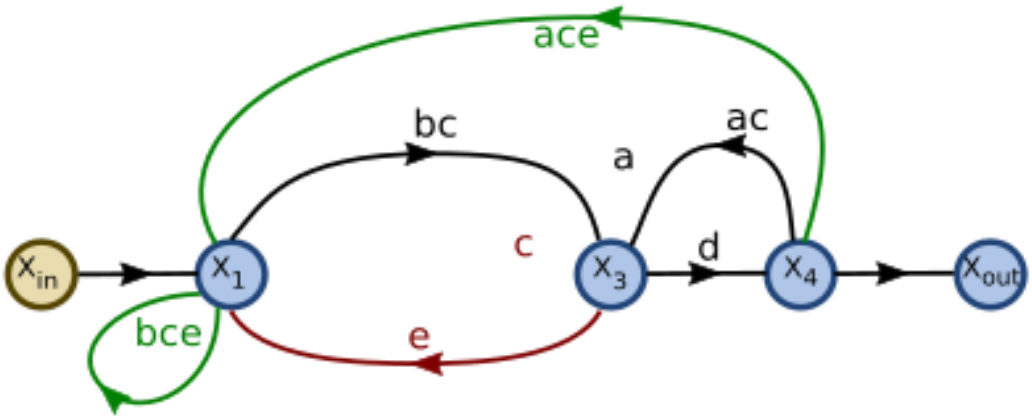
Signal Flow Graph Solver

Control Systems Analysis



Khadija Assem Saad Hamad

ID : 27

Source Code : https://github.com/khadijaAssem/SignalFlowGraphSolver

Overview

Signal flow graph is an alternative method to block diagram representation, It consists of a network in which nodes are connected by directed branches.Mason's rule is used for reducing the signal-flow graph to a single transfer function.

Mason's Formula states that : $$\frac{C(s)}{R(s)} = \frac{\sum_{i=1}^{n} P i \Delta i}{\Delta}$$

where ,

n = number of forward paths.

Pi = the i th forward-path gain.

Δ = Determinant of the system

Δi = Determinant of the ith forward path

Project Description

This project is the implementation of the Mason's Formula, it takes the signal flow graph as an input and outputs the corresponding transfer function.

Main Features :

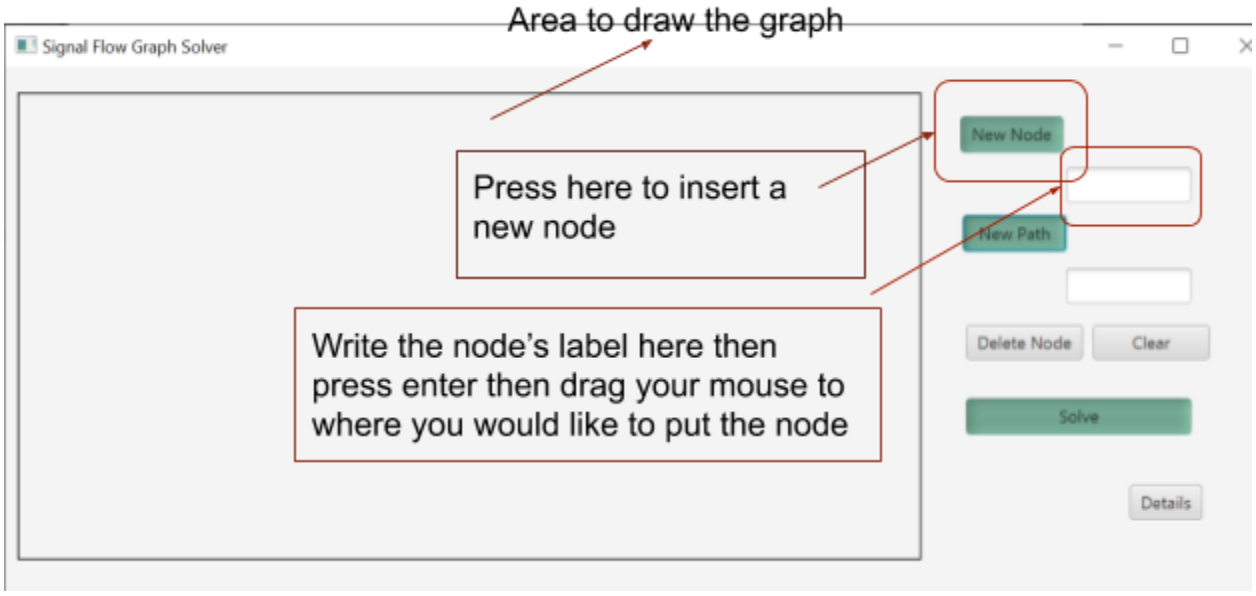
1. You can draw the signal flow graph and put the nodes and paths in a very simple way.
2. You can also delete nodes easily.

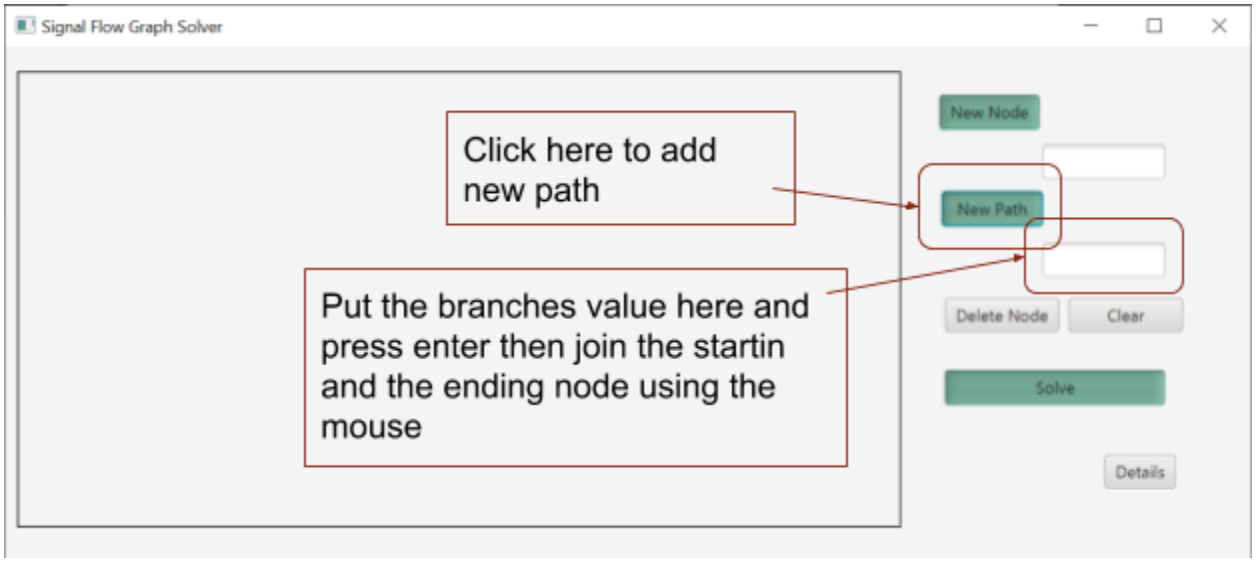
3. The program supports both numerical and symbolic representation.
4. After finishing the program can give you clear details about the calculations and most of the intermediate calculations for calculating the transfer function.

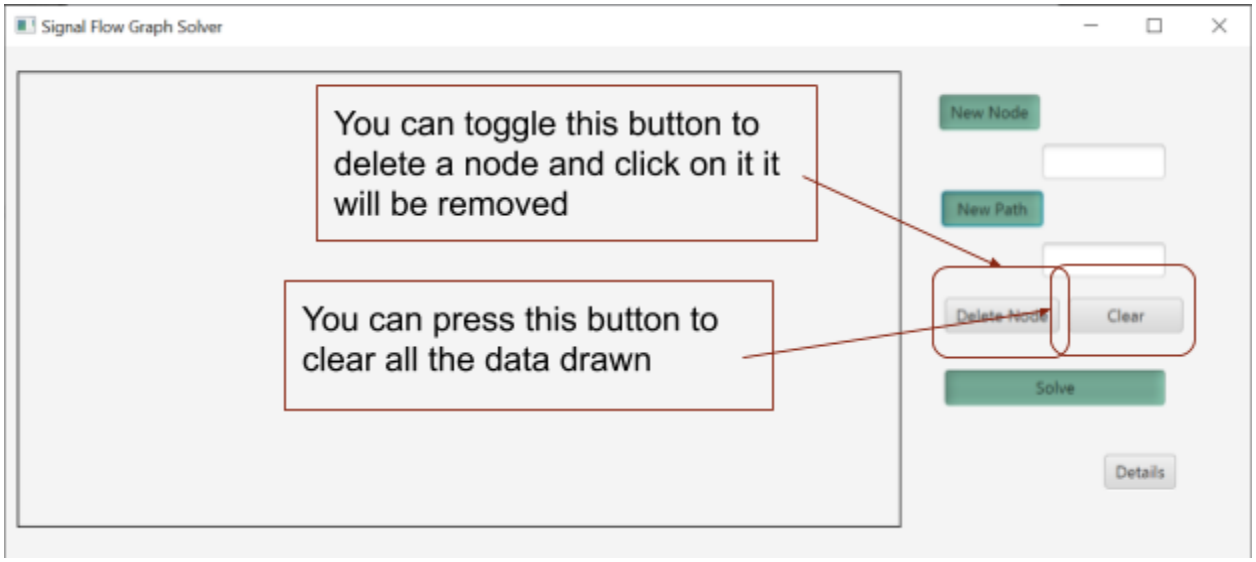
Main Algorithms Used:

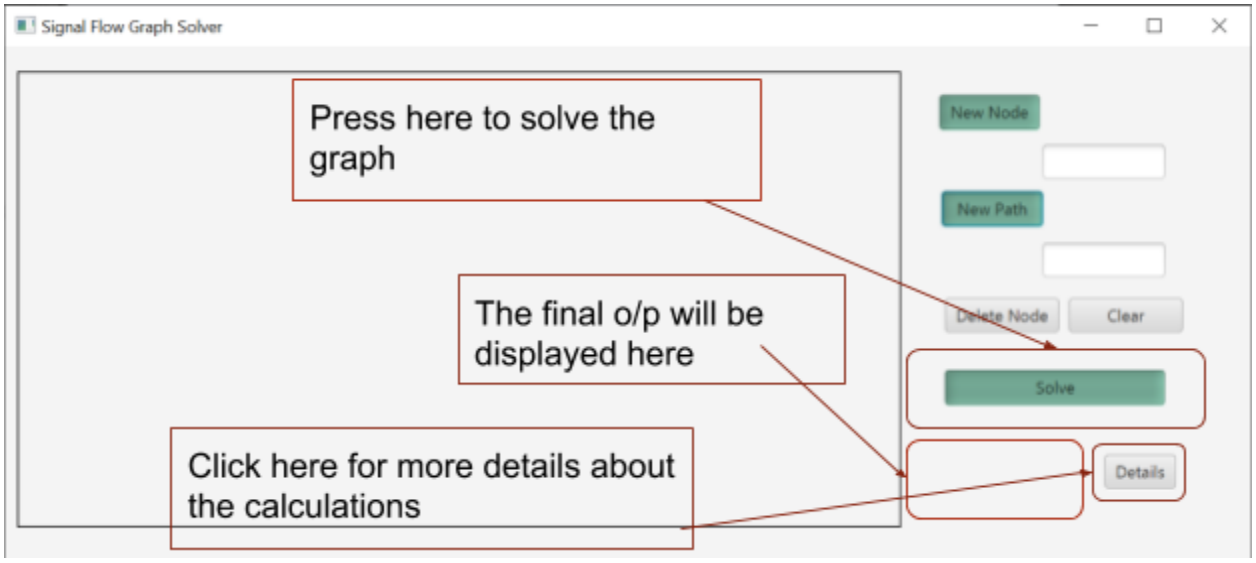
1. First the program uses the DFS (depth first search) algorithm to find all forward passes .
2. Combinations to find the non intersecting loops .

User Guide :

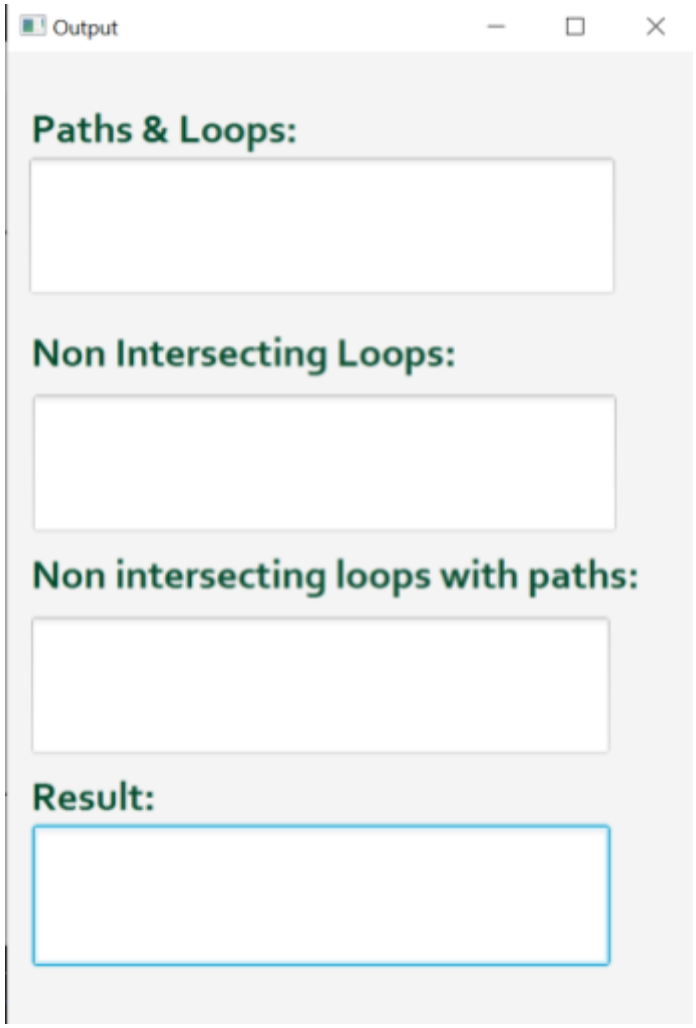



Signal Flow Graph Solver
Click here to add new path
Put the branches value here and press enter then join the startin and the ending node using the mouse
New Node
New Path
Delete Node
Clear
Solve
Details


Signal Flow Graph Solver
You can toggle this button to delete a node and click on it it will be removed
You can press this button to clear all the data drawn
New Node
New Path
Delete Node
Clear
Solve
Details


Signal Flow Graph Solver
Press here to solve the graph
The final o/p will be displayed here
Click here for more details about the calculations
New Node
New Path
Delete Node
Clear
Solve
Details

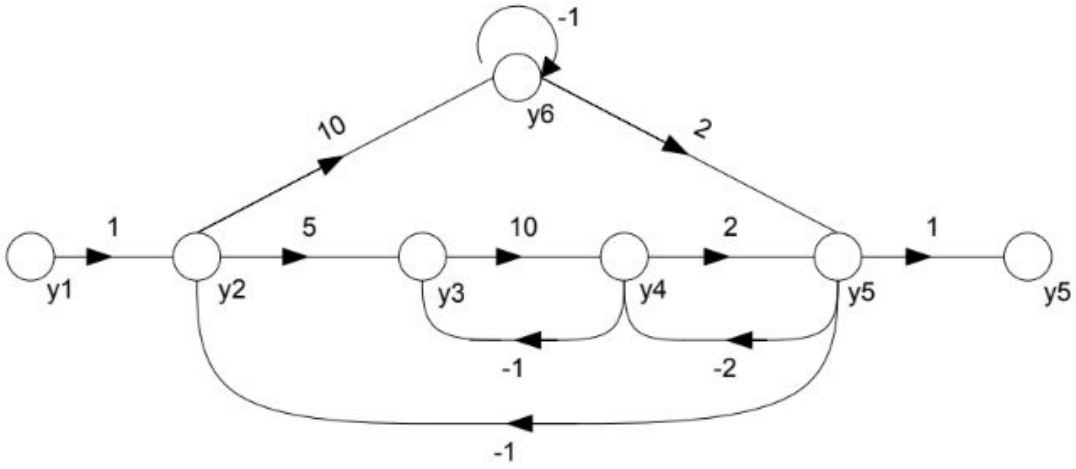
The detailed output is then displayed here :

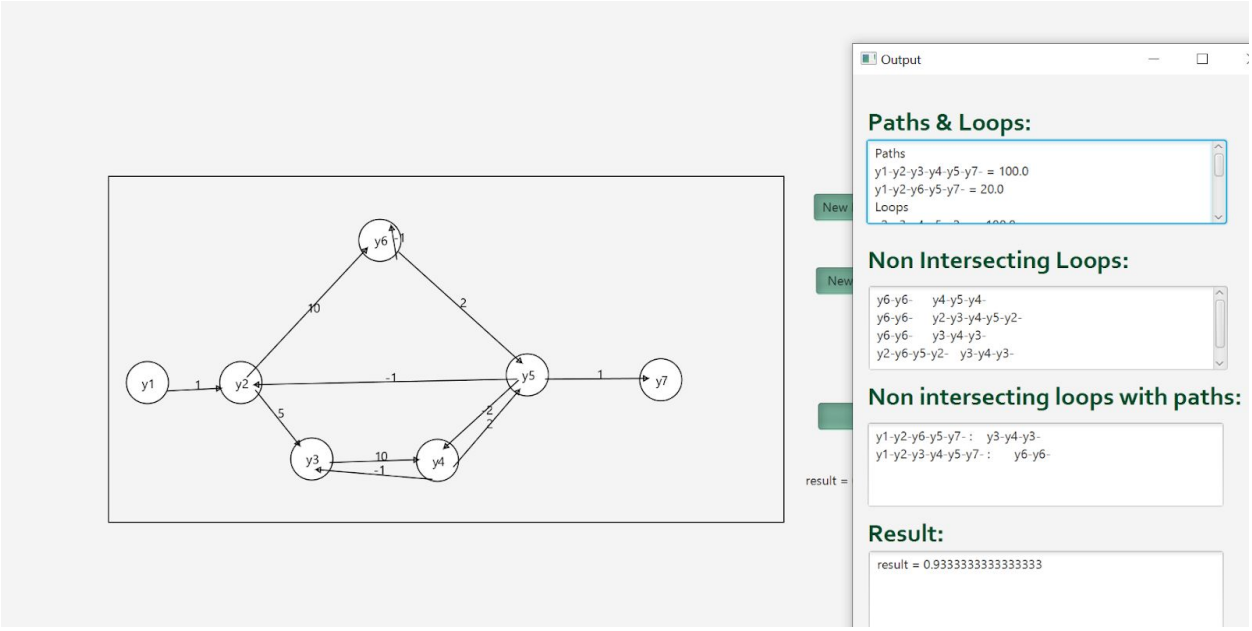


Sample Runs :

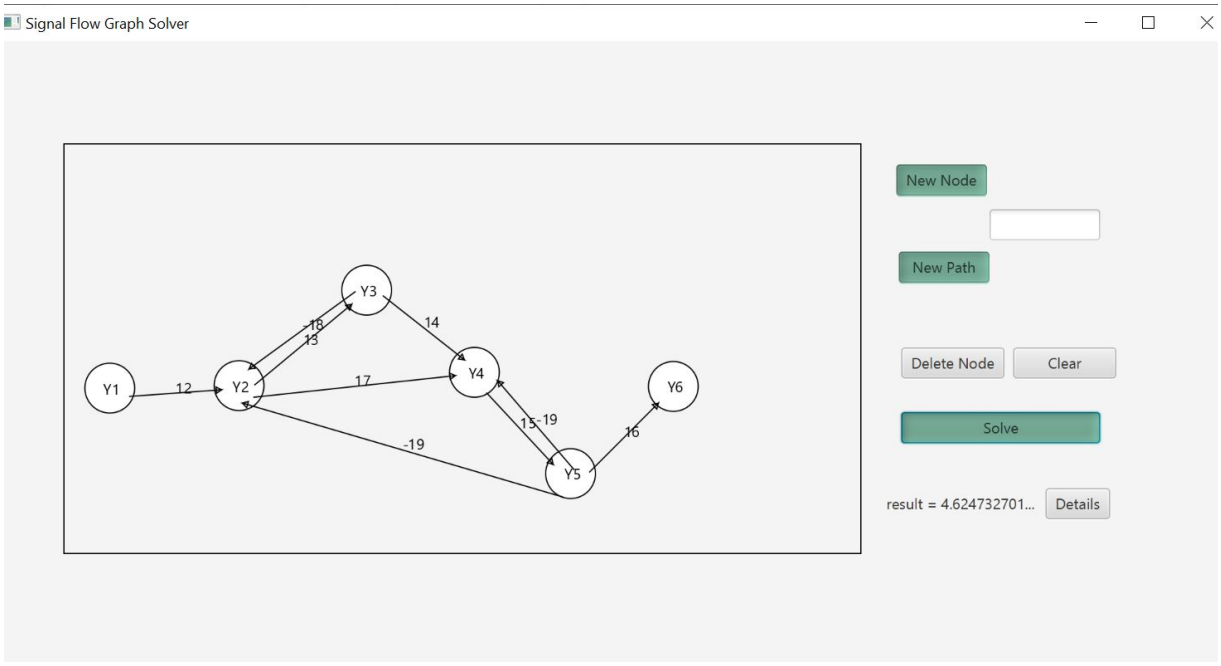
For the numeric input :

Example 1



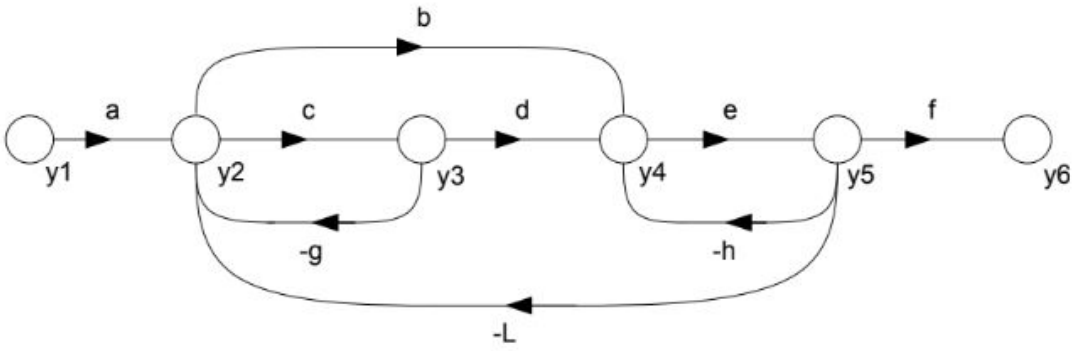


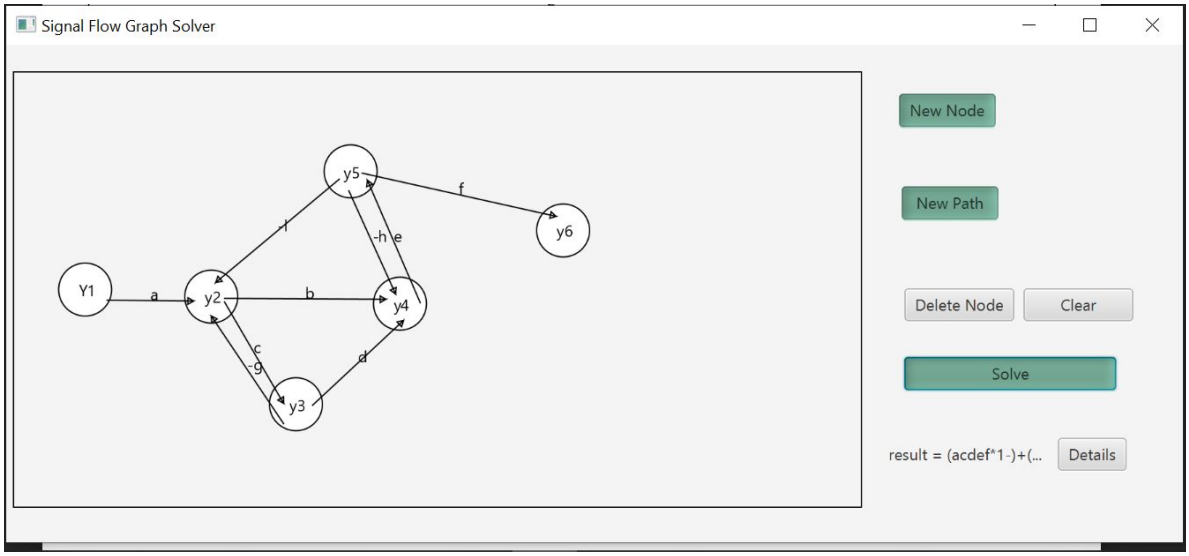
Example 2

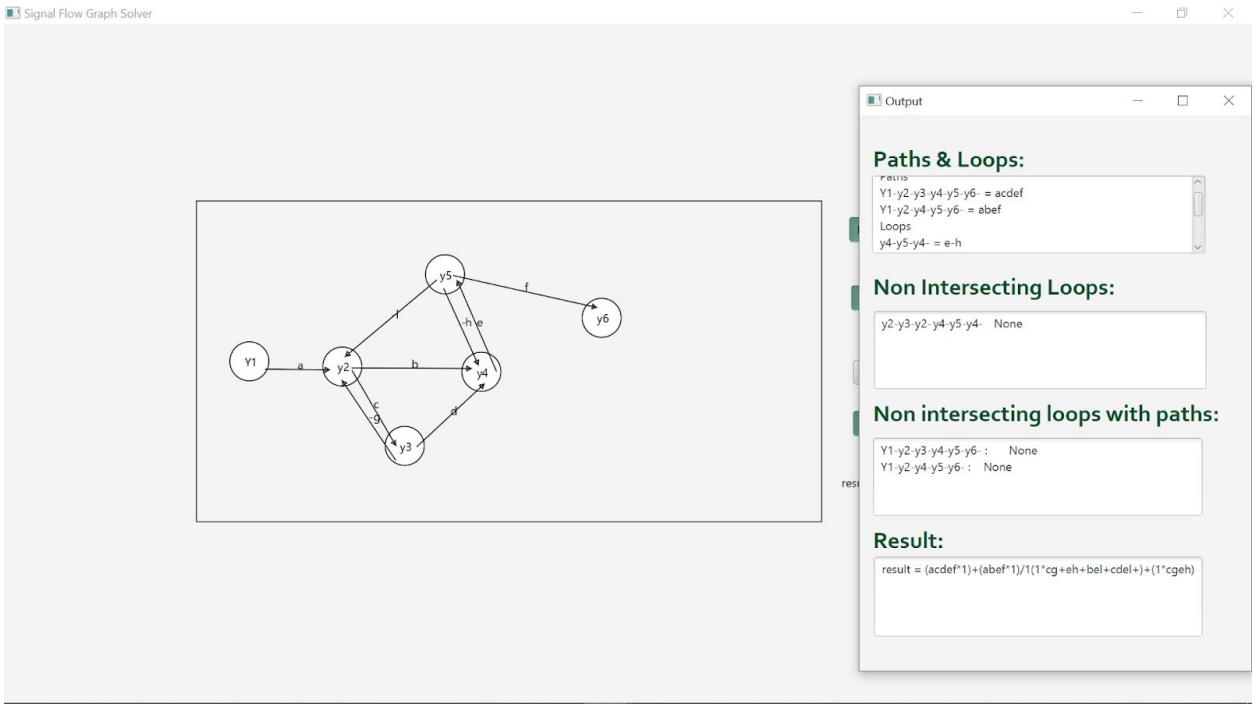


For alphabetic input

Example 1




Signal Flow Graph Solver
y5
f
y6
-i
-h
e
Y1
a
y2
b
y4
c
-g
d
y3
New Node
New Path
Delete Node
Clear
Solve
result = (acdef*1-)+(...
Details


Signal Flow Graph Solver
y5
f
y6
-i
-h
e
Y1
a
y2
b
y4
c
-g
d
y3
Output
Paths & Loops:
Y1-y2-y3-y4-y5-y6- = acdef
Y1-y2-y4-y5-y6- = abef
Loops
y4-y5-y4- = e-h
Non Intersecting Loops:
y2-y3-y2-y4-y5-y4- None
Non intersecting loops with paths:
Y1-y2-y3-y4-y5-y6- : None
Y1-y2-y4-y5-y6- : None
Result:
result = (acdef*1)+(abef*1)/1(1*cg+eh+bel+cdel+)+(1*cgeh)