DSR-graph, positive feedback loops and injectivity
November 2014

Initializations

```
> restart :
> interface(rtablesize = 40) :
> with(ListTools) :
> with(LinearAlgebra) :
> with(GraphTheory) :
> with(combinat) :
> _Envsignum0 := 0 :
```

Execute all the comands in this section below. You can select it and then press the symbol "!" above. This will execute the selected region.

▶ To execute before proceeding

A is the stoichiometric matrix of the network.

Step 1.
Create the DSR-graph and find if there is a positive feedback loop through the competition.

To create the DSR-graph, do:

```
> findedgesDSR(mylabels, A)
```

where mylabels contains the names of the nodes.

Step 2
The next step is to find out which of the selected networks can have multiple steady states.
Here are a few tests to run before using the toolbox.

Injectivity test:

```
> isinjective(A)
```

$$0 \quad (1)$$

If the answer is 1, then classify the network as NOT MULTISTATIONARY: there can be either none or one positive steady state
If the answer is 0, then apply the next test.

At this point, if there are not many networks one can use the toolbox. If there are many networks, one can apply the next test. It takes though some time to run.

> $isinjectiveextended(A)$

$$2 \tag{2}$$

If the answer is 1, then classify the network as NOT MULTISTATIONARY.
If the answer is 2, then classify the network as MULTISTATIONARY.
If the answer is 3, then the network does not have positive steady states.
If the answer is 0, then use the toolbox.

Step 3
For the networks that are multistationary, decide whether multistationarity can be attributed to competition, and find the positive feedback loops underlying multistationarity.

For that, we do:

> $Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :$

> $selected := findloops(\,)$

The procedure returns the list of positive feedback loops that underly multistationarity.
You can draw the loops after giving a label to the species, see the examples below.

The test could also be applied to the non-multistationary networks. Some of them would return some loops, and for some of them nothing would be returned.

Step 4.
The next step would be to understand what causes multistationarity. If it relates to the loops, then we should test if the networks that do not have multistationarity have that loop structure in their DSR-graph.

This part is still open.

Tricks:
If network N is multistationary and N' is another network that is identical to N **but has extra reactions** that do not change the dimension of the stoichiometric space, i.e. the rank of N' is the same as the rank of N, then N' is also multistationary.

Example 1

Consider the network
A+B< -> C
D+B->E ->A+F
A->D
F->B

The stoichiometric matrix A is:

> $A := Transpose(Matrix([[-1,-1,1,0,0,0],-[-1,-1,1,0,0,0],[0,-1,0,-1,1,0],[1,0,0,0,-1,1],[-1,0,0,1,0,0],[0,1,0,0,0,-1]]));$

$$A := \begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \tag{2.1}$$

Step 1.
Create the DSR-graph and find if there is a positive feedback loop through the competition.

> $mylabels := [\text{"A"}, \text{"B"}, \text{"C"}, \text{"D"}, \text{"E"}, \text{"F"}, seq(R_i, i = 1..6)]:$

> $findedgesDSR(mylabels, A)$

$$\{[[\text{"A"}, R_1], 1], [[\text{"A"}, R_5], 1], [[\text{"B"}, R_1], 1], [[\text{"B"}, R_3], 1], [[\text{"C"}, R_2], 1], [[\text{"D"}, R_3], 1], [[\text{"E"}, R_4], 1], [[\text{"F"}, R_6], 1], [[R_1, \text{"A"}], -1], [[R_1, \text{"B"}], -1], [[R_1, \text{"C"}], 1], [[R_2, \text{"A"}], 1], [[R_2, \text{"B"}], 1], [[R_2, \text{"C"}], -1], [[R_3, \text{"B"}], -1], [[R_3, \text{"D"}], -1], [[R_3, \text{"E"}], 1], [[R_4, \text{"A"}], 1], [[R_4, \text{"E"}], -1], [[R_4, \text{"F"}], 1], [[R_5, \text{"A"}], -1], [[R_5, \text{"D"}], 1], [[R_6, \text{"B"}], 1], [[R_6, \text{"F"}], -1]\} \tag{2.2}$$

If one wants to draw the graph, then one does:

> $G := createDSRgraphsigned(mylabels, A, findZ(A)):$

> $DrawGraph(G)$

Step 2

Injectivity test:

> $isinjective(A)$

$$0 \tag{2.3}$$

Because the answer is 0, we apply the next test.

> $isinjectiveextended(A)$

$$2 \tag{2.4}$$

Because the answer is 2, we classify the network as MULTISTATIONARY.

Step 3
For the networks that are multistationary, decide whether multistationarity can be attributed to competition, and find the positive feedback loops underlying multistationarity.

> $Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :$

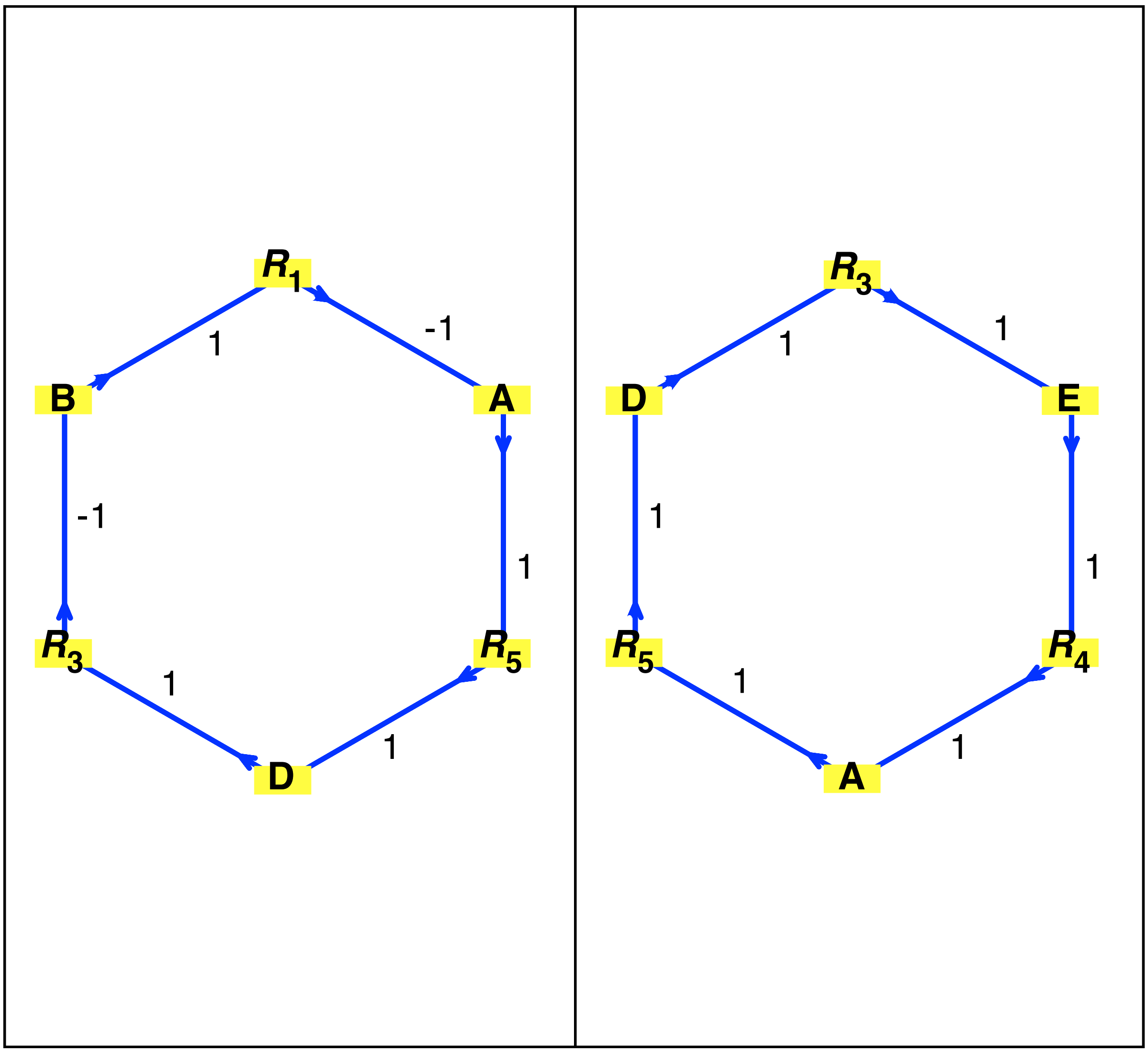
> $selected := findloops(\,)$

$$selected := [[[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]], [[[R_3, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], 1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1]]] \qquad (2.5)$$

These are the positive feedback loops. You can draw the loops after giving a label to the species:

> $speciesord := [\text{"A"}, \text{"B"}, \text{"C"}, \text{"D"}, \text{"E"}, \text{"F"}] :$

> $drawloops(selected, speciesord)$



Example 2

A+B< -> C
D+B->G ->D+F
A->D
G->C
F->B

The stoichiometric matrix A is:

> $A := Transpose(Matrix([[-1,-1, 1, 0, 0, 0],-[-1,-1, 1, 0, 0, 0], [0,-1, 0,-1, 0, 1], [0, 0, 0, 1, 1,-1], [-1, 0, 0, 1, 0, 0], [0, 0, 1, 0, 0,-1], [0, 1, 0, 0,-1, 0]]));$

$$A := \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 \end{bmatrix} \quad \textbf{(3.1)}$$

Step 1.
Create the DSR-graph and find if there is a positive feedback loop through the competition.

> $mylabels := ["A", "B", "C", "D", "F", "G", seq(R_i, i = 1..7)]:$

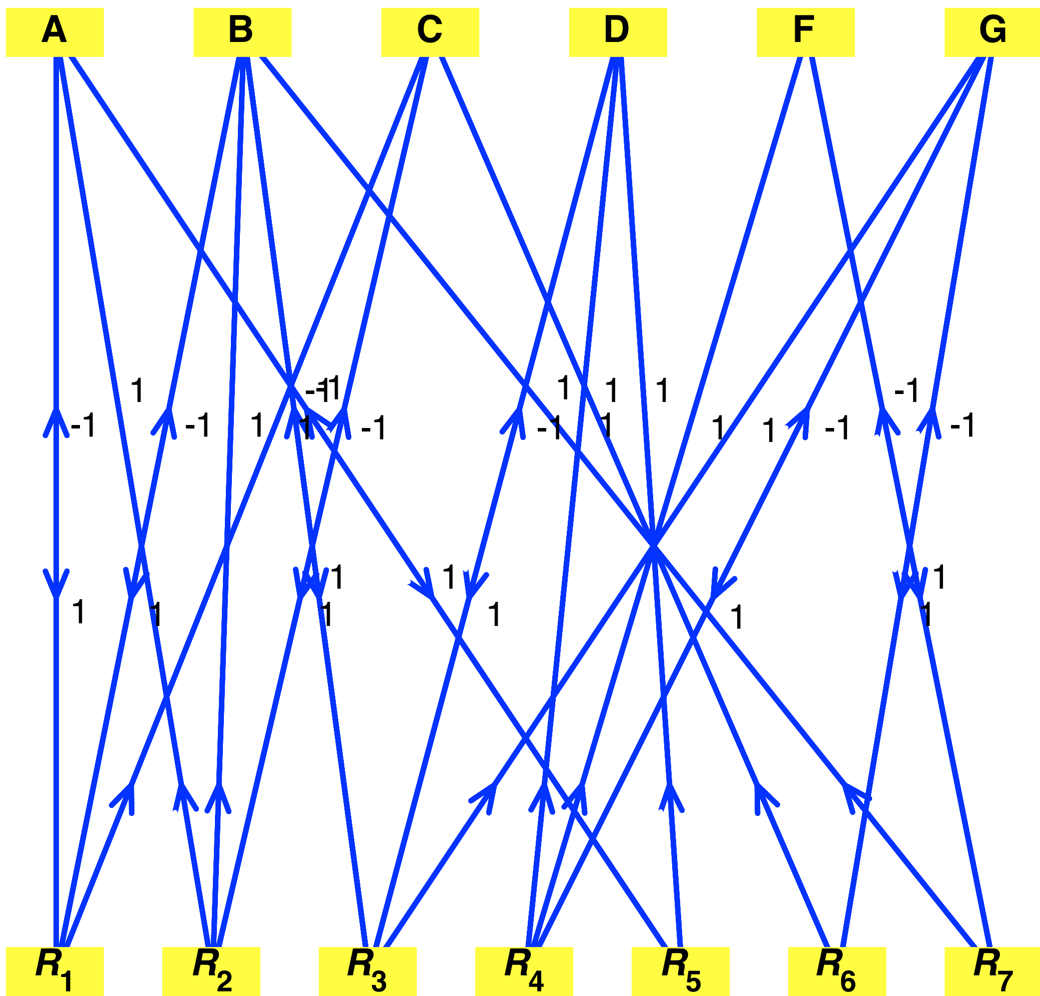
> $findedgesDSR(mylabels, A)$

$\{[["A", R_1], 1], [["A", R_5], 1], [["B", R_1], 1], [["B", R_3], 1], [["C", R_2], 1], [["D", R_3], 1], [["F", R_7], 1], [["G", R_4], 1], [["G", R_6], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], -1], [[R_3, "D"], -1], [[R_3, "G"], 1], [[R_4, "D"], 1], [[R_4, "F"], 1], [[R_4, "G"], -1], [[R_5, "A"], -1], [[R_5, "D"], 1], [[R_6, "C"], 1], [[R_6, "G"], -1], [[R_7, "B"], 1], [[R_7, "F"], -1]\}$ **(3.2)**

If one wants to draw the graph, then one does:

> $G := createDSRgraphsigned(mylabels, A, findZ(A)) :$

> $DrawGraph(G)$



Step 2

Injectivity test:

> $isinjective(A)$

$$0 \tag{3.3}$$

Because the answer is 0, we apply the next test.

> $isinjectiveextended(A)$

$$0 \tag{3.4}$$

Because the answer is 0, you should use the toolbox.
The toolbox says YES, the network admits multiple steady states

Step 3
We find the positive feedback loops.

> $Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :$

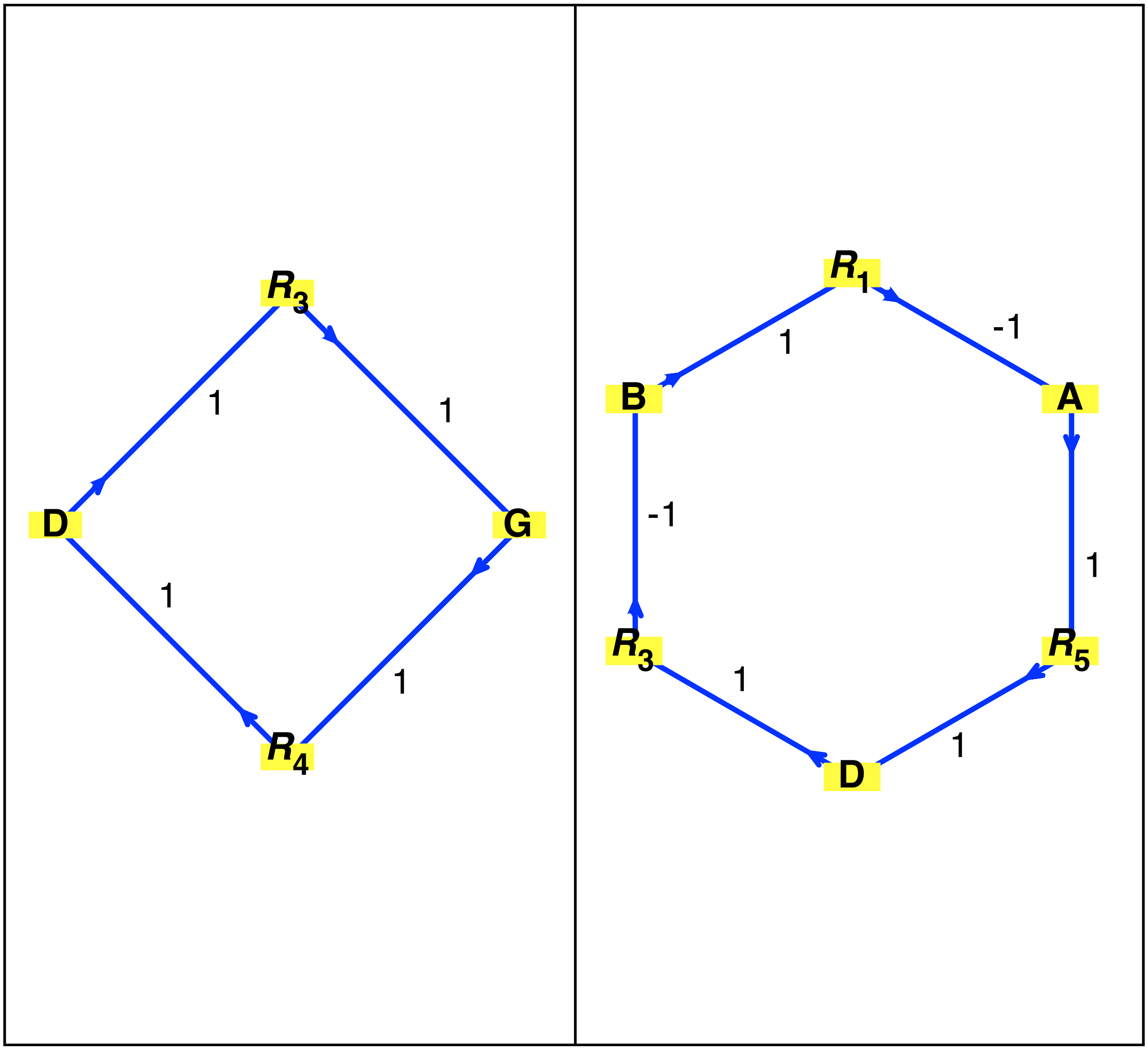
> $selected := findloops(\,)$

$$selected := [[[[R_3, S_6], 1], [[S_6, R_4], 1], [[R_4, S_4], 1], [[S_4, R_3], 1]], [[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]]] \tag{3.5}$$

These are the positive feedback loops. You can draw the loops after giving a label to the species:

> $speciesord := ["A", "B", "C", "D", "F", "G"] :$

> $drawloops(selected, speciesord)$



Example 3

Consider the network
A+B< -> C

D+B->E ->A+F
A->D
F->B

The stoichiometric matrix A is:

> $A := Transpose(Matrix([[-1,-1,1,0,0,0],-[-1,-1,1,0,0,0],[0,-1,0,-1,1,0],[1,0,0,0,-1,1],[-1,0,0,1,0,0],[0,1,0,0,0,-1]]));$

$$A := \begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \tag{4.1}$$

Step 1.
Create the DSR-graph and find if there is a positive feedback loop through the competition.

> $mylabels := ["A", "B", "C", "D", "E", "F", seq(R_i, i = 1..6)]:$

> $findedgesDSR(mylabels, A)$

$$\{[["A", R_1], 1], [["A", R_5], 1], [["B", R_1], 1], [["B", R_3], 1], [["C", R_2], 1], [["D", R_3], 1], [["E", R_4], 1], [["F", R_6], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], -1], [[R_3, "D"], -1], [[R_3, "E"], 1], [[R_4, "A"], 1], [[R_4, "E"], -1], [[R_4, "F"], 1], [[R_5, "A"], -1], [[R_5, "D"], 1], [[R_6, "B"], 1], [[R_6, "F"], -1]\} \tag{4.2}$$

If one wants to draw the graph, then one does:

> $G := createDSRgraphsigned(mylabels, A, findZ(A)):$

> $DrawGraph(G)$

Step 2

Injectivity test:

> $isinjective(A)$

$$0 \tag{4.3}$$

Because the answer is 0, we apply the next test.

> $isinjectiveextended(A)$

$$2 \quad (4.4)$$

Because the answer is 2, we classify the network as MULTISTATIONARY.

Step 3

For the networks that are multistationary, decide whether multistationarity can be attributed to competition, and find the positive feedback loops underlying multistationarity.

> $Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :$

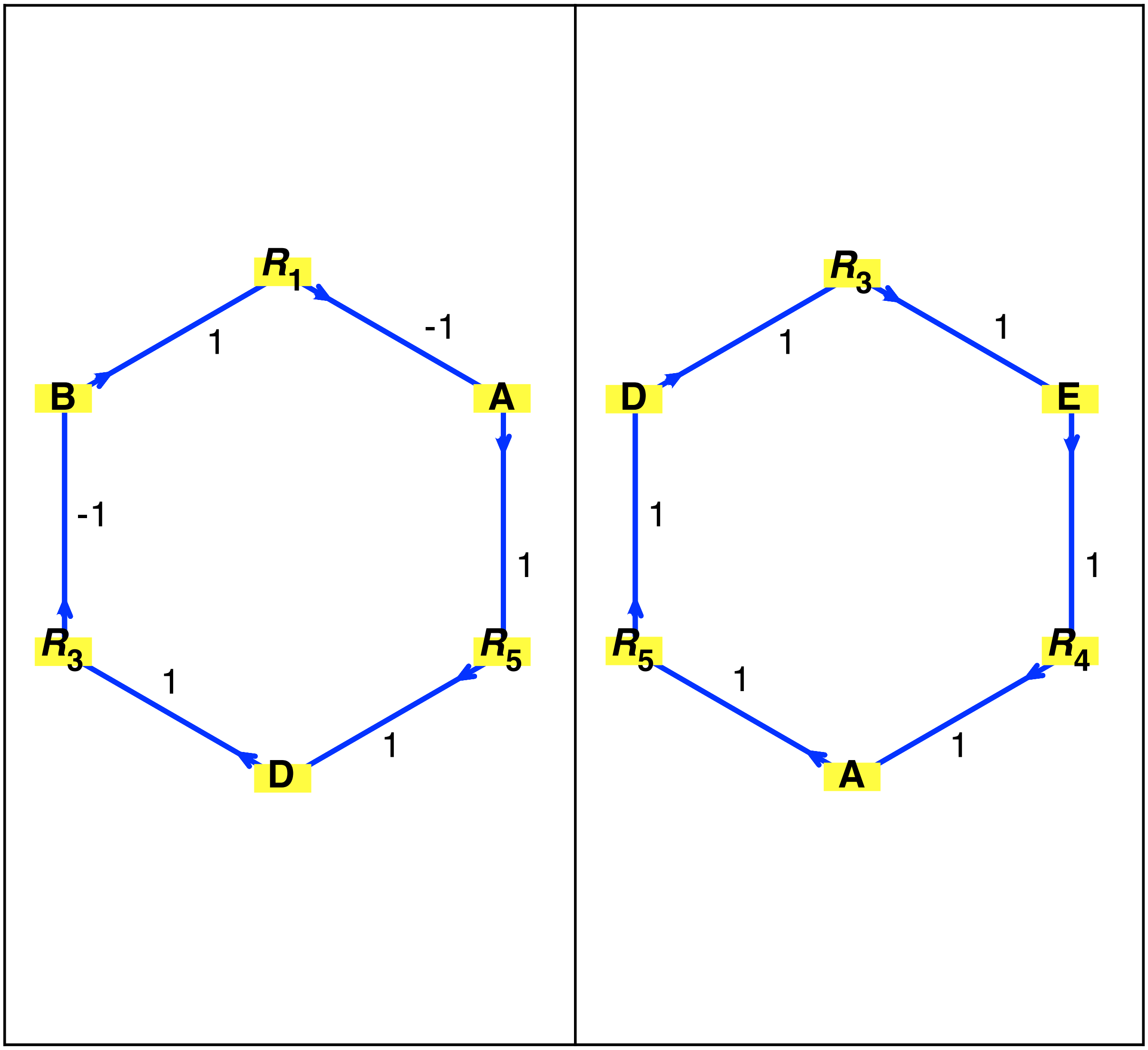
> $selected := findloops(\,)$

$$selected := [[[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]], [[[R_3, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], 1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1]]] \quad (4.5)$$

These are the positive feedback loops. You can draw the loops after giving a label to the species:

> $speciesord := [\text{"A"}, \text{"B"}, \text{"C"}, \text{"D"}, \text{"E"}, \text{"F"}] :$

> $drawloops(selected, speciesord)$



R_1
-1
1
B
A
-1
1
R_3
R_5
1
1
D
R_3
1
1
D
E
1
1
R_5
R_4
1
1
A