Instructor: Edmund M. Clarke TAs: Soonho Kong, David Henriques

Due date: 09/21/2011 `cmu15414ta@gmail.com`

Assignment 2

Problem 1

Consider a formula $\phi = \omega_1 \wedge \cdots \wedge \omega_6$ where

$$\begin{aligned}
\omega_1 &= (x_1 \vee x_3 \vee x_4)\\
\omega_2 &= (x_1 \vee x_3 \vee \neg x_4)\\
\omega_3 &= (x_1 \vee \neg x_3 \vee x_4)\\
\omega_4 &= (x_1 \vee \neg x_3 \vee \neg x_4)\\
\omega_5 &= (\neg x_1 \vee x_2)\\
\omega_6 &= (\neg x_1 \vee \neg x_2)
\end{aligned}$$

Part A. The result of *resolving* a clause $(x_1 \vee \cdots \vee x_n \vee r)$ with another clause $(\neg r \vee y_1 \vee \cdots \vee y_m)$ on r is the disjunction of the literals in the set $\{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_m\}$. Note that the resulting clause is logically implied by the conjunction of the two original clauses.

1. Let ω_{10} be the result of resolving ω_1 with ω_2 on x_4. What is ω_{10}?
2. Let ω_{11} be the result of resolving ω_3 with ω_4 on x_4. What is ω_{11}?
3. (a) On what variable can ω_{10} and ω_{11} be resolved?
   (b) Let ω_{12} be the result of this resolution. What is ω_{12}?
4. Let ω_{13} be the result of resolving ω_5 and ω_6 on an appropriate literal. What is ω_{13}?
5. Let ω_{14} be the result of resolving ω_{12} and ω_{13} on an appropriate literal. What is ω_{14}?
6. What does the above tell you about the satisfiability of ϕ? Briefly explain in one or two sentences.

Part B. Consider a SAT solver as described in the GRASP lecture slides, in particular the slides titled "Top-level of GRASP-like solver" and "Learning Algorithm". Suppose that the decision heuristic is to pick the lowest-numbered unassigned variable and assign it the value `false`. Consider executing this SAT solver on the formula $\phi = \omega_1 \wedge \cdots \wedge \omega_6$. For every conflict encountered, give:

1. the implication graph (using the format given on the slide titled "Implication Graphs"),
2. the initial conflict assignment (from Step 1 on the "Learning Algorithm" slide),
3. the final conflict assignment (from Step 4 of the slide),
4. the learned clause, and
5. the decision level to which to backtrack (if the learned clause is non-empty).

Include decision levels in your conflict assignments; use the format "$\{x_5{=}1@6,\ x_6{=}1@6\}$".

Problem 2: Pigeonhole Problem

Consider the problem of placing n pigeons in m pigeonholes. If $n > m$, then at least one pigeonhole must contain more than one pigeon.

Given n pigeons and m pigeonholes, we wish to write a CNF formula ϕ such that ϕ is satisfiable iff each pigeon can be put in some pigeonhole and each pigeonhole has at most one pigeon. It turns out that a direct encoding of this pigeonhole problem is quite difficult for SAT solvers when $n > m$.

Let $z_{p,h}$ be a propositional variable which is true when pigeon p is placed in hole h.

Notation: If S is a set of clauses, we'll write "$\bigwedge S$" to denote the conjunction of these clauses. For example, $(\bigwedge\{C_1, C_2, C_3\}) = (C_1 \wedge C_2 \wedge C_3)$.

Let E be a conjunction of clauses which encode the requirement that each pigeon must be in some hole, defined as follows:

$$\begin{aligned} E &= \bigwedge \left\{ (z_{p,1} \vee z_{p,2} \vee \ldots \vee z_{p,m}) \;\middle|\; 1 \leq p \leq n \right\} \\ &= (z_{1,1} \vee z_{1,2} \vee \ldots \vee z_{1,m}) \quad \wedge \\ &\quad\; (z_{2,1} \vee z_{2,2} \vee \ldots \vee z_{2,m}) \quad \wedge \\ &\qquad\qquad \cdots \qquad\qquad\quad \wedge \\ &\quad\; (z_{n,1} \vee z_{n,2} \vee \ldots \vee z_{n,m}) \end{aligned}$$

Let H_h be a conjunction of clauses which encode the requirement that hole h cannot have more than one pigeon, defined as follows:

$$H_h = \bigwedge \left\{ (\neg z_{i,h} \vee \neg z_{j,h}) \;\middle|\; 1 \leq i < j \leq n \right\}$$

The final formula is

$$\phi = E \wedge H_1 \wedge H_2 \wedge \ldots \wedge H_m$$

To give this problem to a SAT solver that needs integer-numbered variables, we renumber as follows: $z_{p,h}$ becomes x_{p*m+h}.

Consider the SAT solver given on Slide 13 ("DPLL Solver") of the Lecture 2 slides. Assume the decision heuristic is as follows: Pick the first literal in the first unresolved clause and assign it `true`.

1. For the case of 4 pigeons and 3 holes, draw the decision tree followed by the SAT solver. (Only include decision literals, not forced literals.) A conflict should be indicated by a special leaf node, as in Slide 21. (You may use a simple star to indicate a conflict leaf node.)
2. Given m pigeonholes and $m + 1$ pigeons, how many conflicts would occur, as a function of m? (Each time the line "`if [] in ClauseList: return False`" returns `false` counts as one conflict.) Explain your reasoning.

Problem 3: MiniSAT

Most fast SAT solvers use the `DIMACS` input format. A CNF formula $\phi = C_1 \wedge \ldots \wedge C_m$, where $C_i = \ell_{i,1} \wedge \ldots \wedge \ell_{i,n_i}$, is encoded in DIMACS as follows:

$$
\begin{array}{l}
\texttt{p cnf } \textit{NumVars NumClauses} \\
\ell_{1,1}\ \ell_{1,2}\ \ldots\ \ell_{1,n_1}\ \texttt{0} \\
\ell_{2,1}\ \ell_{2,2}\ \ldots\ \ell_{2,n_2}\ \texttt{0} \\
\ldots \\
\ell_{m,1}\ \ell_{m,2}\ \ldots\ \ell_{m,n_m}\ \texttt{0}
\end{array}
$$

Each clause is described in a line terminated by a zero. As an example, the formula
$(x_1) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_4 \vee \neg x_1) \wedge (\neg x_1 \vee \neg x_2 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee x_4)$ would be encoded as:

```
p cnf 4 5
1 0
2 -3 0
-4 -1 0
-1 -2 3 4 0
-2 4 0
```

For this assignment we will use the `MiniSat` SAT solver, which is one of the fastest SAT solvers currently available.

You can find a binary executable of the `MiniSat` solver for Linux in

`/afs/andrew.cmu.edu/usr16/wklieber/public/`

You can find DIMACS files for the pigeonhole problem at

`http://www.cl.cam.ac.uk/~tw333/software/pigeonhole/`

Run MiniSAT on pigeon-3.cnf through pigeon-8.cnf (inclusive) and report the number of conflicts for each file. (The numbers will not be exactly the same as the numbers given by the formula in your answer to Problem 2, but they shouldn't differ by more than a factor of 2.)