

## COMP 610 FINAL PROJECT Working Synopsis

HARISH PINDI

In this algorithm, I have used two types of approaches to solve for Maximum truth values from the given File. Namely,

1. Brute force Approach
2. Quick Approach.

Working of the algorithm.

- Firstly, by means of main method, I am fetching the data from the file, using Buffer Reader.
- In general, the file includes lots of variables.
- To distinguish the variables that are used. I am splitting each line of the variables by means of space in to two parts, Variable1 and Variable 2. These variables are stored in array List variable temporarily.
- Depending up on the user's choice the algorithm works.

### 1. Brute force approach.

- We are passing the array list object which contains each of the variables.
- Now for each line there occurs duplicates.
- By means of **getDistinct** function , I am distinguishing the distinct variables used .
- Depending up on the number of variables used in the file, I can generate the truth table by means of  $2^n$  combinations by means of passing array list variable in to the truth table variable.

- Now My aim is to check the maximum number of satisfiable cases generated by the clauses from the file using the truth table combinations. For this I am using the for loop to iterate through each truth value in the truth table.
- For this operation I am sending the truth value combinations to Max2sat function.
- These Max2Sat function will count the maximum satisfiable cases generated from the truth table combinations.
- Once all the truth table combinations got finished, it will enlist the maximum number of truth cases generated using the combination.

## **2. Quick Approach.**

- IN this approach, we use greedy strategy.
- At first all the distinct variables get stored in an array.
- We assign all these variables to the variable x.
- Next for each of the clauses we have in the file we assign first variable to y and next variable to be Z.
- We then compare variable X values to each of the clause values of Y and Z.
- If both are positive then we return positive or else negative. Also, we count the positives and negatives generated from each of the clause.
- At last we will count the positives and negatives. If the count of the pos > Neg we assign T or else F.
- These obtained truth values will now use Max2Sat function to count the Max truth values obtained from the given file.

## Running procedure:

I have assigned the each file directory in the main method.

## Outputs Generated:

### Test1(75032)

```
C:\Users\harishp647\Desktop\New folder>java HarishMax2Sat

Press 1 for Brute-force approach
Press 2 for quick approach

Enter your selection: 2

T T T F T T T T T T T T T T T T T T T T T T 75032

C:\Users\harishp647\Desktop\New folder>
```

### Test2(151071)

```
Command Prompt

Note: HarishMax2Sat.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\harishp647\Desktop\New folder>java HarishMax2Sat

Press 1 for Brute-force approach
Press 2 for quick approach

Enter your selection: 2

F T T T T T F T T F F T T T T T T T T T F T F T F T F T T T T T T F F F T
T T T T F F T T T T T F F F F F F T F F T T F T T T F T T F T T T T F F F F
F F T F F T F T F F T T T F T F T T F T 151071
```

### Test3: (171992)

```
C:\Users\harishp647\Desktop\New folder>java HarishMax2Sat

Press 1 for Brute-force approach
Press 2 for quick approach

Enter your selection: 2

F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F 171992

C:\Users\harishp647\Desktop\New folder>
```

