CS4244 Project1 Design

Structure

Our design comprises five parts:

1. **cnf**
   In this part, the data structure used in the solver is built from bottom to top. This package includes:

   - **Literal**: The basic unit which has three properties:
     - *symbol*: the proposition wrapped in this literal.
     - *assignment*: the assignment of the proposition.
     - *negation*: whether this proposition is negative.
   - **Clause**: A set of literals.
   - **CNF**: A set of clauses.
   - **CNFConstructorString**: The constructor used to build a CNF from input of a customary format. We use it in debugging.
   - **CNFConstructorDIMACS**: The constructor used to build a CNF from input of a DIMACS format.

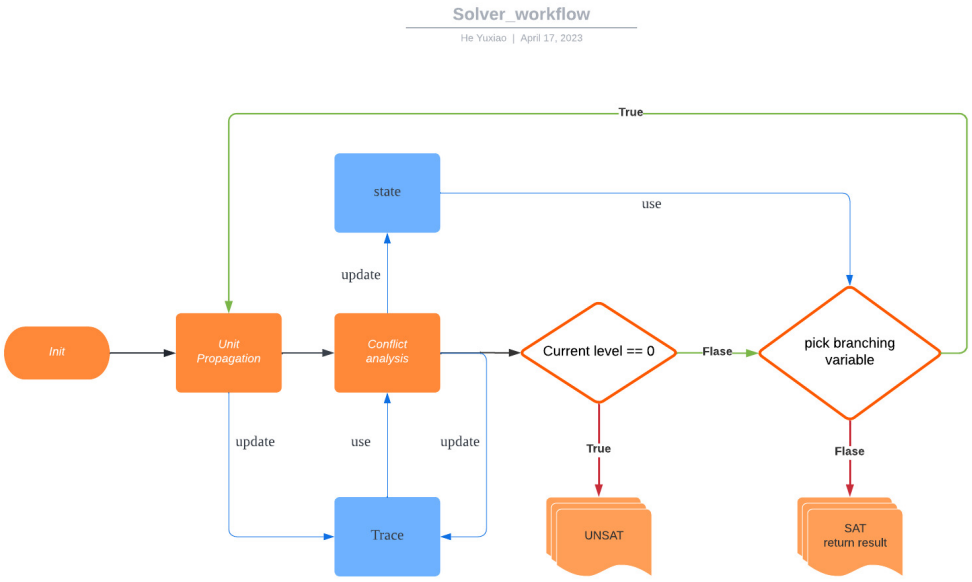
2. **trace**
   In this part, we build the data structure to record how each proposition is assigned. It is used in conflict analysis. This package includes:

   - **TraceUnit**: The unit which records how one proposition is assigned. It has three properties:
     - *literal*: the literal which the proposition is wrapped in.
     - *clause*: the clause which determines the assignment of the proposition if it is assigned in unit propagation; otherwise, this field is null.
     - *level*: the decision level at which the proposition is assigned.
   - **Trace**: A set of TraceUnits.

3. **state**
   This package includes the data structure needed in the VSIDS algorithm. It is used in picking the branching variable.

   - **score**: A structure that contains a literal and its score.
   - **state**: A set of scores.

4. **solver**
   This package only contains one class: Solver, which is our SAT solver. The workflow of the solver is shown in the diagram below:



5. **Main**
   The main function takes in a user input and converts it into a CNF. Then, it invokes the solveSAT method in Solver to get the result.

PickBranchingVariable

We chose a decision heuristic called VSIDS (Variable State Independent Decaying Sum) to pick the branching variable.
The idea is that we want to focus on the recently most critical propositions each time we do a picking.
It is a better heuristic than random choice and has been used in most modern SAT solvers.

ConflictAnalysis

We do conflict analysis by finding the UIP (unique implication point) and using it to get the reason set, which is the new clause learned in conflict.
The UIP can help us find a better conflict cut, making the most use of the information we have before the conflict. By doing so, the new clause learned will contain more constraints, which can accelerate the reasoning.