Lógica en la Informática / Logic in Computer Science

Thursday April 12th, 2012

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

SOLUTIONS

1) Which is the minimal known algorithmic cost of deciding the satisfiability of the following types of propositional formulas:

- **a:** Arbitrary propositional formulas: **Answer:** exponential (it is NP-complete)
- **b:** 3-SAT: **Answer:** exponential (it is NP-complete)
- **c:** 2-SAT: **Answer:** linear
- **d:** DNF; **Answer:** linear
- **e:** CNF; **Answer:** exponential (it is NP-complete)
- **f:** Horn-SAT: **Answer:** linear

2) If you have a SAT solver that can only handle clauses with at most 3 literals, what would you do to decide the satisfiability of arbitrary propositional formulas?

Answer: Apply Tseitin's transformation, which gives only clauses of lenght at most 3. Let F be an arbitrary formula. Assume the nodes of F are numbered. Tseitin's transformation introduces a new variable z_i for each non-leaf node i of F. If node i is of the form $and(x, y)$, we generate three clauses: $\neg z_i \vee x$, $\neg z_i \vee y$, and $z_i \vee \neg x \vee \neg y$. If it is of the form $or(x, y)$, we generate three clauses: $z_i \vee \neg x$, $z_i \vee \neg y$, and $\neg z_i \vee x \vee y$. If the node is of the form $not(x)$, we generate the two clauses $z_i \vee \neg x$, and $\neg z_i \vee x$ (or simply replace all occurrences of z_i by $\neg x$, introducing no variable z_i for this node). Let S be the resulting set of clauses of this Tseitin transformation of F. Assume the root variable of F is z_1. Then F is satisfiable if and only if $S \cup \{z_1\}$ is satisfiable.

3) The new telecom company TC wants every person in Spain to have a TC shop in his own city or in some other city at less than 50 Km distance. But TC wants to do this with no more than K shops. Of ourse, they have a map where they can see, for every city, which other cities there are at less than 50Km (if any). How would you use the Barcelogic SAT solver to determine in which cities they should open a TC shop?

Answer: Assume the N Spanish cities are numbered from 1 to N. Introduce N propositional variables s_i meaning that "city i gets a shop". We need to encode two properties: A) No more than K of the N propositional variables $s_1 \dots s_N$ are true. This can be done using any cardinality constraint encoding of $s_1 + \dots + s_N \leq K$, for example using sorting networks. B) For each city i, we also need to ensure that there is a shop in i itself or in some other city at less than 50 Km distance from i: if $i_1, \dots, i_p$ are the cities at less than 50Km of i, we add the clause $s_i \vee s_{i_1} \vee \dots \vee s_{i_p}$. This amounts to one such clause per city i. Note that if there are no cities at less than 50 Km from i, this is a unit clause: then there *must* be a shop in i.