SAT-SOLVER

(Using Semantic-tableau)

Team Members:

- Gavish Garg (200385)
- Bhavya Garg (200270)

1. Introduction

This is a brief introduction of the SAT-SOLVER implementation using python.
A SAT-SOLVER basically takes input of clauses in CNF form and returns whether the given set of clauses are **SAT**isfiable or not.

A CNF file format is **an ASCII file format**. ... This begins with a lowercase "p" followed by a space, followed by the problem type, which for CNF files is "cnf", followed by the number of variables followed by the number of clauses. The remainder of the file contains lines defining the clauses, one by one.

Example :

```
c  simple_v3_c2.cnf
c
p cnf 3 2
1 -3 0
2 3 -1 0
```

- Each clause is a **disjunction** of Literals
- The given formula is a **conjunction** of these literals.

For the model to be **SAT**isfiable, all the clauses in the given CNF file should be TRUE.

2. Code

1. Taking the input of the clauses in the CNF file.
2. Using a recursive approach to find a MODEL for the given formula.

WORKING

1. We use a Linked List to store all the clauses. Each clause in the linked list is also a List itself.
2. We use a recursive approach to find a **MODEL** for the given formula.
3. Our program builds a list "**L**" which adds a literal from a particular clause and then adds another literal from the next clause. However if it causes two **CONTRADICTORY** literals to be present simultaneously, then it checks for another literal from the same clause.
4. And similarly checks for a literal in the next clause.

LOGIC

1. The list "L" tries to add a literal from each clause, which implies that if it can add one literal from each clause then the given formula is SATisfiable.
2. And one of the **MODEL**s for the satisfiability is the List "**L**".
3. If there is no such list possible then it returns **UNSAT**.

3. Assumptions

We are assuming that the given list of clauses is not empty. If it was empty then the SAT solver should have returned a Model. And there can be an infinite number of models in this case as the given set of clauses in Empty.

4. Limitations

1. The above approach is not time efficient.
2. If the number of variables or the number of the clauses increases , time taken for solving is too much.