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 $(F \vee G) \vee H \equiv F \vee (G \vee H)$.

Answer: $(F \vee G) \vee H \equiv F \vee (G \vee H)$
iff [by definition of logical equivalence $\equiv$]
$(F \vee G) \vee H$ and $F \vee (G \vee H)$ have the same models
iff [by definition of model]
for all I, $eval_I((F \vee G) \vee H) = eval_I(F \vee (G \vee H))$
iff [by definition of eval of $\vee$]
for all I, $max(eval_I(F \vee G), eval_I(H)) = max(eval_I(F), eval_I(G \vee H))$
iff [by definition of eval of $\vee$]
for all I, $max(max(eval_I(F), eval_I(G)), eval_I(H)) = max(eval_I(F), max(eval_I(G), eval_I(H)))$
iff [by definition of max]
for all I, $max(eval_I(F), eval_I(G), eval_I(H)) = max(eval_I(F), eval_I(G), eval_I(H))$.

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?

Answer: Consider n propositional variables x_i, for $i \in \{1..n\}$, meaning "feature i is installed". For each feature i in C there will be a unit clause x_i. The constraints of the first type are Horn clauses with one positive literal, and the contraints of the second type are Horn clauses with zero positive literals. If there is a model, it indicates which features to install. This is a Horn-SAT problem, that is solvable in linear time.

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 efficient and simple as possible.

Answer:

Variables:

$s_{i,h}$ means: "meeting i starts at hour h", for $1 \le i \le M$ and $10 \le h \le 21$
$r_{i,k}$ means: "meeting i takes place at room k", for $1 \le i \le M$ and $1 \le k \le K_1 + K_2$

Clauses:

-Each meeting i starts at some hour that allows it to finish before 22h:
One clause $s_{i,10} \vee \ldots \vee s_{i,22-d_i}$ for each i in $1..M$.

-Each meeting i takes place in some (at least one) room. For each i in $1..M$,
If $|S_i| \le 25$, one clause $r_{i,1} \vee \ldots \vee r_{i,K_1+K_2}$
If $|S_i| > 25$, one clause $r_{i,K_1+1} \vee \ldots \vee r_{i,K_1+K_2}$
Note: if $|S_i| > 50$ we do not need to call the SAT solver: the problem is unsatisfiable.

-No meeting i violates any of it's attendants blockings:
One clause $\neg s_{i,h}$
for $1 \le i \le M$ and $10 \le h \le 21$ if there is some blocking (j, h') with $j \in S_i$ and such that h' is among the meeting's hours, i.e., $h \le h' < h + d_i$.

-No two meetings i and i' overlap in time if they share some attendant:
One clause $\neg s_{i,h} \vee \neg s_{i',h'}$
for $1 \le i \le M,\ 1 \le i' \le M$ such that $S_i \cap S_{i'} \neq \emptyset$, and hours $10 \le h \le h' \le 21$ such that the two meetings overlap, i.e., $h' < h + d_i$.

-No two meetings i and i' overlap in time if they share the same room k:
One clause $\neg s_{i,h} \vee \neg s_{i',h'} \vee \neg r_{i,k} \vee \neg r_{i',k}$
for $1 \le i \le M,\ 1 \le i' \le M$ and $1 \le k \le K_1 + K_2$ and hours $10 \le h \le h' \le 21$ such that the two meetings overlap, i.e., $h' < h + d_i$.

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):- ...
```

Answer:

```
room(N,R):-N<26,!,numSmallRooms(S),numLargeRooms(L), T is S+L,           between(1,T,R).
room(N,R):-N<51,  numSmallRooms(S),numLargeRooms(L), T is S+L, K is S+1, between(K,T,R).

blockingProblem(H,S,D):- End is H+D-1, member(J,S), blocking(J,X), between(H,End,X).

attendantsOverlap(S,D,H, Sched):- member([_,S2,D2,H2,_],Sched), member(J,S),member(J,S2),
          End is  H+D-1, End2 is H2+D2-1, between(H,End,X), between(H2,End2,X).

roomsOverlap(     D,H,R, Sched):- member([_, _,D2,H2,R],Sched),
          End is  H+D-1, End2 is H2+D2-1, between(H,End,X), between(H2,End2,X).
```

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.

Answer:
5a: yes. Only a finite number of cases have to be checked to compute $eval_I(F)$ in the standard way.

5b: no. It is well known to be decidable whether a given quadratic equation with just one variable (such as $3x^2+4x+5 = 0$) has any integer solution. However, this is undecidable in general for equations $P = 0$ if P can be an arbitrary polynomial with products between variables, such as $x^3y + 2\,z^5 + \ldots$. But it would be decidable if one could evaluate arbitrary formulas in the integers!

This is because we can easily express any polynomial P by a first-order term T_P, for example, $sum(prod(prod(x, prod(x, x)), y), \ldots)$, such that

$P = 0$ has no integer solution **iff** $I \models \forall x, y, z ... \neg eq(T_p, zero)$

where D_I are the integers and eq_I, $zero_I$, sum_I, $prod_I$ are the functions $=$, 0, $+$, and $*$.

5c: F being a set of Horn clauses does not help: in the above example for 5b, $\forall x, y, z ... \neg eq(T_p, zero)$ is a unit (that is, Horn) clause.

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.

Answer:

6a: Giving a counter example, an interpretation I such that $I \models F$ but $I \not\models G$.

6b: By proving that $F \wedge \neg G$ is unsatisfiable, turning it into clausal form S, and obtaining the empty clause from S by resolution and factoring.

6c: Yes. $F \models G$. We prove it as in 6b. Here F gives two clauses: $1:\ p(a, x)$ and $2:\ \neg q(c_y)$ (here c_y is the Skolem constant introduced for y).
$\neg G$ is $\neg(\exists v \exists w\, \neg q(w) \wedge p(v, a))$ which becomes $\forall v \forall w\, \neg(\neg q(w) \wedge p(v, a))$ which becomes $\forall v \forall w\, q(w) \vee \neg p(v, a)$ which becomes the clause $3:\ q(w) \vee \neg p(v, a)$.
By one step of resolution between 1 and 3 with mgu $\{x = a, v = a\}$ we get clause $4 : q(w)$, and with one more step of resolution between 2 and 4 with mgu $\{w = c_y\}$ we get the empty clause.

6d: No. $F \not\models G$. We prove it as in 6a. Consider the interpretation I where $D_I = \{a, b\}$ and p_I is interpreted as equality on this domain. Then $I \models F$ but $I \not\models G$.