Assignment-2 CS202 Report

Team member:

1.Deepak Sangle (200860)

2.Vartika (201089)

1. Sat-solver (using DPLL)

• Implementation-

The DPLL is essentially a backtracking algorithm. The algorithm is building solution while trying assignments, you have a partial solution which might prove successful or not-successful as you go on.

The basic idea of the algorithm is:

1. "Guess" a variable
2. Find all unit clauses and assign the needed value
3. Iteratively retry step 2 until there is no change (found transitive closure)
4. If the current assignment cannot yield true for all clauses - fold back from recursion and retry a different assignment
5. If it can - "guess" another variable (recursively invoke and return to 1)

Termination:

1. If CNF formula doesn't contain any clause, then the formula is satisfiable and programme terminates.
2. If CNF formula contains at least one empty clause then the formula is unsatisfiable.
3. Let Φ be the CNF formula:

Algorithm DPLL takes set of clauses Φ as input and gives a truth value indicating whether Φ is satisfiable or not. Pseudo code is given below:-

```
function DPLL(Φ)
  while there is a unit clause {l} in Φ do
    Φ ← unit-propagate(l, Φ);
  while there is a literal l that occurs pure in Φ do
    Φ ← pure-literal-assign(l, Φ);
```

```
if $\Phi$ is empty then
    return true;
if $\Phi$ contains an empty clause then
    return false;
$l \leftarrow$ choose-literal($\Phi$);
return DPLL($\Phi \wedge \{l\}$) or DPLL($\Phi \wedge \{not(l)\}$);
```

Functions used in code:-

1. **DPLL:**

It is a recursive function which returns whether the CNF formula is satisfiable or unsatisfiable.

Initially it finds the clause with minimum number of literals .If this clause has no literals, then formula is unsatisfiable. Else, we select first literal (p) from this clause and assign true to it. Then we simplify the input by removing the clauses which contains p and remove ~p from all the clauses .This is known as unit propagation.

2. Simplify():

This function takes input as the set of clauses and the literal (p). It simplifies the input clauses by removing the clauses which contains p and remove ~p from all the clauses. This function also calls unit_literal() function to remove single literal clauses. It returns the modified set of clauses.

3. **unit_literal():**

This function removes all such clauses which contains only 1 literal and returns the modified set of clauses.

4. **smallest_clause():**

This function returns the index of the clause which contains minimum number of literals from the input set of clauses.

- **Limitation-** It takes longer time to execute for large value of propositions.

S.NO.	**Test case**	**Time Taken(s)**

1	uf20-01	0.03
2	uf20-02	0.01
3	uf20-03	0.12
4	uf20-04	0.20
5	uf150-01	45.83