Case Study 1: The Satisfiability Problem for Proprositional Logic

Due Date Upload your report on or before 5:00PM, 2/23/2015. 24 hour rules apply.

Motivation One of the key computational problem for propositional logic is the *satisfiability problem* (abbrev. as `SAT`). Formally, it is the decision problem to test whether a given propositional formula ϕ is *satisfiable*. Note that a propositional formula ϕ is tautology (i.e. $\models \phi$) if and only if the negation of the formula, $\neg\phi$, is not satisfiable.

For those who are familiar with the notion of NP completeness, we recall that the Cook-Levin theorem. It states that the decision problem `SAT` is NP-complete. This means that if there is a polynomial time algorithm for solving `SAT`, then there are polynomial time algorithms for solving many well known hard computational problems such as the travelling saleman's problem etc. (these problems are in the class NP). Hence, when a problem is NP complete, it is "unlikely" that we can find a polynomial time algorithm to solve that problem. While the satisfiability problem for propositional formula is NP complete, the satisfiability problem for subclasses of propositional formulas may not. In this case study, you will gain an experience to solve a number of satisfiability problem via a satisfiabilty solver.

Introduction

The software package `SAT4J` is a *satisfiability solver* which are designed to solve the satisfiability problem. It can also be used as standalone problem or as a JAVA library. It can also work together with other software tools such as the Alloy program analyzer (we will explore this software tool at the latter part of the semester).

Before starting this case study, please read the following documents carefully:

1. Getting started with SAT4J, (Nov 15, 2010 Version)
2. Satisfiability Suggested Format (May 8, 1993 Version)
3. HR Chapter 1.

The installation of SAT4J and a short tutorial will be given by our TA in class. Now, in this part of the homework, you are asked to use SAT4J in your machine, either as a standalone SAT solver or as a JAVA library to complete the following tasks:

Task 1 (40 point) Verifying Horn Formulas

Horn Formulas are defined in HR, Definition 1.46. For each of the Horn formulas given in HR Exercise 1.5, Question 15 part b) to part f), Explain how you will use the `SAT4J` to verify if they are satisfiable. Provide the answers and document all the key steps.

Task 2 (60 point) Solve the four queens' problem

The problem can be phrased as follows:

> *Four Queens' Problem*: Is it possible to place four chess queens on an 4×4 chessboard so that no two queens attack each other.

To complete this task, first show how you are going to formulate the four queens' problem via propositional logic. Describe how you can make use of `SAT4J` to solve the problem. Document all the steps you use in your report (For example, you may include the screen shots while running SAT4J for specific propositional formulas you formulated.) Comment on the feasibilty of using your solution to solve the general N queens problem where N is a large integer.

Submission Please submit an electronic copy in word or pdf format via blackboard. Further details regarding submission will be given in blackboard.