CS402 (By Aashish Waikar-20196055)

Implementation of DPLL based SAT-solver

Introduction

In this project, I have implemented a DPLL based SAT solver with some optimisations. The underlying algorithm is the following:-

Input: CNF formula F .

1)Initialise A to be the empty list of assignments.

2)While there is a unit clause {L} in F | A , add assignment L 7→ 1 to A.

3)If F | A contains no clauses then stop and output A.

4)If F | A contains the empty clause then apply the learning procedure to add a new clause C to F . If C is the empty clause then stop and output "UNSAT". Otherwise backtrack to the highest level at which C is a unit clause. Go to Line 2.

5) Apply the decision strategy to determine a new decision assignment P 7→ b to be added to A. Go to Line 2.

Basic Implementation

The platform I'm using for this project is python2.7. I have implemented many functions with different purposes. I've designed a function which takes input in the DIMACS format and converts it to a list of clauses. Then the parent DPLL function is called which has UnitPropagation and Backtracking as its helper functions.

Optimisations and Heuristics for variable ordering

I have used the *JW(Jeroslow Wang)* heuristic for variable ordering and have implemented clause learning. Initially, I had many options of the heuristic which I could implement. I started with *MOM* heuristic but then I read some articles online which stated better performance of JW. So, I changed the heuristic I used to JW and it improved the performance significantly.