COMP2008 Coursework: Implementing a simple DPLL solver

Juan Antonio Navarro Pérez

Due date: December 7, 2012

Your task in this coursework is to implement a simple DPLL solver that takes as input a formula in cnf (conjunctive normal form), and returns either 'satisfiable' or 'unsatisfiable' depending on whether the input formula is or not satisfiable.

You are free to choose the programming language you prefer, and make use of its standard libraries (but nothing else). I recommend Python, Java, or C++. If you wish to use something else, let me know first. Your program should read from the standard input, and write to the standard output so that, e.g., if `sample.cnf` is an input file then you can call it using, e.g.,

```
$ python dpll.py < sample.cnf
satisfiable
```

The input will be given as several lines of text, the first one of the form

```
p cnf n m
```

where n (≤ 20) is the number of distinct propositions $p_1, \ldots, p_n$ that may be used in the formula, and m (≤ 100) is the number of clauses in the formula. The remaining of the input contains m clauses, one by one on each line. For each clause we list the literals that it contains, but instead of writing p_5 or $\neg p_7$, we will simply use the integer numbers 5 or -7. Each clause is terminated with the number 0. This is known as 'DIMACS' format. For example, the cnf formula

$$(\neg p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2 \vee p_3) \wedge p_1 \wedge \neg p_3$$

would be represented in DIMACS as follows

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

Your submission will be run against 10 different examples. You get one point, out of a total of 10, for each correct answer.