brother.
D: Ney likes logic.
E: Our teacher is a nice guy who knows a lot about football and logic.

6) Complete the following graph coloring program (see problem 2). Do `makeConstraints` recursively, using `#\=` and the built-in predicate `nth1(I,L,X)` ("the `I`th element of the list `L` is `X`").

```
:- use_module(library(clpfd)).
numVertices(5).
edges([ 1-2, 1-3, 2-3, 2-4, 2-5, 3-5 ]).
numColors(3).

main:- numVertices(N),edges(Edges), listOfNPrologVars(N,Vars), ...
       Vars ins ...
       makeConstraints(Edges,Vars),
       ...
       write(Vars), nl.

makeConstraints(...

listOfNPrologVars(...
```

Lógica en la Informática / Logic in Computer Science

Thursday January 12th, 2017

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

Note on evaluation:
eval(propositional logic) = max{ eval(Problems 1,2,3,4), eval(partial exam) }.
eval(first-order logic) = eval(Problems 5,6,7).

1) Let F and G be arbitrary propositional formulas.

1a) Can it happen that $F \models G$ and $F \models \neg G$? Prove it from the definitions of propositional logic.

1b) Assume $F \models G$. Prove, from the definitions of propositional logic, that then $F \equiv F \wedge G$.
Hint: prove $F \equiv F \wedge G$ distinguishing the two cases: $I \models F$ and $I \not\models F$ (and use $F \models G$ in one case).

2) The Tseitin transformation T transforms an arbitrary propositional formula F into a CNF (a set of clauses with auxiliary variables) $T(F)$ that is *equisatisfiable*: F is SAT iff $T(F)$ is SAT. Moreover, the size of $T(F)$ is linear in the size of F. Answer **very briefly:** Is there any known transformation T' into an equisatisfiable linear-size DNF? If yes, which one? If not, why?

3) Answer **very briefly:** What is 2-SAT? Is it polynomial? Why?

4) Answer **very briefly:** 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 $a_1x_1 + \ldots + a_nx_n \leq k$?

5) Is the following first-order formula satisfiable? If not, explain why. If yes, give a model (and no explanations).
$\forall x \ \neg p(x,x) \ \wedge$
$\forall x \, \forall y \, \forall z \ (\ p(x,y) \wedge p(y,z) \rightarrow p(x,z) \ ) \ \wedge$
$\forall x \, \exists y \ p(x,y) \ \wedge$
$\forall x \, \forall y \ (\ p(x,y) \rightarrow \exists z \, (p(x,z) \wedge p(z,y)) \ )$

6) **6a)** Explain in a few words how you would formally prove, given two first-order formulas F and G, that $F \not\models G$.
6b) Same question for $F \models G$.
6c) 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.
6d) 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.

7) Write a program `return(L,A)` in swi prolog (using `library(clpfd)` or not, feel free) that outputs (writes) the *minimal* number of coins needed for returning the amount `A` if (infinitely many) coins of each value of the list `L` are available. Two examples:
`?-return([1,5,6],10).` writes `2` (since 2*5 = 10).
`?-return([1,2,5,13,17,35,157],361).` writes `5` (since 1*13 + 2*17 + 2*157 = 361).

Lgica en la Informtica / Logic in Computer Science
June 20th, 2017. 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 Consider the at-most-one (AMO) constraint, expressing that at most one of the propositional variables $x_1 \dots x_n$ is true, also written $x_1 + \cdots + x_n \leq 1$. Consider:

1) the encoding for AMO you know that needs the smallest (in terms of n) number of clauses, and
2) the encoding that needs the smallest number of auxiliary variables.

For each case, write **giving no further explanations**: a) the name of the encoding, b) which, and how many, auxiliary variables it uses, c) which, and how many, clauses (always expressing how many in terms of n).

2 My friend John says that he has found a new way to speed up SAT solving. Before starting his SAT solver, he removes from the set of clauses S some clauses he calls "unnecessary":

A: if there is some variable x that appears only in positive literals of clauses of S, then he removes from S all clauses containing x

B: similarly, if some variable y appears in S only in negative literals then he removes from S all clauses containing y.

Note that after eliminating some "unnecessary" clauses, step A or B may be (or become) applicable for other variables, so John continues doing this until no more variables of type A or B exist and then launches his solver on a (hopefully) much smaller set of clauses. Is John's idea correct? Explain why, **in very few words**.

3A: What is the complexity of 2-SAT? (just answer, no explanations needed).

3B: Any set of propositional *positive clauses*, that is, clauses with only positive literals (no negations), is of course satisfiable, because the interpretation making all variables true is a model. What is the complexity of deciding the satisfiablity of a given "2-or-positive" set of clauses S, that is, such that every clause in S is either positive or two-literal (or both)? Explain why, **in very few words**.

Hint: with two-literal clauses we can express that one variable is the negation of another variable.

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

```
brother(joan,pere).
father(enric,joan).
uncle(N,U):- father(N,F), brother(F,U).

?- uncle(X,Y).
X = enric,
Y = pere.
```

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

5: For each statement, say whether it is true or false and show why **in an as simple and short as possible way**:

5A: The formula $\forall x \, \exists y \; (p(x, f(y)) \wedge \neg p(x, y))$ is satisfiable.

5B: $\forall x \, \forall y \, \exists z \; q(x, z, y) \models \forall x \, \exists z \, \forall y \; q(x, z, y)$.

6: My good old friend John says that he has written a C++ program P that takes as input an arbitrary first-order formula F, and such that, if F is a tautology, P always outputs "yes" after a finite amount of time, and if F is not a tautology, P outputs "no" or it does not terminate.

Is this possible? If this is not possible, explain why. If it is possible, explain how P would work. **A very short answer suffices.**

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

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.

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.

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:

$$\neg v \vee \neg u \qquad \neg v \vee \neg z \qquad \neg v \vee \neg x \vee \neg y \qquad \neg u \vee \neg z \vee \neg y \qquad \neg u \vee \neg z \vee \neg x$$

Write the minimal set of clauses for $2x + 3y + 5z + 6u + 8v \geq 11$ (give no explanations).

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.

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.

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 $\exists u\, \exists v\, \mathit{better}(u, v)$ 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.**