Beatrice Piras s259888

Exam of the 28-01-2020 Report of the program (18 points version)

**Implementation choices and data structure used:**

I used a matrix of a structure (val) to save the data from the file.

The structure val includes an int value and an int negation.

The matrix has N (number of AND operations) rows and M (number of total variables) columns as the file gives us the numbers in the first row.

In every row of the matrix there are the OR operations (the ones represented inside a parenthesis)

*Example: (x1+x2+x3NOT) is saved in a row of the matrix*

The presence of a variable in the row is indicated with a '1', and the absence of this with a '0'. For the negation operand, the structure val includes an int negation that is initialized to 0 for all the matrix but set to 1 when the number acquired from the file is negative.

To find the solution of the variables, I first created an array of integers sol and initialized to -1, I then modified its values ( to 1 or 0) through a recursive function using powerset and a check function when the sol array is completely filled with values.

In the check function I call a check_row function for every row, to check that the OR operations in each row give 1 as result, so if there is at least one variable (or its negation if it is the case) that has 1 as solution in the sol array. If all rows give 1 then the solution is confirmed and the recursion function terminates.

The result is then printed in standard output, or if there aren't any solutions a message of unsatisfaction is printed.

**Differences in the code:**

The code implemented is mostly the same as the code written in paper.

A small addiction I made to my code is a function to initialize the row of the matrix to 0.

The only other differences are some minor syntax mistakes: a parameter forgotten when calling a function, a parenthesis not closed, an [i] I forgot to delete after changing an int from int * in the file_read function.