1. Introduction:

A python code is written to generate Reduced Ordered Binary Decision Diagram for a given variable ordering. It has various definitions like to find Shannon Co-factors, to generate BDD, to apply reducing operations, to plot the reduced BDD.

2. Inputs:

There are two inputs that a user should input to plot the ROBDD.

1. Boolean function: It shall be given in SOP format with complemented variables as uppercase letters.
   E.g.: (ab + a'b') should be entered as (ab + AB)

2. Variable Ordering: It shall be given as a string from the keyboard separated by '<'.
   E.g.: If the order of variables is [a, b, c], it should be entered as a<b<c

3. Outputs:

The output is displayed in two formats.

1. csv file: A csv file is generated with function pointers to each node and each node has two children: 0 Pointer and 1 Pointer. Each node is labelled with a variable in the column next to pointers.

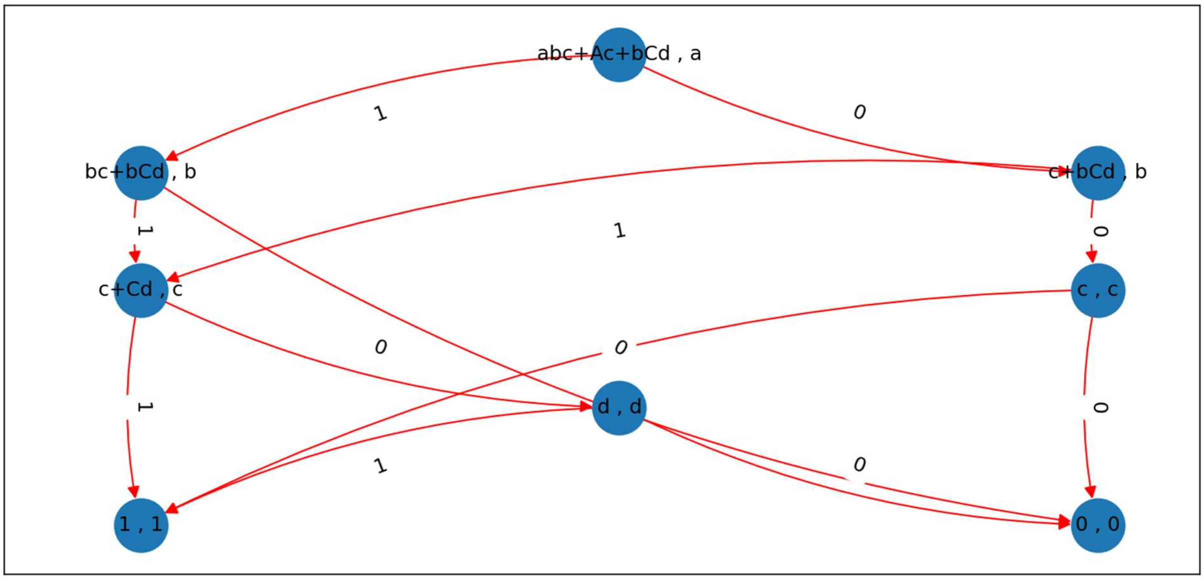
2. Directed Acyclic Graph: Each node is represented by a label of function and variable, separated by a comma (,). The nodes are connected by edges, with labels '0', '1' representing the 0 – edge and 1 – edge.

3.1 Sample Outputs:

Note: The output shown in below table are formatted for better view. The actual view of the output will be .csv file, without borders and alignment rules of MS Excel.

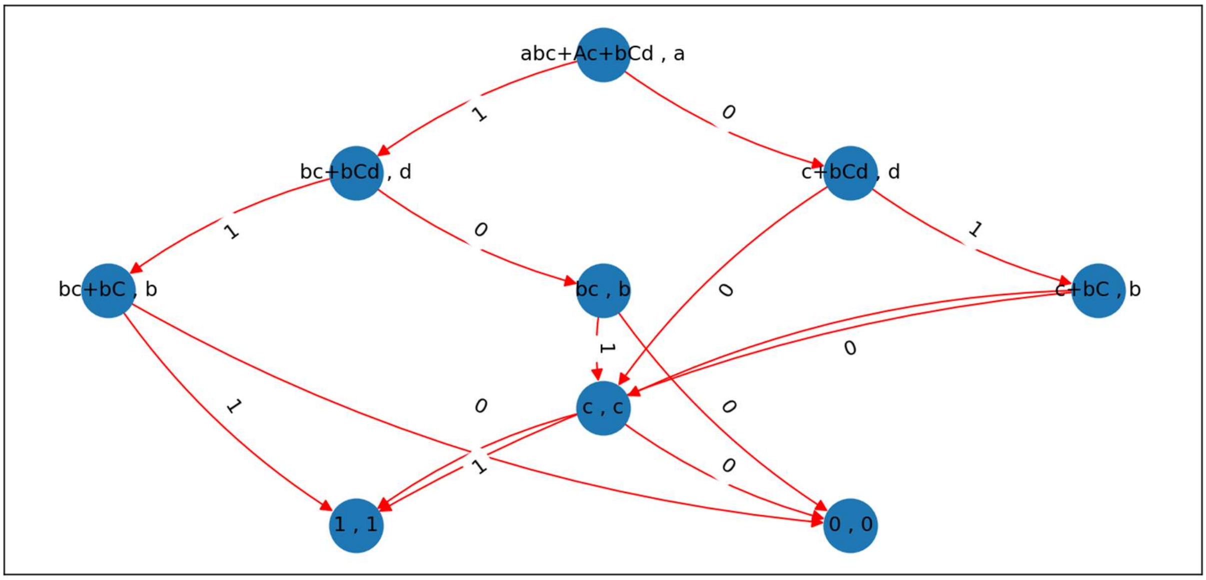
1. Boolean Function: abc+Ac+bCd   Variable Ordering: a<b<c<d

Pointer	Main node	0 Pointer	0 Node	1 Pointer	1 Node
abc+Ac+bCd	a	c+bCd	b	bc+bCd	b
c+bCd	b	c	c	c+Cd	c
bc+bCd	b	0	0	c+Cd	c
c	c	0	0	1	1
c+Cd	c	d	d	1	1
d	d	0	0	1	1



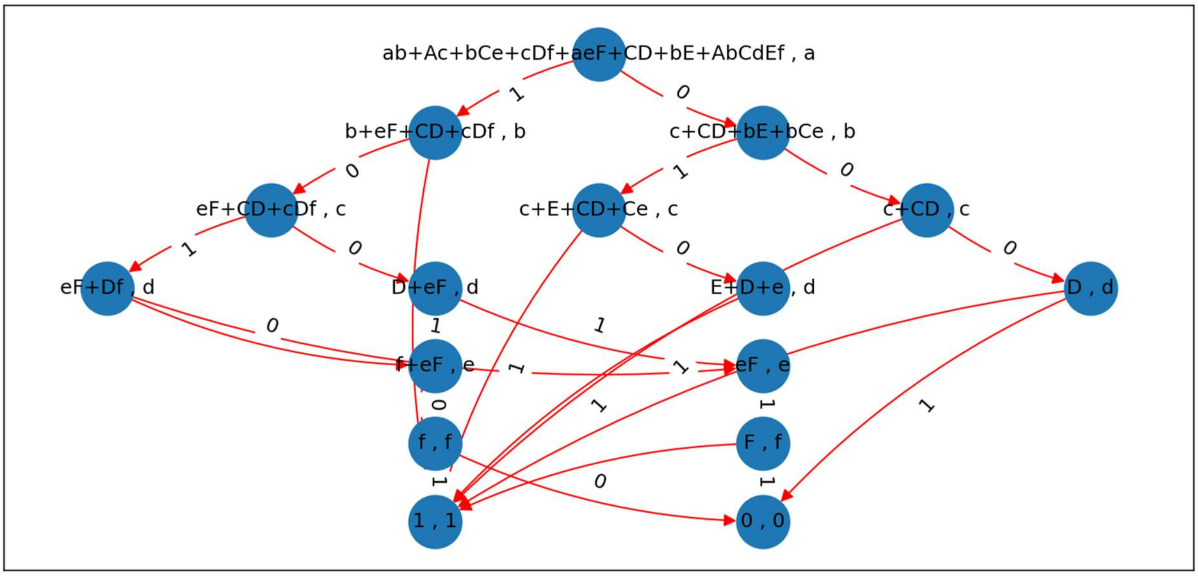
2. Boolean Function: abc+Ac+bCd   Variable Ordering: a<d<b<c

Pointer	Main node	0 Pointer	0 Node	1 Pointer	1 Node
abc+Ac+bCd	a	c+bCd	d	bc+bCd	d
c+bCd	d	c	c	c+bC	b
bc+bCd	d	bc	b	bc+bC	b
c+bC	b	c	c	1	1
bc	b	0	0	c	c
bc+bC	b	0	0	1	1
c	c	0	0	1	1



3. Boolean Function: ab+Ac+bCe+cDf+aeF+CD+bE+AbCdEf   Variable Ordering: a<b<c<d<e<f

Pointer	Main node	0 Pointer	0 Node	1 Pointer	1 Node
ab+Ac+bCe+cDf+aeF+CD+bE+AbCdEf	a	c+CD+bE+bCe	b	b+eF+CD+cDf	b
c+CD+bE+bCe	b	c+CD	c	c+E+CD+Ce	c
b+eF+CD+cDf	b	eF+CD+cDf	c	1	1
c+CD	c	D	d	1	1
c+E+CD+Ce	c	E+D+e	d	1	1
eF+CD+cDf	c	D+eF	d	eF+Df	d
D	d	1	1	0	0
E+D+e	d	1	1	1	1
D+eF	d	1	1	eF	e
eF+Df	d	f+eF	e	eF	e
eF	e	0	0	F	f
f+eF	e	f	f	1	1
F	f	1	1	0	0
f	f	0	0	1	1



4. Future scope

This code can be further modified by removing the data frame and plotting the graphs in a much cleaner and understandable way.