Draw intersecting positive feedback loops for all bistable and monostable networks
January 2015

Initializations

To execute before proceeding

Comment

Hypothesis 1: two PF loops, one of them is competition loop, S to R interesecting edge, competitor species on the edge and competed species not on the edge.

Hypothesis 2: (?with two competing futile cycles) two PF loops, one of them is competition loop, S to R interesecting edge, substrate (?after intermediate complex one species is transformed into another species) not on the edge.

Bistable networks

All 8 bistable networks meet the condition in hypothesis 2; network 1, 2, 3, 5, 7, 8 meet condition in hypothesis 1.

Network 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]]));
```

$$A := \begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \qquad \textbf{(4.1.1)}$$

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)
```

$$\{[["A", R_1], 1], [["A", R_5], 1], [["B", R_1], 1], [["B", R_3], 1], [["C", R_2], 1], [["D", R_3], 1], [["E", R_4], 1], [["F", R_6], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], -1], [[R_3, "D"], -1], [[R_3, "E"], 1], [[R_4, "A"], 1], [[R_4, "E"], -1], [[R_4, "F"], 1], [[R_5, "A"], -1], [[R_5, "D"], 1], [[R_6, "B"], 1], [[R_6, "F"], -1]\} \qquad \textbf{(4.1.2)}$$

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

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

Step 2

Injectivity test:

```
> isinjective(A)
```

$$0 \qquad \textbf{(4.1.3)}$$

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

```
> isinjectiveextended(A)
```

$$2 \qquad \textbf{(4.1.4)}$$

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

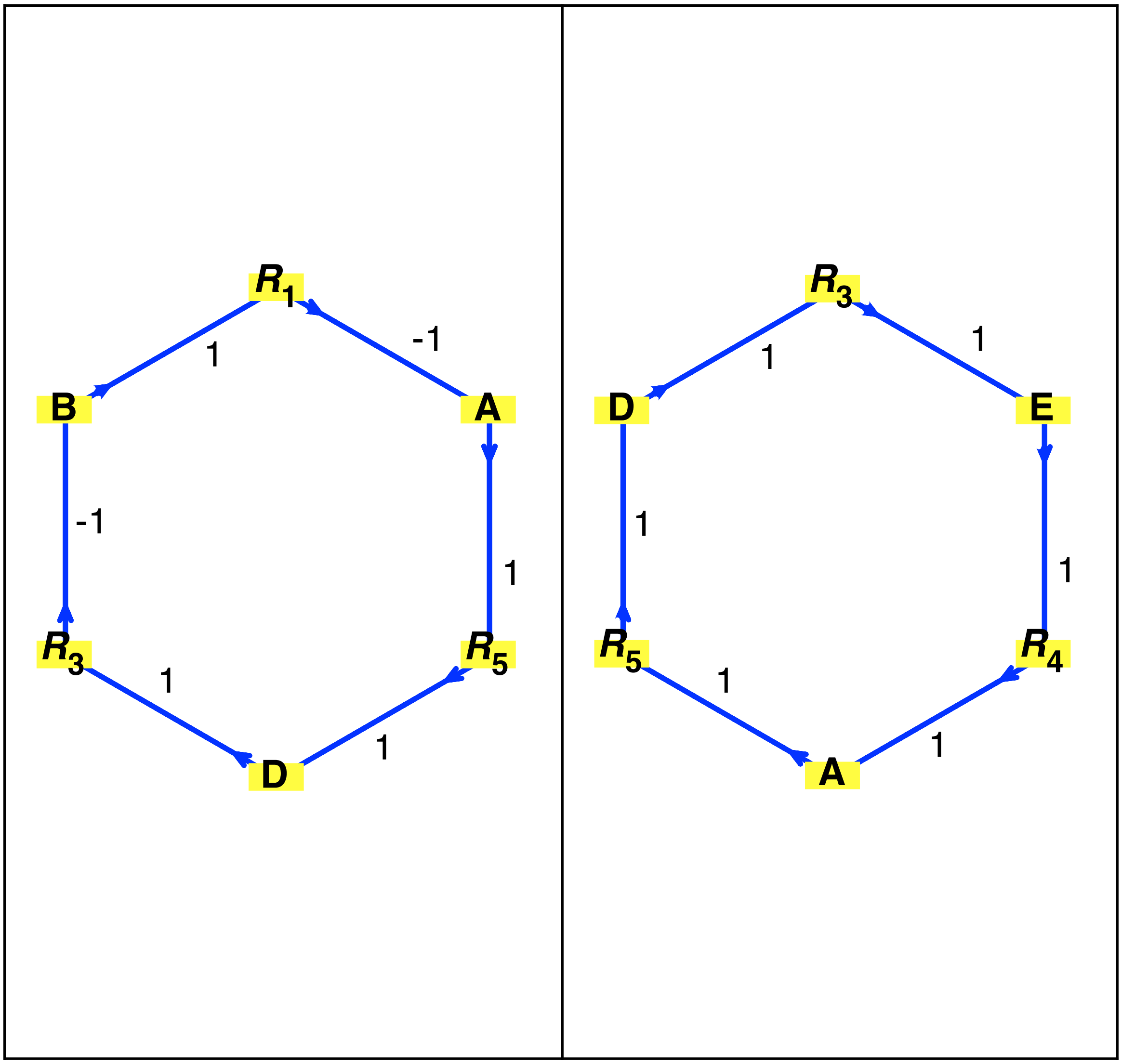
Step 3
For the networks that are multistationary, decide whether multistationarity can be attributed to competition, and find the positive feedback loops underlying multistationarity.

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

$$selected := [[[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]], [[[R_3, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], 1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1]]] \qquad (4.1.5)$$

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

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```



Network 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]]));
```

$$A := \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 \end{bmatrix} \qquad \textbf{(4.2.1)}$$

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)
```

$$\{[["A", R_1], 1], [["A", R_5], 1], [["B", R_1], 1], [["B", R_3], 1], [["C", R_2], 1], [["D", R_3], 1], [["F", R_7], 1], [["G", R_4], 1], [["G", R_6], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], -1], [[R_3, "D"], -1], [[R_3, "G"], 1], [[R_4, "D"], 1], [[R_4, "F"], 1], [[R_4, "G"], -1], [[R_5, "A"], -1], [[R_5, "D"], 1], [[R_6, "C"], 1], [[R_6, "G"], -1], [[R_7, "B"], 1], [[R_7, "F"], -1]\} \qquad \textbf{(4.2.2)}$$

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

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

Step 2

Injectivity test:

```
> isinjective(A)
                                   0                                   (4.2.3)
```

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

```
> isinjectiveextended(A)
```

$$0 \qquad \textbf{(4.2.4)}$$

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

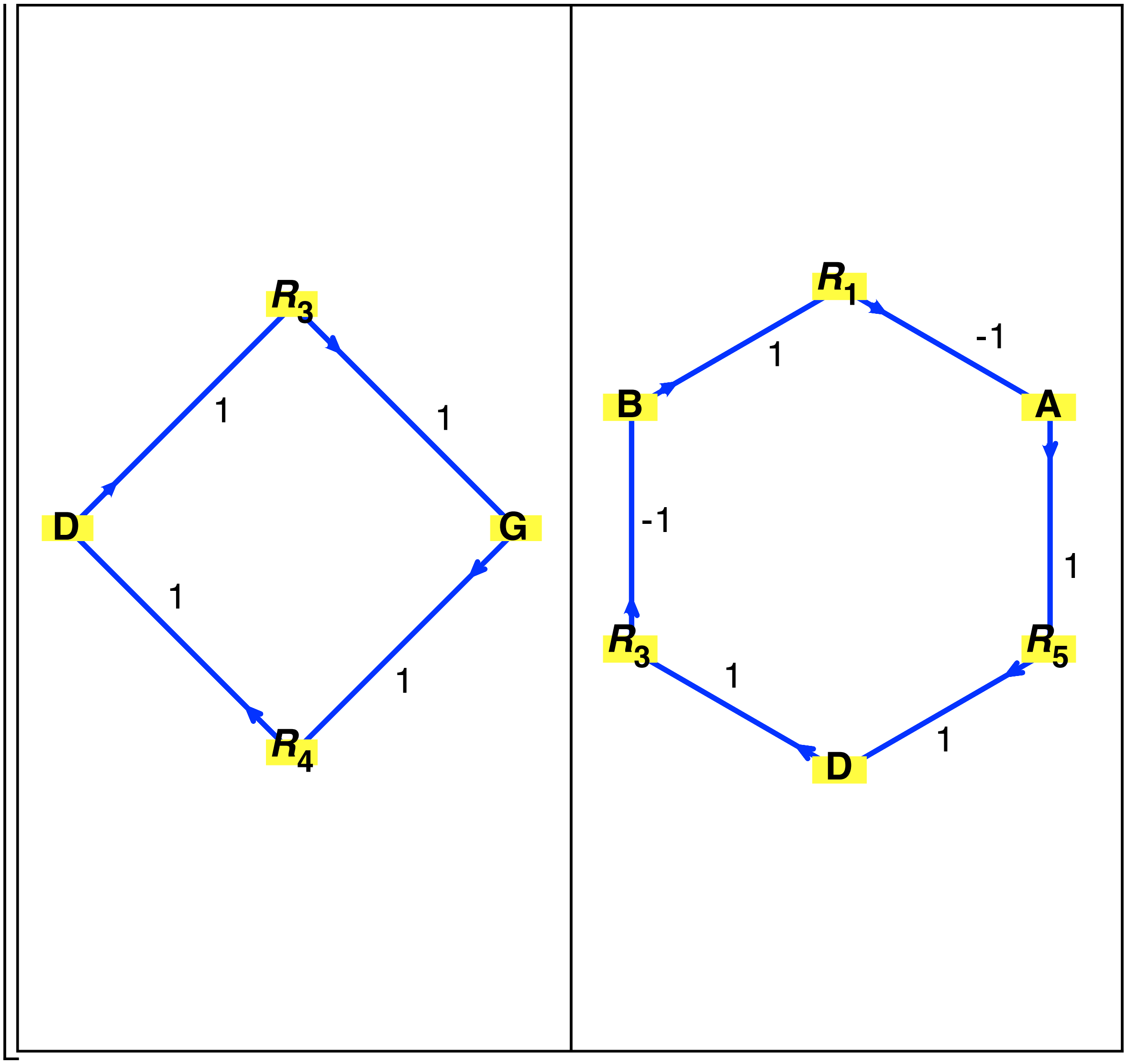
Step 3
We find the positive feedback loops.

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

$$selected := [[[[R_3, S_6], 1], [[S_6, R_4], 1], [[R_4, S_4], 1], [[S_4, R_3], 1]], [[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]]] \qquad \textbf{(4.2.5)}$$

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)
```



Network 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]]));
```

(4.3.1)

$$A := \begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \tag{4.3.1}$$

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

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

$$\{[["A", R_1], 1], [["A", R_5], 1], [["B", R_1], 1], [["B", R_3], 1], [["C", R_2], 1], [["D", R_3], 1], [["E", R_4], 1], [["F", R_6], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], -1], [[R_3, "D"], -1], [[R_3, "E"], 1], [[R_4, "A"], 1], [[R_4, "E"], -1], [[R_4, "F"], 1], [[R_5, "A"], -1], [[R_5, "D"], 1], [[R_6, "B"], 1], [[R_6, "F"], -1]\} \tag{4.3.2}$$

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

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

Step 2

Injectivity test:

```
> isinjective(A)
```

$$0 \tag{4.3.3}$$

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

```
> isinjectiveextended(A)
```

$$2 \tag{4.3.4}$$

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

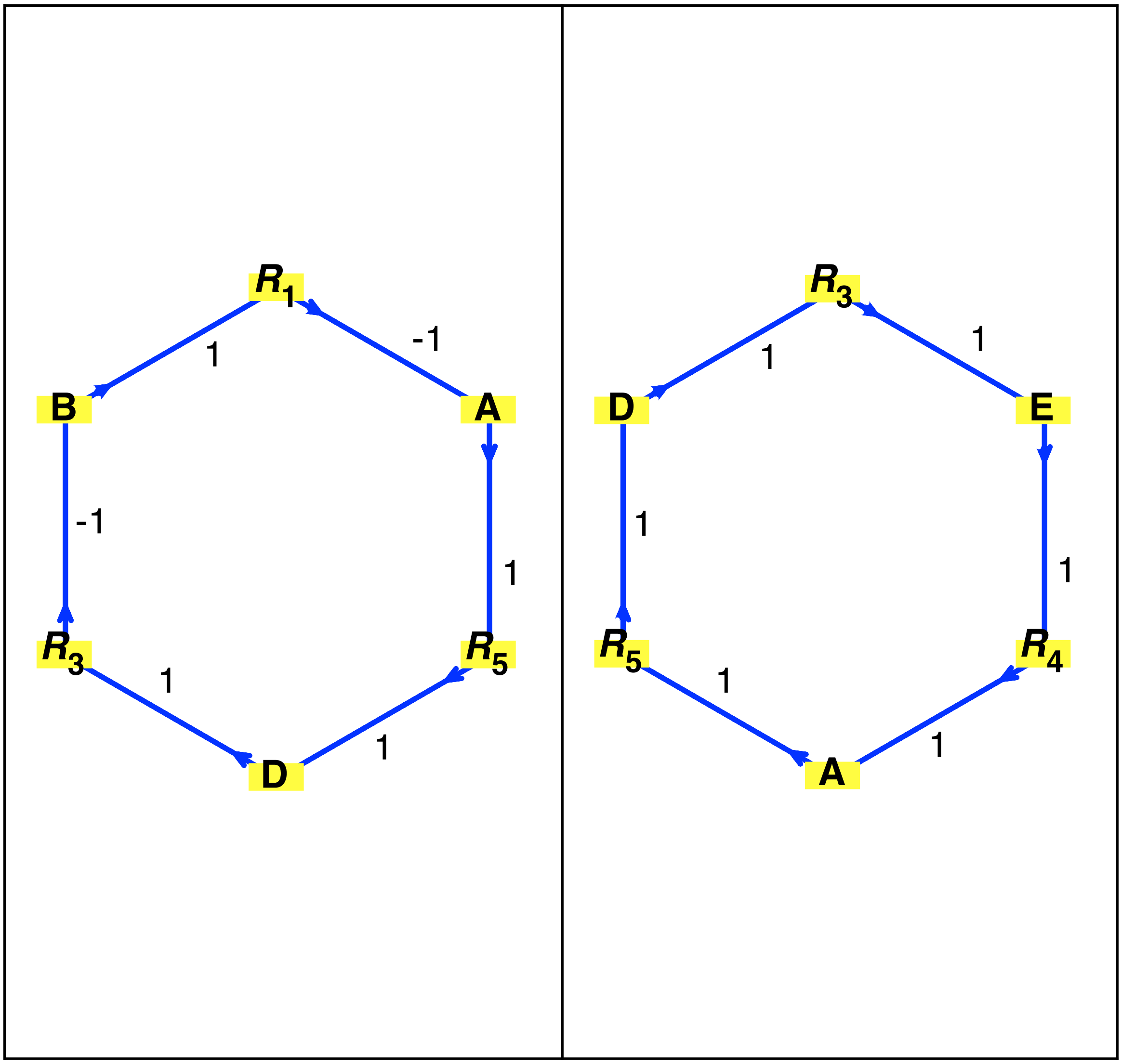
Step 3
For the networks that are multistationary, decide whether multistationarity can be attributed to competition, and find the positive feedback loops underlying multistationarity.

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

$$selected := [[[[R_1, S_1], -1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1], [[R_3, S_2], -1], [[S_2, R_1], 1]], [[[R_3, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], 1], [[S_1, R_5], 1], [[R_5, S_4], 1], [[S_4, R_3], 1]]] \quad \textbf{(4.3.5)}$$

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

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```



Network 4

$A + B \rightleftharpoons C \rightarrow \mathrm{D} + A$

$A + E \rightleftharpoons F \rightarrow G + A$
$B \rightleftharpoons E$
$\mathrm{D} \rightleftharpoons G$
$C \rightleftharpoons F$
$\mathrm{D} \rightarrow B$
$G \rightarrow E$

The stoichiometric matrix A is:

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

$$A := \begin{bmatrix} -1 & 1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad \textbf{(4.4.1)}$$

```
>
```

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

```
> mylabels := ["A", "B", "C", "D", "E", "F", "G", seq(R[i], i = 1 ..14)]:
> findedgesDSR(mylabels, A)
```

{[["A", R_1], 1], [["A", R_4], 1], [["B", R_1], 1], [["B", R_7], 1], [["C", R_2], 1], [["C", R_3], 1], [["C", R_{11}], 1], [["D", R_9], 1], [["D", R_{13}], 1], [["E", R_4], 1], [["E", R_8], 1], [["F", R_5], 1], [["F", R_6], 1], [["F", R_{12}], 1], [["G", R_{10}], 1], [["G", R_{14}], 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, "A"], 1], [[R_3, "C"], -1], [[R_3, "D"], 1], [[R_4, "A"], -1], [[R_4, "E"], -1], [[R_4, "F"], 1], [[R_5, "A"], 1], [[R_5, "E"], 1], [[R_5, "F"], -1], [[R_6, "A"], 1], [[R_6, "F"], -1], [[R_6, "G"], 1], [[R_7, "B"], -1], [[R_7, "E"], 1], [[R_8, "B"], 1], [[R_8, "E"], -1], [[R_9, "D"], -1], [[R_9, "G"], 1], [[R_{10}, "D"], 1], [[R_{10}, "G"], -1], [[R_{11}, "C"], -1], [[R_{11}, "F"], 1], [[R_{12}, "C"], 1], [[R_{12}, "F"], -1], [[R_{13}, "B"], 1], [[R_{13}, "D"], -1], [[R_{14}, "E"], 1], [[R_{14}, "G"], -1]} **(4.4.2)**

```
>
```

check injectivity

```
> isinjective(A)
```

$$0 \qquad \textbf{(4.4.3)}$$

```
> isinjectiveextended(A)
```

$$0 \qquad \textbf{(4.4.4)}$$

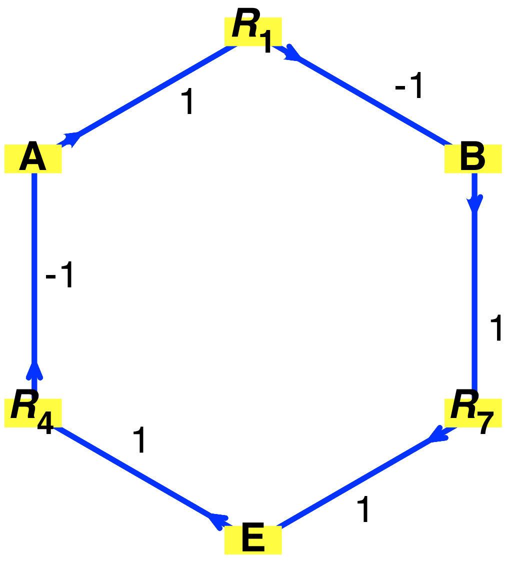
```
>
```

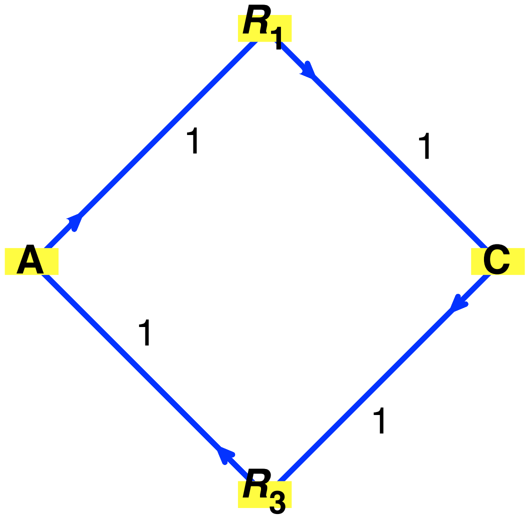
Draw the loops

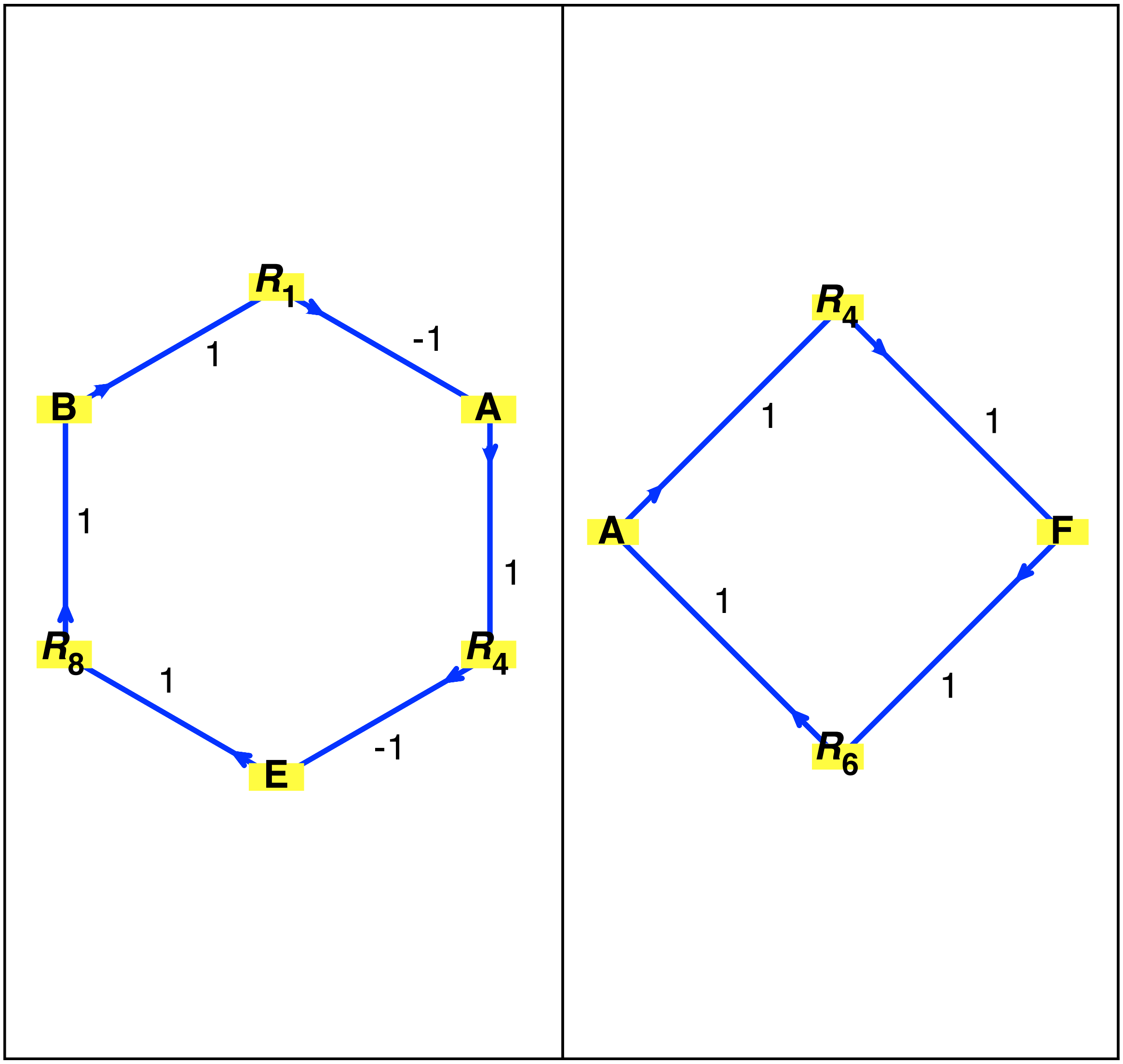
```
> Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :
> selected := findloops( )
```

$$selected := [[[[R_1, S_2], -1], [[S_2, R_7], 1], [[R_7, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], -1], [[S_1, R_1], 1]], [[[R_1, S_3], 1], [[S_3, R_3], 1], [[R_3, S_1], 1], [[S_1, R_1], 1]], [[[R_1, S_1], -1], [[S_1, R_4], 1], [[R_4, S_5], -1], [[S_5, R_8], 1], [[R_8, S_2], 1], [[S_2, R_1], 1]], [[[R_4, S_6], 1], [[S_6, R_6], 1], [[R_6, S_1], 1], [[S_1, R_4], 1]]] \qquad \textbf{(4.4.5)}$$

```
> speciesord := ["A", "B", "C", "D", "E", "F", "G"] :
> drawloops(selected, speciesord)
```


R1
A
B
1
-1
-1
1
R4
R7
1
1
E


R1
1
1
A
C
1
1
R3



```
>
```

Network 5

$A + B \rightleftharpoons C \rightarrow \mathrm{D} + B$
$A + E \rightleftharpoons F \rightarrow \mathrm{D} + E$
$B \rightleftharpoons E$
$C \rightleftharpoons F$
$\mathrm{D} \rightarrow A$

The stoichiometric matrix A is:

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

$$A := \begin{bmatrix} -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix} \qquad \textbf{(4.5.1)}$$

```
>
```

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 ..11)] :
> findedgesDSR(mylabels, A)
```

$$\{[[\text{"A"}, R_1], 1], [[\text{"A"}, R_4], 1], [[\text{"B"}, R_1], 1], [[\text{"B"}, R_7], 1], [[\text{"C"}, R_2], 1], [[\text{"C"}, R_3], 1], [[\text{"C"}, R_9], 1], [[\text{"D"}, R_{11}], 1], [[\text{"E"}, R_4], 1], [[\text{"E"}, R_8], 1], [[\text{"F"}, R_5], 1], [[\text{"F"}, R_6], 1], [[\text{"F"}, R_{10}], 1], [[R_1, \text{"A"}], -1], [[R_1, \text{"B"}], -1], [[R_1, \text{"C"}], 1], [[R_2, \text{"A"}], 1], [[R_2, \text{"B"}], 1], [[R_2, \text{"C"}], -1], [[R_3, \text{"B"}], 1], [[R_3, \text{"C"}], -1], [[R_3, \text{"D"}], 1], [[R_4, \text{"A"}], -1], [[R_4, \text{"E"}], -1], [[R_4, \text{"F"}], 1], [[R_5, \text{"A"}], 1], [[R_5, \text{"E"}], 1], [[R_5, \text{"F"}], -1], [[R_6, \text{"D"}], 1], [[R_6, \text{"E"}], 1], [[R_6, \text{"F"}], -1], [[R_7, \text{"B"}], -1], [[R_7, \text{"E"}], 1], [[R_8, \text{"B"}], 1], [[R_8, \text{"E"}], -1], [[R_9, \text{"C"}], -1], [[R_9, \text{"F"}], 1], [[R_{10}, \text{"C"}], 1], [[R_{10}, \text{"F"}], -1], [[R_{11}, \text{"A"}], 1], [[R_{11}, \text{"D"}], -1]\} \qquad \textbf{(4.5.2)}$$

```
>
```

check injectivity

```
> isinjective(A)
```

$$0 \qquad \textbf{(4.5.3)}$$

```
> isinjectiveextended(A)
```

$$0 \qquad \textbf{(4.5.4)}$$

```
>
```

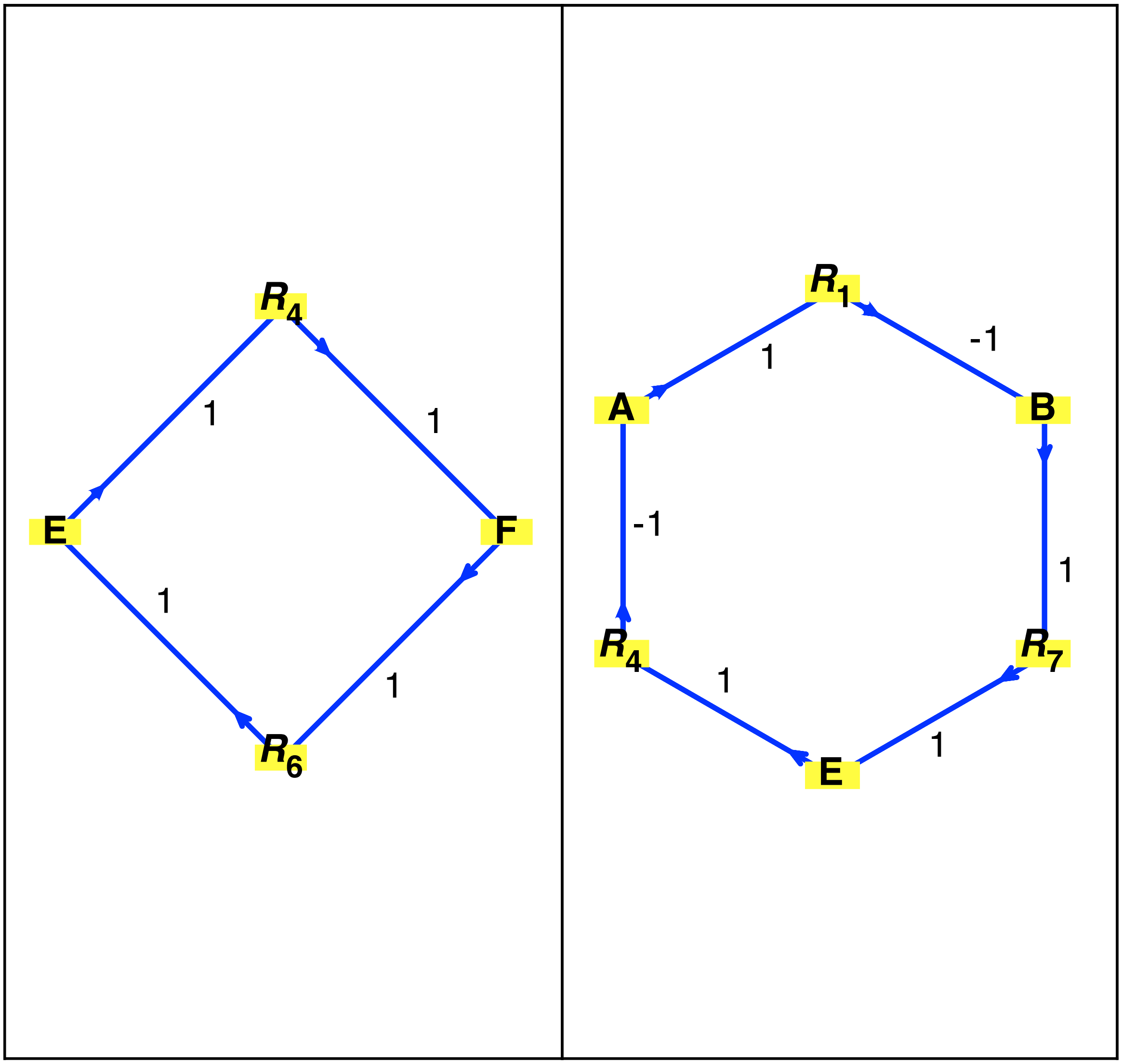
Draw the loops

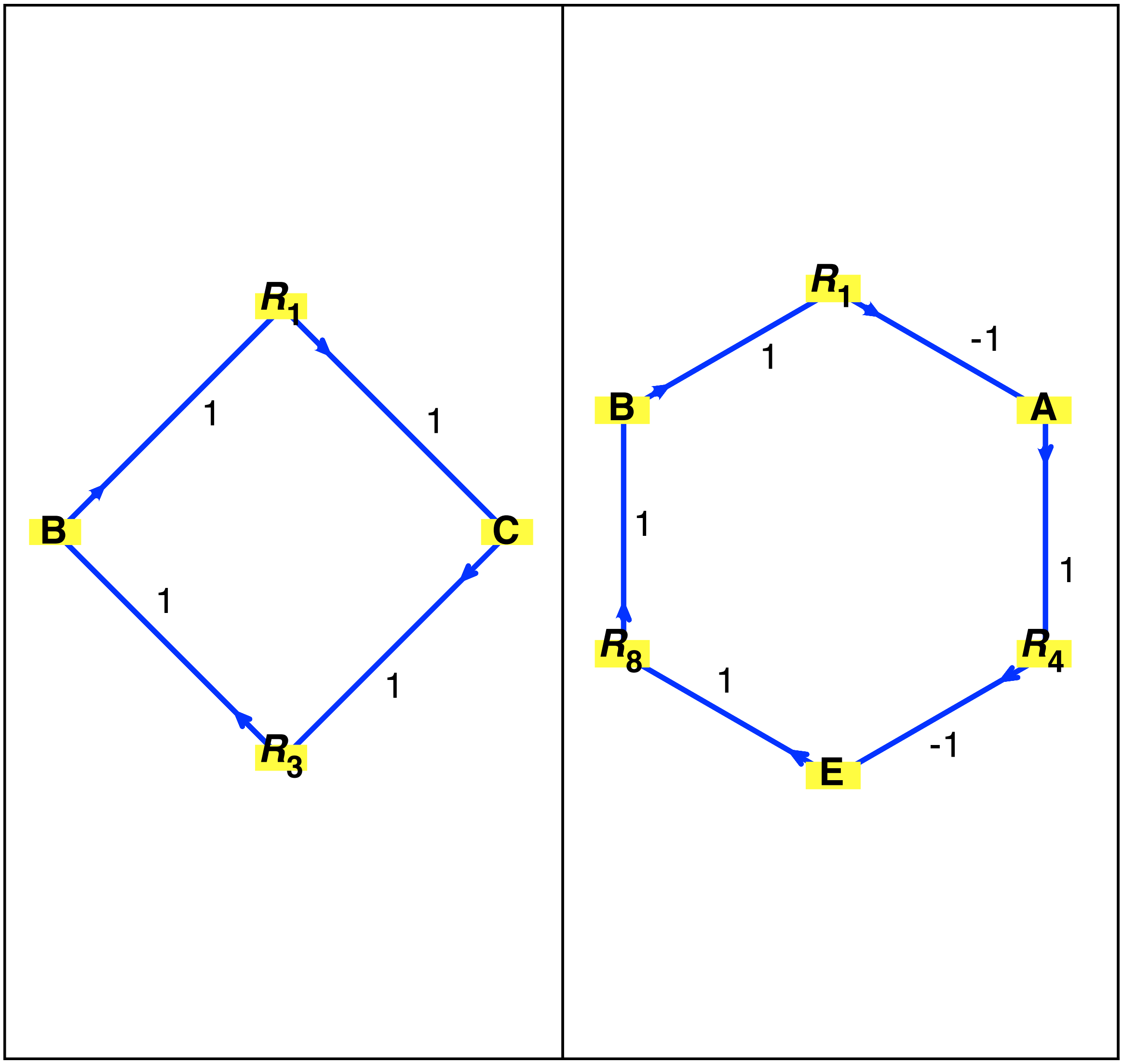
```
> Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) :
> selected := findloops( )
```

$$selected := [[[[R_4, S_6], 1], [[S_6, R_6], 1], [[R_6, S_5], 1], [[S_5, R_4], 1]], [[[R_1, S_2], -1], [[S_2, R_7], 1], [[R_7, S_5], 1], [[S_5, R_4], 1], [[R_4, S_1], -1], [[S_1, R_1], 1]], [[[R_1, \qquad \textbf{(4.5.5)}$$

$S_3], 1], [[S_3, R_3], 1], [[R_3, S_2], 1], [[S_2, R_1], 1]], [[[R_1, S_1], -1], [[S_1, R_4], 1],$
$[[R_4, S_5], -1], [[S_5, R_8], 1], [[R_8, S_2], 1], [[S_2, R_1], 1]]]$

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```





```
>
```

Network 6

$A + B \rightleftharpoons C \rightarrow \mathrm{D} + B$
$\mathrm{D} + B \rightleftharpoons E \rightarrow F + B$
$\mathrm{D} \rightarrow A$
$F \rightarrow \mathrm{D}$

The stoichiometric matrix A is:

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

$$A := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \tag{4.6.1}$$

```
>
```

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..8)] :
> findedgesDSR(mylabels, A)
```

$$\{[["A", R_1], 1], [["B", R_1], 1], [["B", R_4], 1], [["C", R_2], 1], [["C", R_3], 1], [["D", R_4], 1], [["D", R_7], 1], [["E", R_5], 1], [["E", R_6], 1], [["F", R_8], 1], [[R_1, "A"], -1], [[R_1, "B"], -1], [[R_1, "C"], 1], [[R_2, "A"], 1], [[R_2, "B"], 1], [[R_2, "C"], -1], [[R_3, "B"], 1], [[R_3, "C"], -1], [[R_3, "D"], 1], [[R_4, "B"], -1], [[R_4, "D"], -1], [[R_4, "E"], 1], [[R_5, "B"], 1], [[R_5, "D"], 1], [[R_5, "E"], -1], [[R_6, "B"], 1], [[R_6, "E"], -1], [[R_6, "F"], 1], [[R_7, "A"], 1], [[R_7, "D"], -1], [[R_8, "D"], 1], [[R_8, "F"], -1]\} \tag{4.6.2}$$

```
>
```

check injectivity

```
> isinjective(A)
```

$$0 \tag{4.6.3}$$

```
> isinjectiveextended(A)
```

$$2 \tag{4.6.4}$$

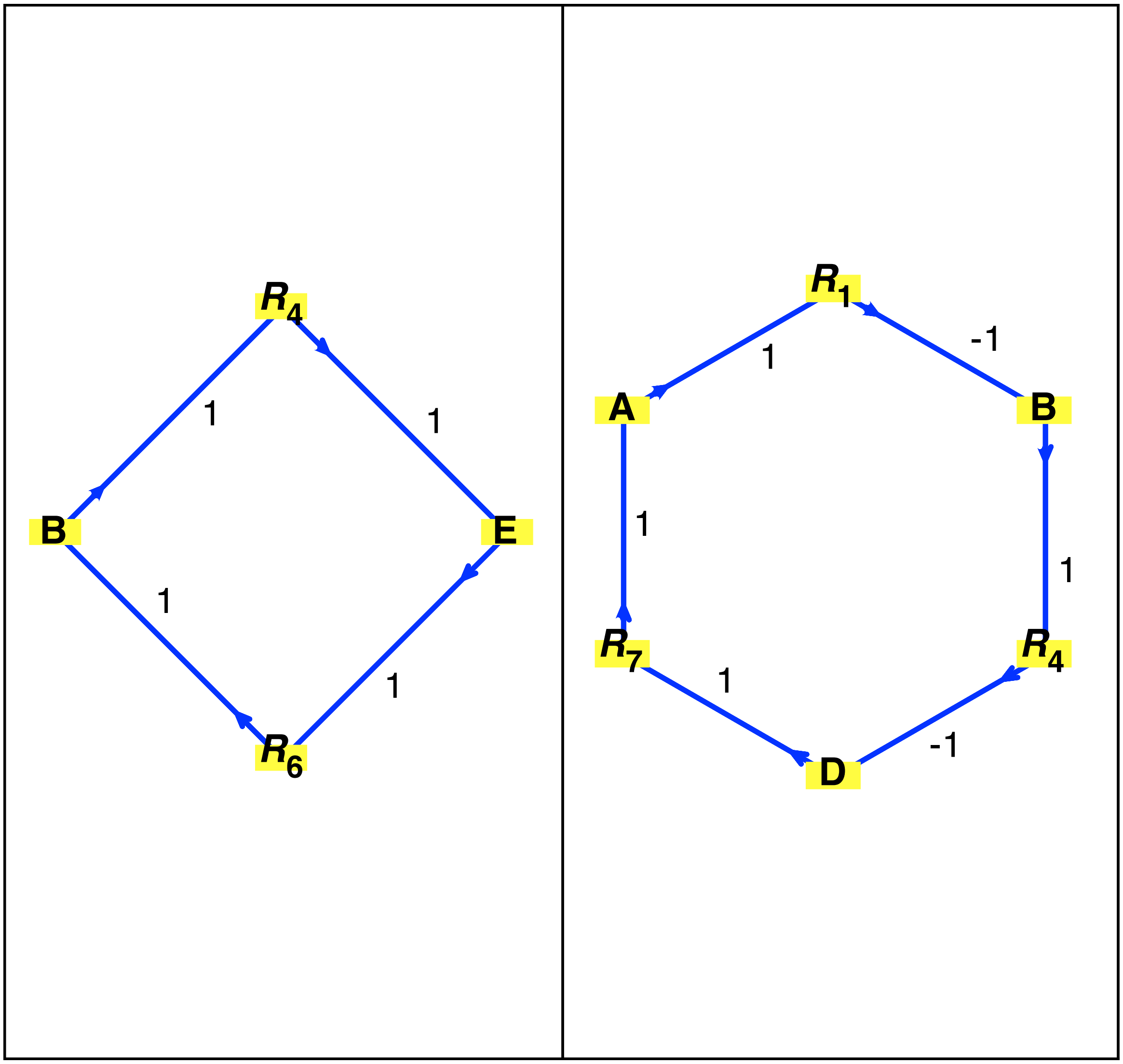
```
>
```

Draw the loops

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

$$selected := [[[[R_4, S_5], 1], [[S_5, R_6], 1], [[R_6, S_2], 1], [[S_2, R_4], 1]], [[[R_1, S_2], -1], [[S_2, R_4], 1], [[R_4, S_4], -1], [[S_4, R_7], 1], [[R_7, S_1], 1], [[S_1, R_1], 1]]] \tag{4.6.5}$$

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```



Network 7 (5 × 5 network No. 1741)

$B \rightarrow A$
$A + \mathrm{D} \rightarrow C$
$A + C \rightleftharpoons E$
$C \rightarrow B + \mathrm{D}$

The stoichiometric matrix A is:

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

(4.7.1)

$$A := \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \qquad \textbf{(4.7.1)}$$

```
>
```

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

```
> mylabels := ["A", "B", "C", "D", "E", seq(R[i], i = 1 ..5)] :
> findedgesDSR(mylabels, A)
```

$\{[["A", R_2], 1], [["A", R_3], 1], [["B", R_1], 1], [["C", R_3], 1], [["C", R_5], 1], [["D", R_2], 1], [["E", R_4], 1], [[R_1, "A"], 1], [[R_1, "B"], -1], [[R_2, "A"], -1], [[R_2, "C"], 1], [[R_2, "D"], -1], [[R_3, "A"], -1], [[R_3, "C"], -1], [[R_3, "E"], 1], [[R_4, "A"], 1], [[R_4, "C"], 1], [[R_4, "E"], -1], [[R_5, "B"], 1], [[R_5, "C"], -1], [[R_5, "D"], 1]\}$ **(4.7.2)**

```
>
```

check injectivity

```
> isinjective(A)
```

0 **(4.7.3)**

```
> isinjectiveextended(A)
```

2 **(4.7.4)**

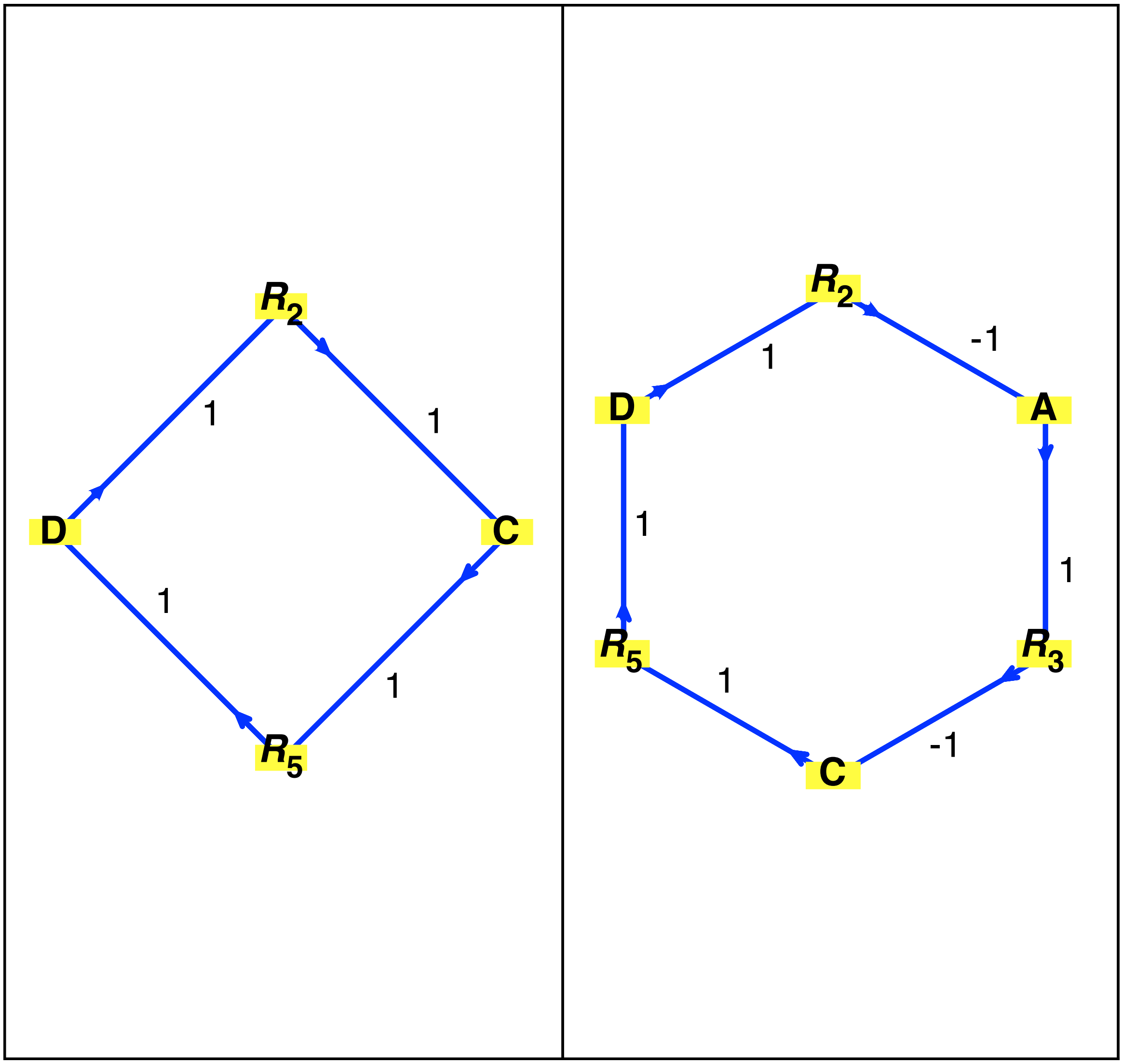
```
>
```

Draw the loops

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

$selected := [[[[R_2, S_3], 1], [[S_3, R_5], 1], [[R_5, S_4], 1], [[S_4, R_2], 1]], [[[R_2, S_1], -1], [[S_1, R_3], 1], [[R_3, S_3], -1], [[S_3, R_5], 1], [[R_5, S_4], 1], [[S_4, R_2], 1]]]$ **(4.7.5)**

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```



```
>
```

Network 8 (5 × 5 network No. 1742)

$B \rightarrow A$
$A + \mathrm{D} \rightarrow C$
$A + C \rightarrow E$
$C \rightarrow B + \mathrm{D}$
$E \rightarrow B + C$

The stoichiometric matrix A is:

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

$$A := \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \qquad \textbf{(4.8.1)}$$

```
>
```

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

```
> mylabels := ["A", "B", "C", "D", "E", seq(R[i], i = 1 ..5)] :
> findedgesDSR(mylabels, A)
```

$$\{[["A", R_2], 1], [["A", R_3], 1], [["B", R_1], 1], [["C", R_3], 1], [["C", R_4], 1], [["D", R_2], 1], [["E", R_5], 1], [[R_1, "A"], 1], [[R_1, "B"], -1], [[R_2, "A"], -1], [[R_2, "C"], 1], [[R_2, "D"], -1], [[R_3, "A"], -1], [[R_3, "C"], -1], [[R_3, "E"], 1], [[R_4, "B"], 1], [[R_4, "C"], -1], [[R_4, "D"], 1], [[R_5, "B"], 1], [[R_5, "C"], 1], [[R_5, "E"], -1]\} \qquad \textbf{(4.8.2)}$$

```
>
```

check injectivity

```
> isinjective(A)
```

$$0 \qquad \textbf{(4.8.3)}$$

```
> isinjectiveextended(A)
```

$$0 \qquad \textbf{(4.8.4)}$$

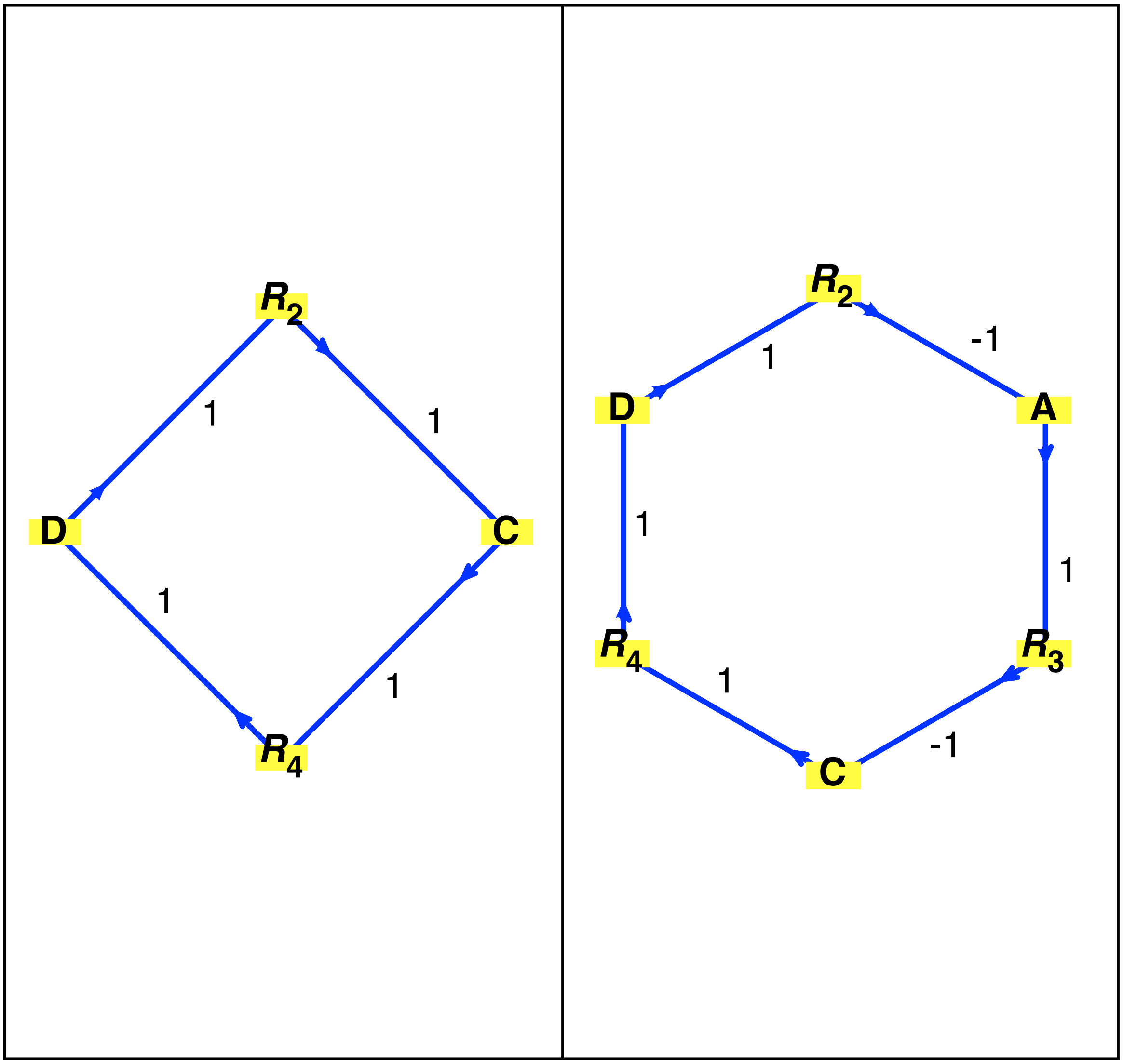
```
>
```

Draw the loops

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

$$selected := [[[[R_2, S_3], 1], [[S_3, R_4], 1], [[R_4, S_4], 1], [[S_4, R_2], 1]], [[[R_2, S_1], -1], [[S_1, R_3], 1], [[R_3, S_3], -1], [[S_3, R_4], 1], [[R_4, S_4], 1], [[S_4, R_2], 1]]] \qquad \textbf{(4.8.5)}$$

```
> speciesord := ["A", "B", "C", "D", "E", "F"] :
> drawloops(selected, speciesord)
```



>

Monostable networks

IMPORTNTANT: Network 2956, 2957, 2958 are similar to bistable networks 1741 & 1742. Need to further analyze their difference.

If we impose the condition strictly as "not in any of intersecting edges", we only have network 2958 meeting both

conditions (however, they are not realistic), network 2956 meeting condition 2 (not realistic), and network 1796, 1518, 2970, (however they are also not realistic)!!

5 × 5 networks with competing PF loop intersecting with other PF loops

```
> speciesord ≔ ["A", "B", "C", "D", "E"] :
> uniquefolder ≔ "5species/nonmultistationary/unique_competitionloop_intersectingloops" :
>
```

Network 2962 (meet neither, not realistic)

$B + C \rightarrow A$

$C + \mathrm{D} \rightarrow B$

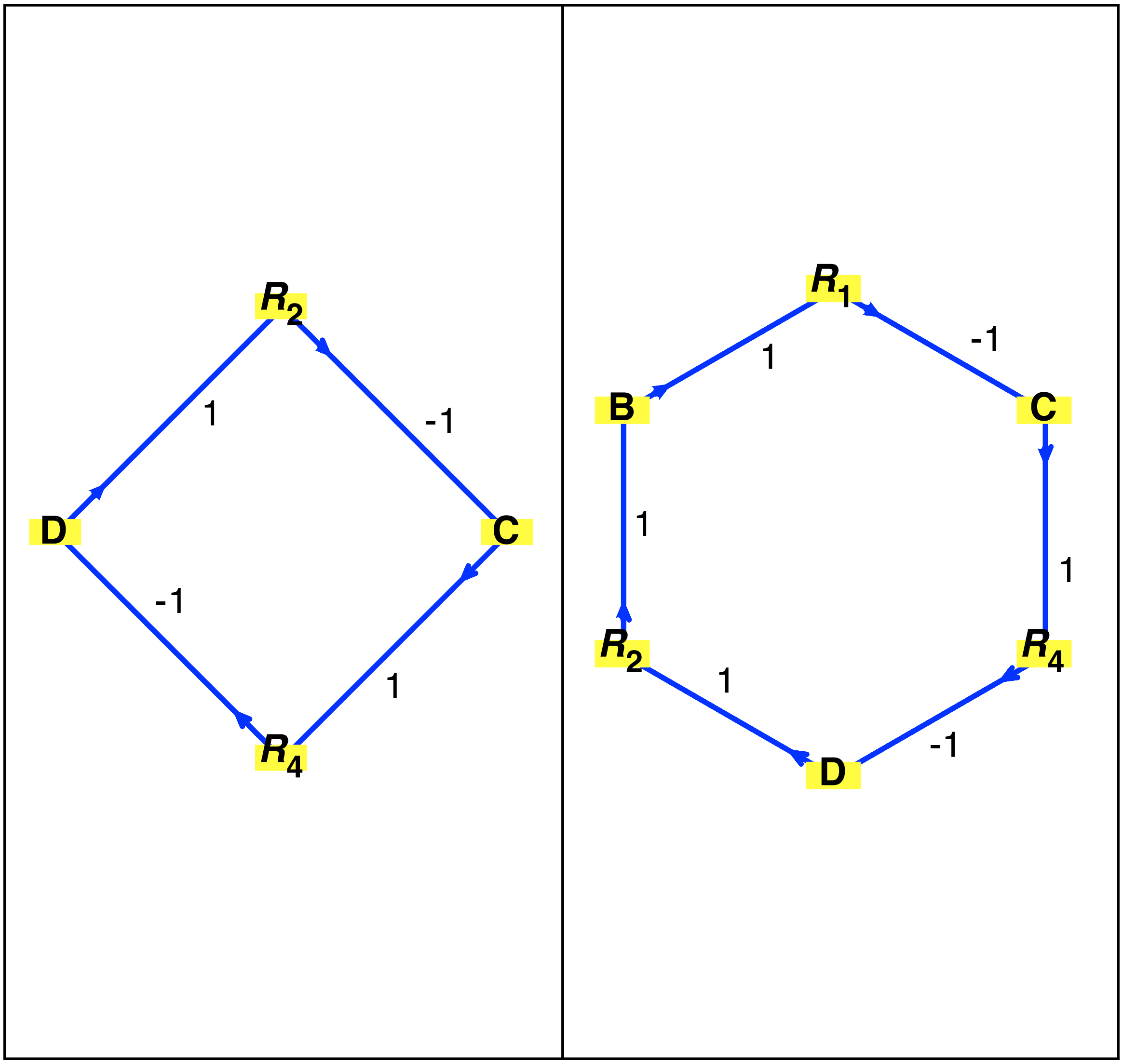
$A \rightarrow B + C$

$C + \mathrm{D} \rightarrow E$

$E \rightarrow C + \mathrm{D}$

```
> A ≔ ImportMatrix(cat(uniquefolder, "/injective1_2962.csv")); Z ≔ findZ(A) : selected
        ≔ findloops(A, Z) : drawloops(selected, speciesord)
```

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

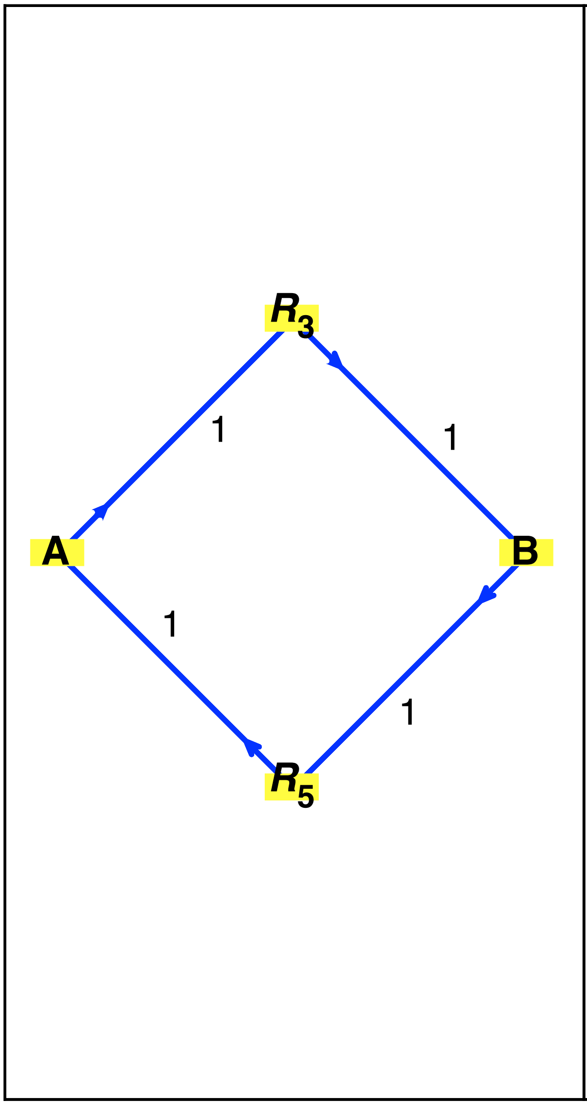


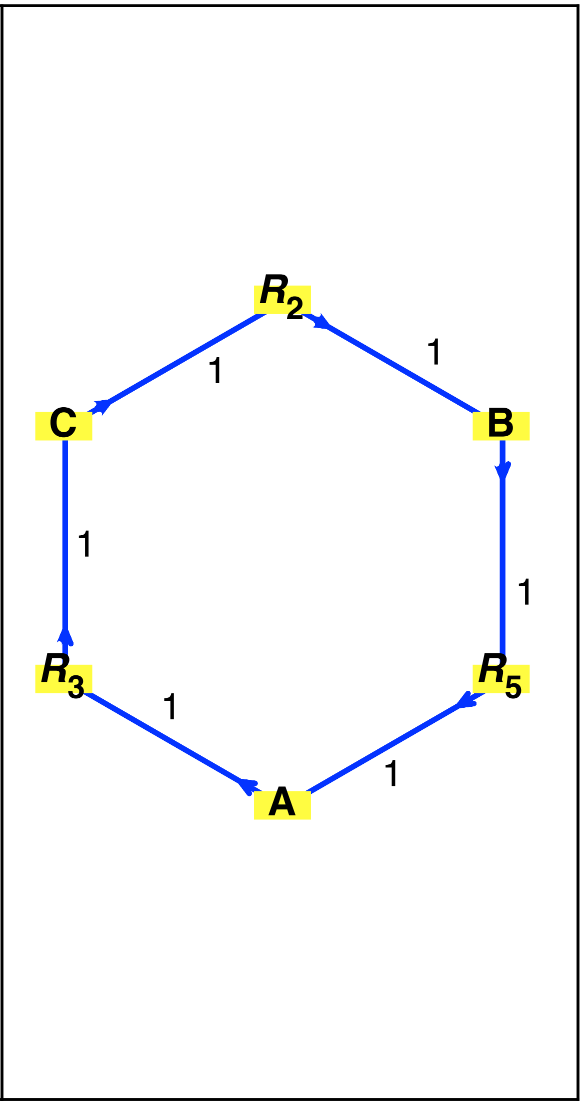
Network 2956 (meet condition 2, not realistic), three competing reactions

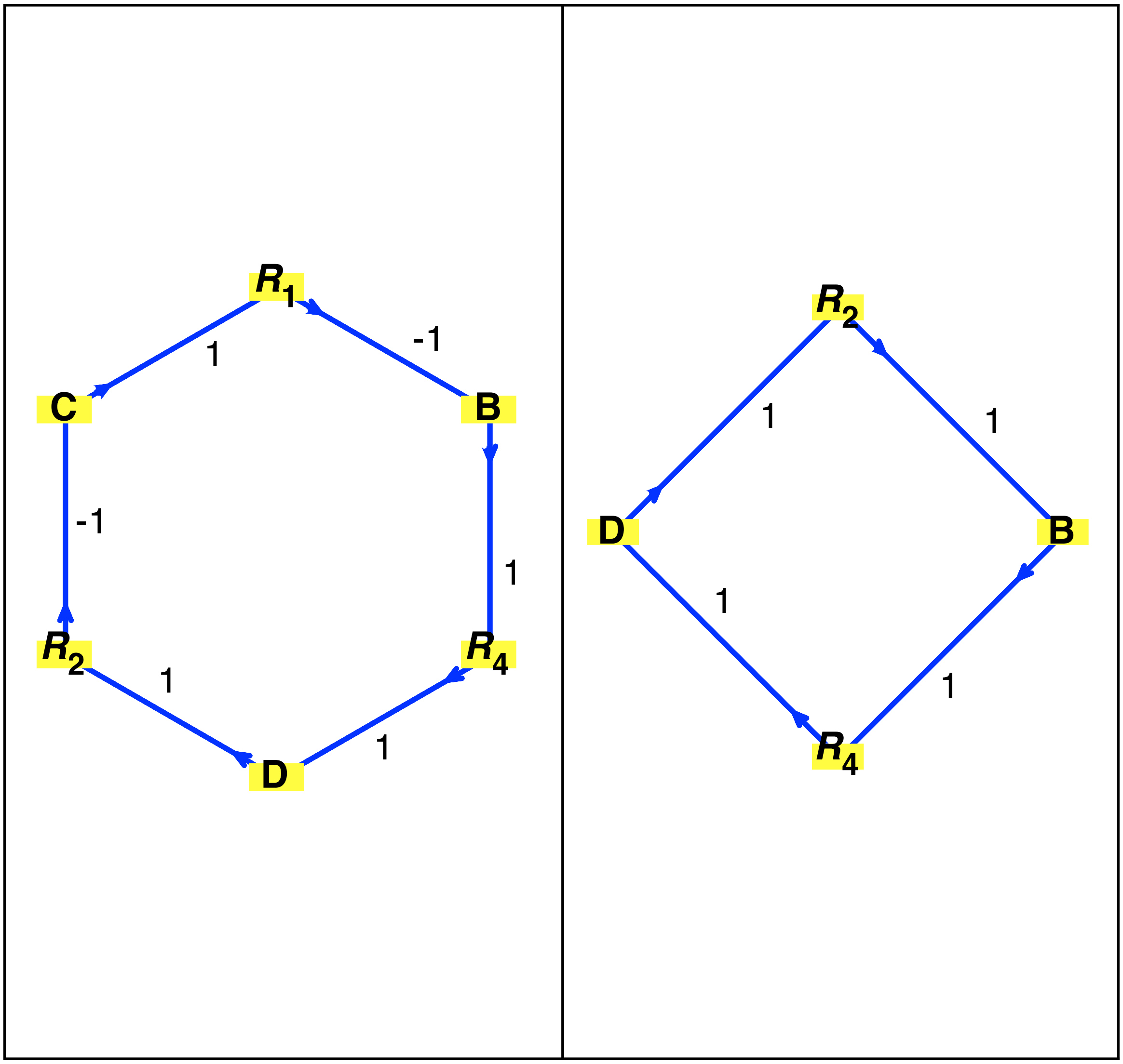
$B + C \rightarrow A$
$\mathrm{D} + C \rightarrow B$
$A \rightarrow B + C$
$B \rightarrow \mathrm{D} + E$
$E + B \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx1_2956.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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







>

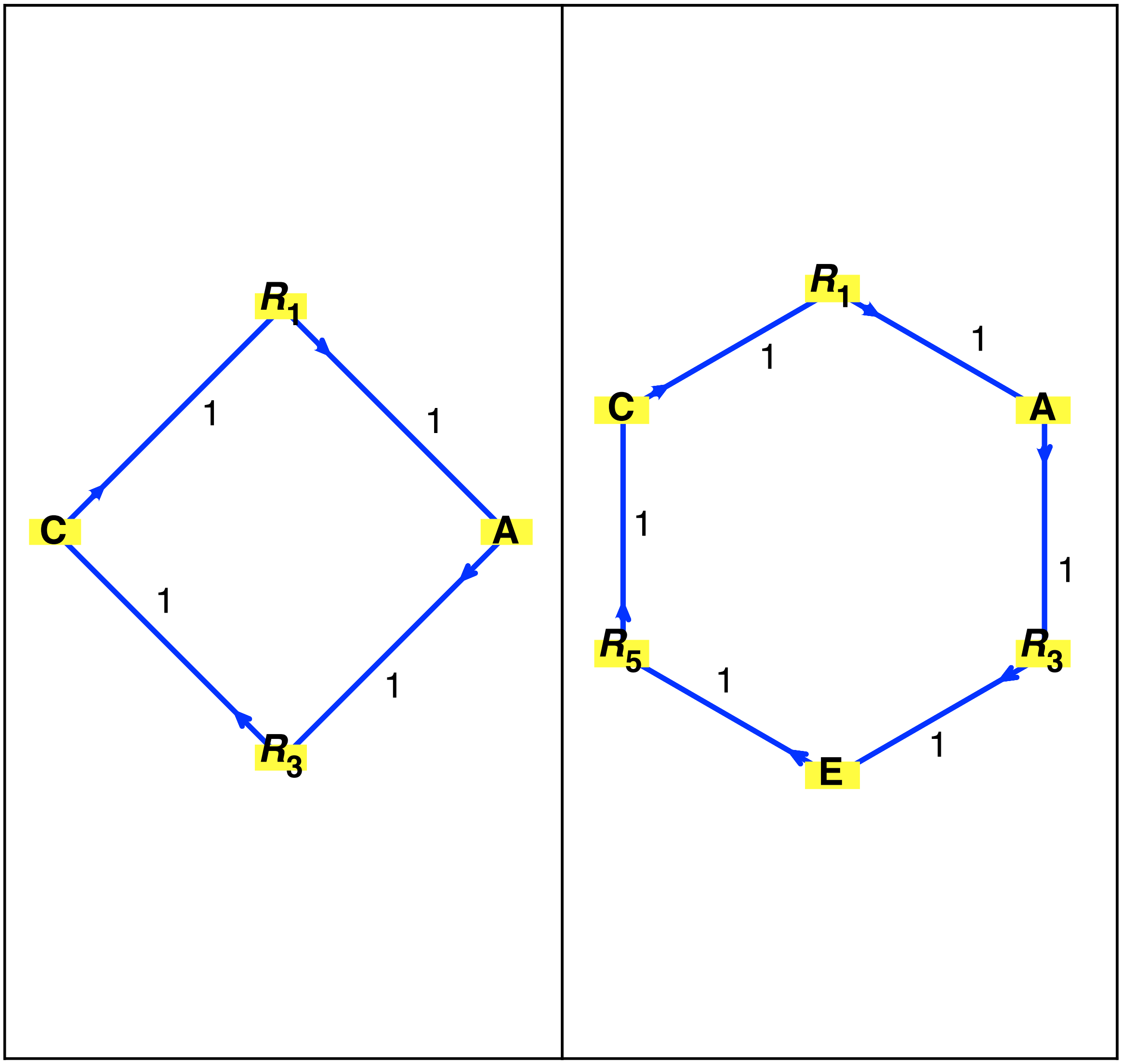
Meet condition 1 & 2. NB: E is same as C

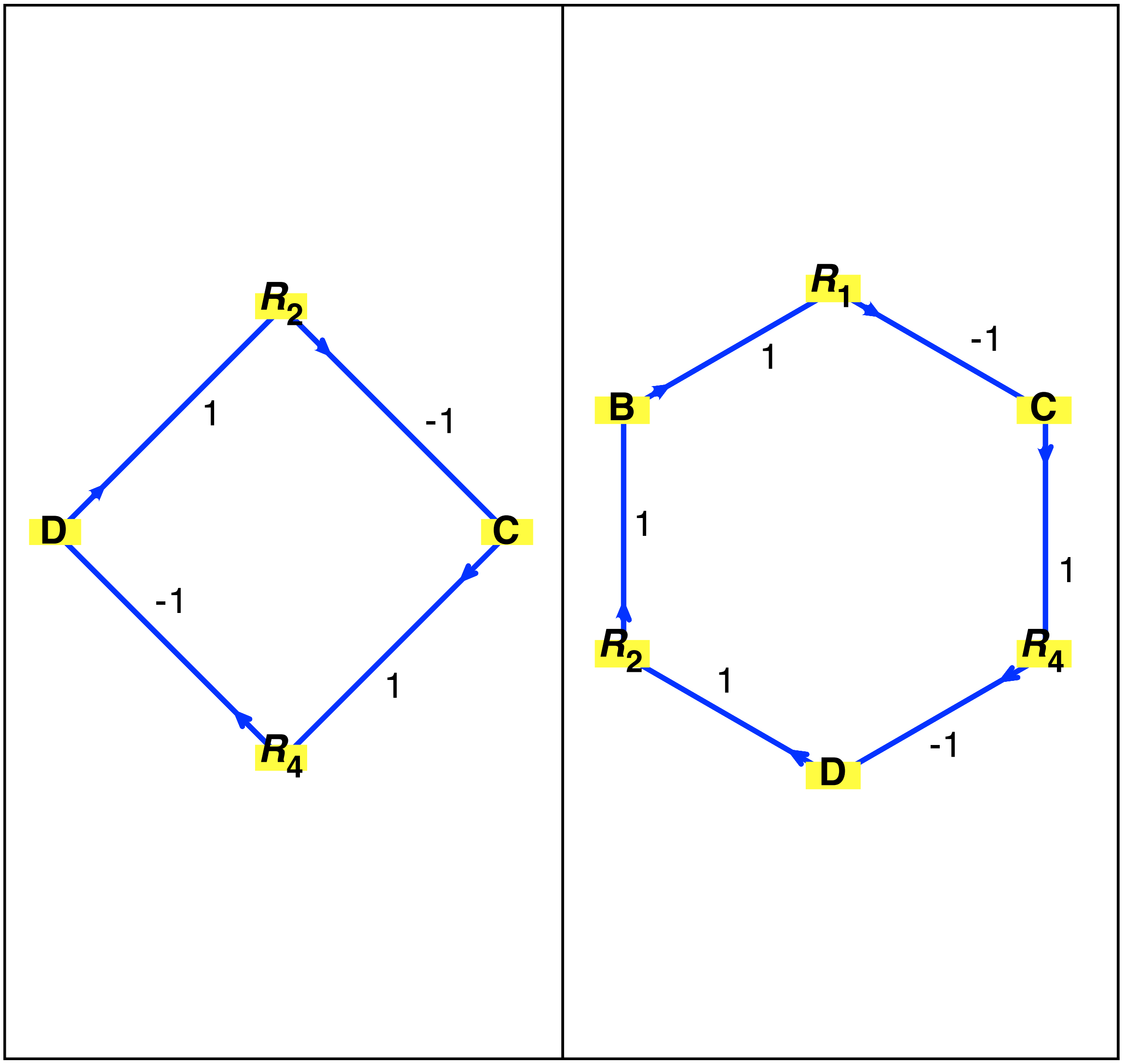
Network 2988 (meet neither, not realistic)

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow C + E$
$C + \mathrm{D} \rightleftharpoons E$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx1_2988.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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


R1
C
A
R3
1
1
1
1
R1
C
A
R5
R3
E
1
1
1
1
1
1



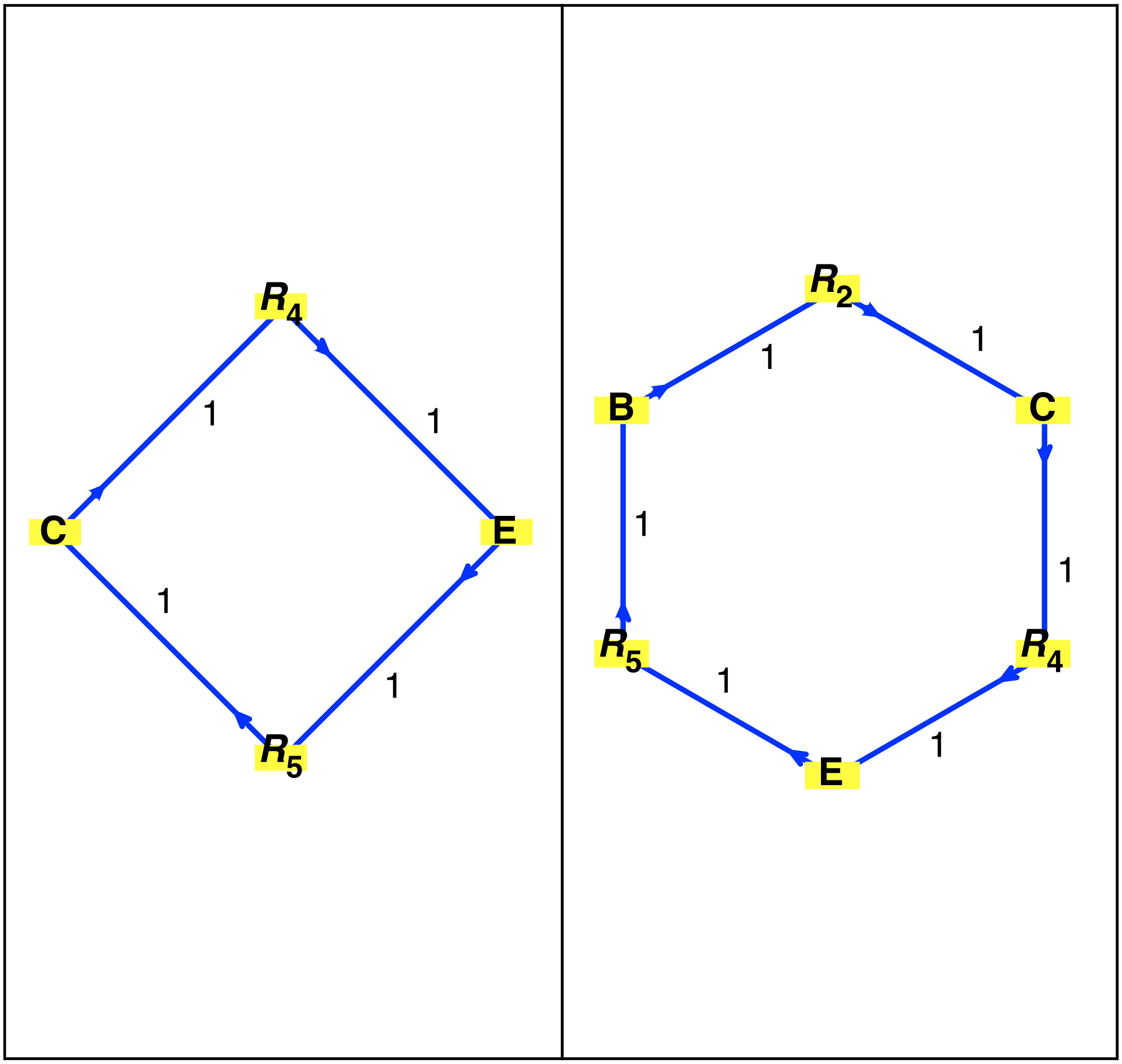
```
>
```

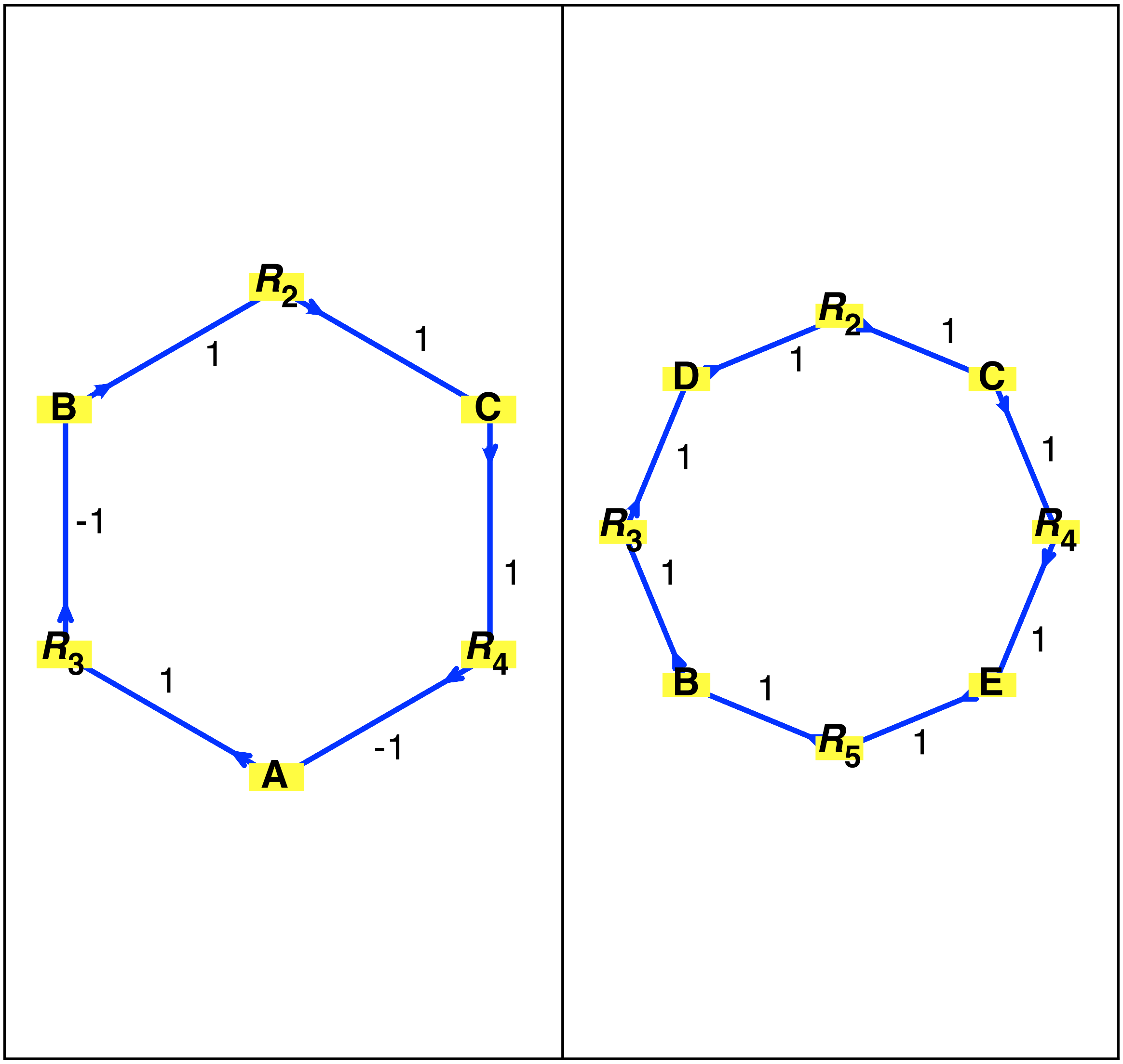
Network 1473 (meet neither, maybe not realistic), three competing reactions

$B \rightarrow A$
$B + \mathrm{D} \rightarrow C$
$A + B \rightarrow \mathrm{D}$
$A + C \rightarrow E$
$E \rightarrow B + C$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_1473.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

Meet both conditions (?). NB: E is tetramer, C is trimer, D is dimer

Network 1474 (meet neither, maybe not realistic), three competing reactions

$B \rightarrow A$

$B + \mathrm{D} \rightarrow C$

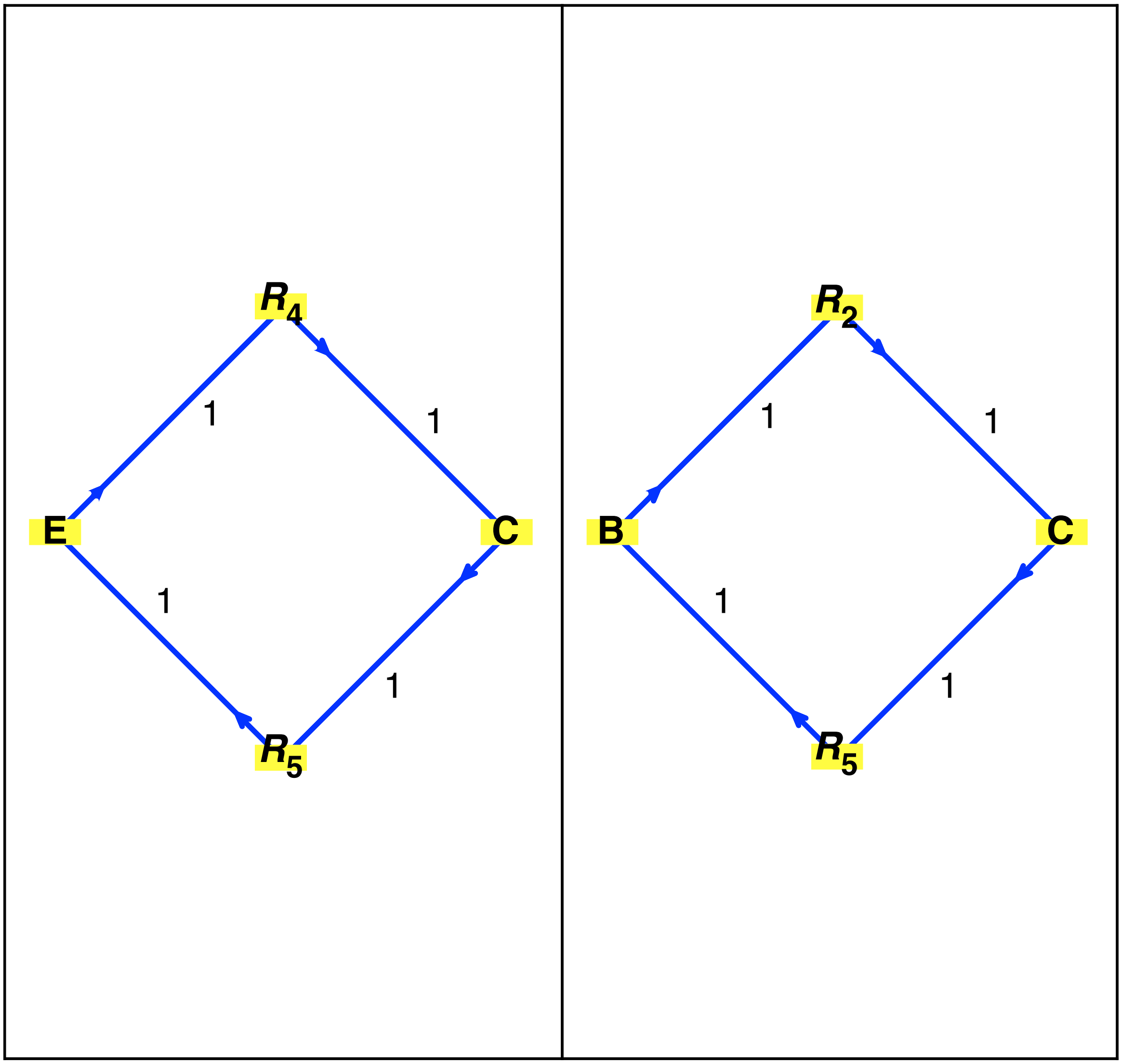
$A + B \rightarrow \mathrm{D}$

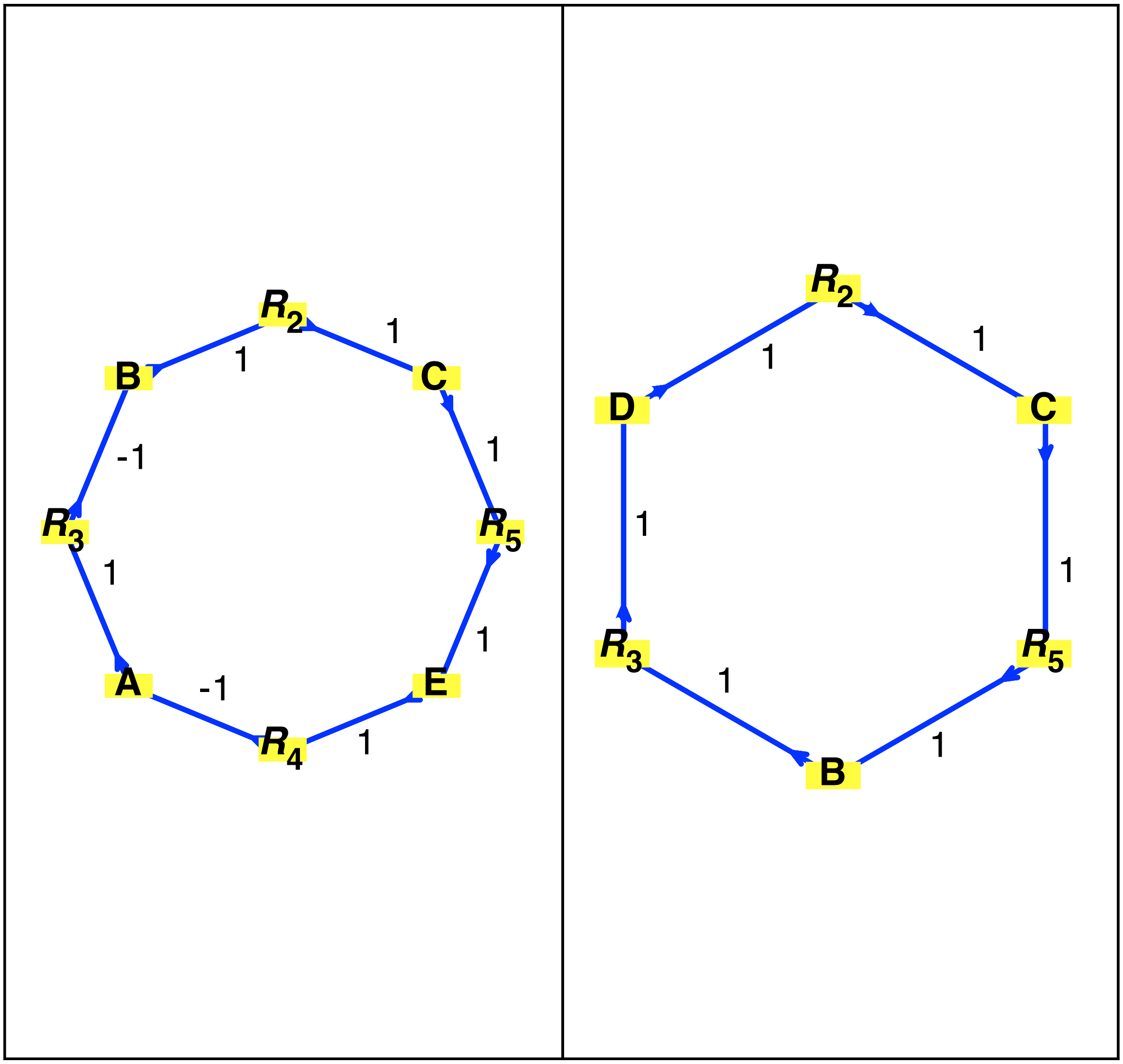
$A + E \rightarrow C$

$C \rightarrow B + E$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_1474.csv")); Z := findZ(A) :
    selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

Meet both (?). NB: C is trimer, D is dimer, E is dimer

Network 1518 (meat condition 1, not realistic)

$B \rightarrow A$

$B + \mathrm{D} \rightarrow C$

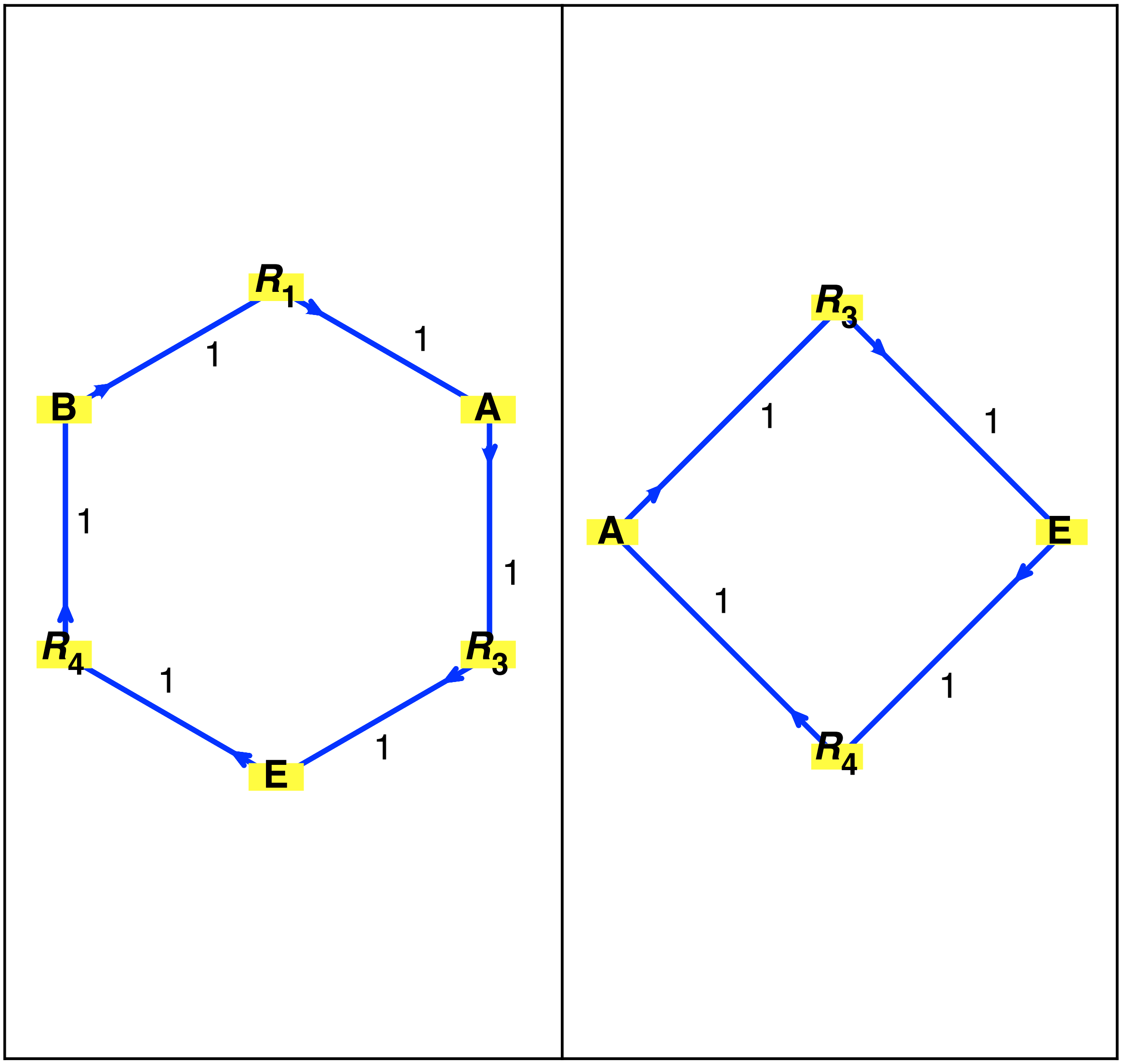
$A + \mathrm{D} \rightarrow E$

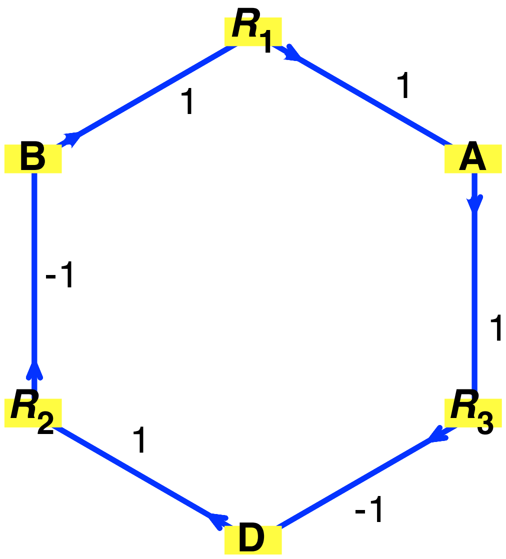
$E \rightarrow A + B$

$C \rightarrow B + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_1518.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

Meet both (?). NB: everything seems going to A

Network 1796 (meat condition 1, not realistic)

$B \rightarrow A$

$A + \mathrm{D} \rightarrow C$

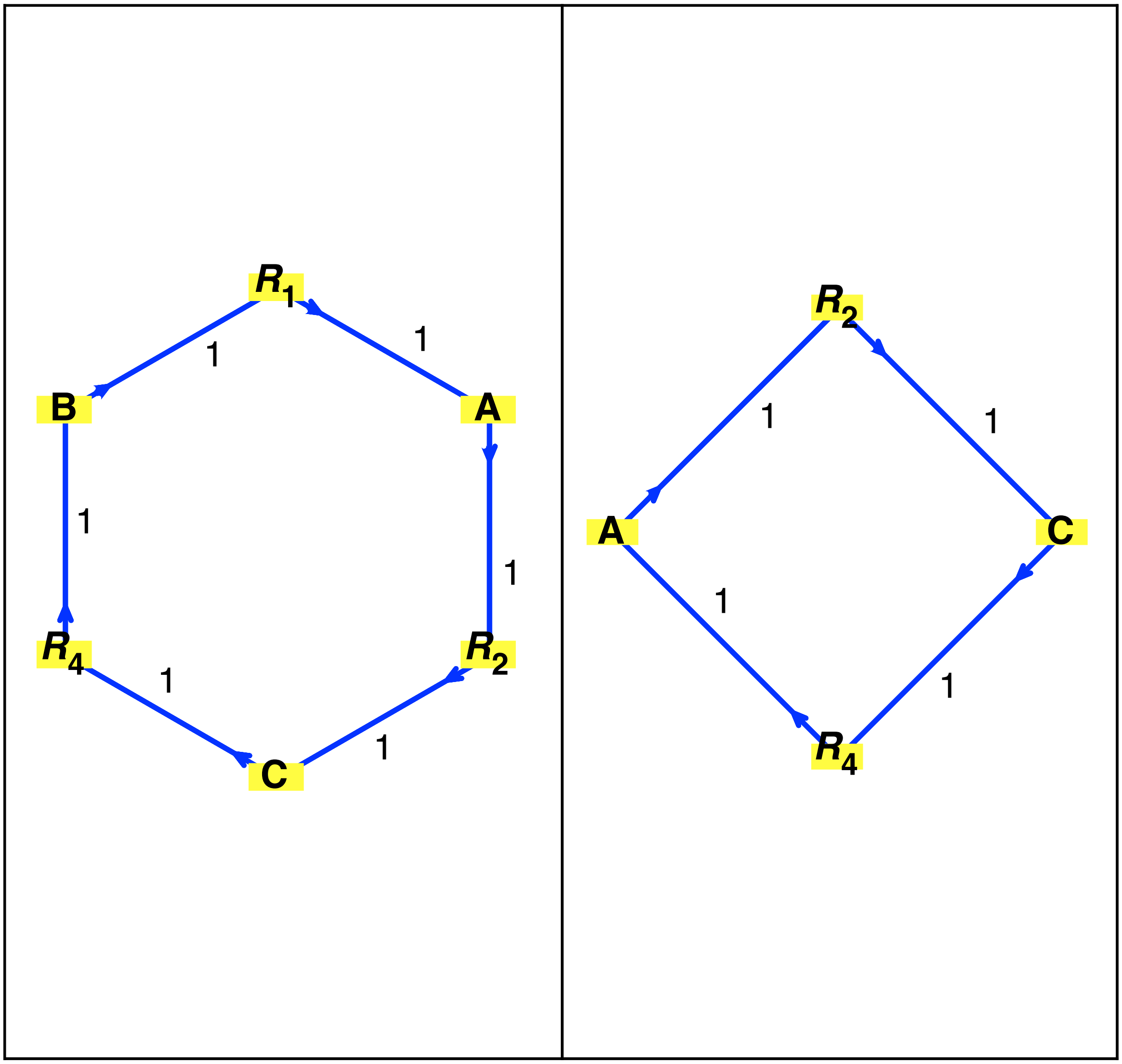
$B + \mathrm{D} \rightarrow E$

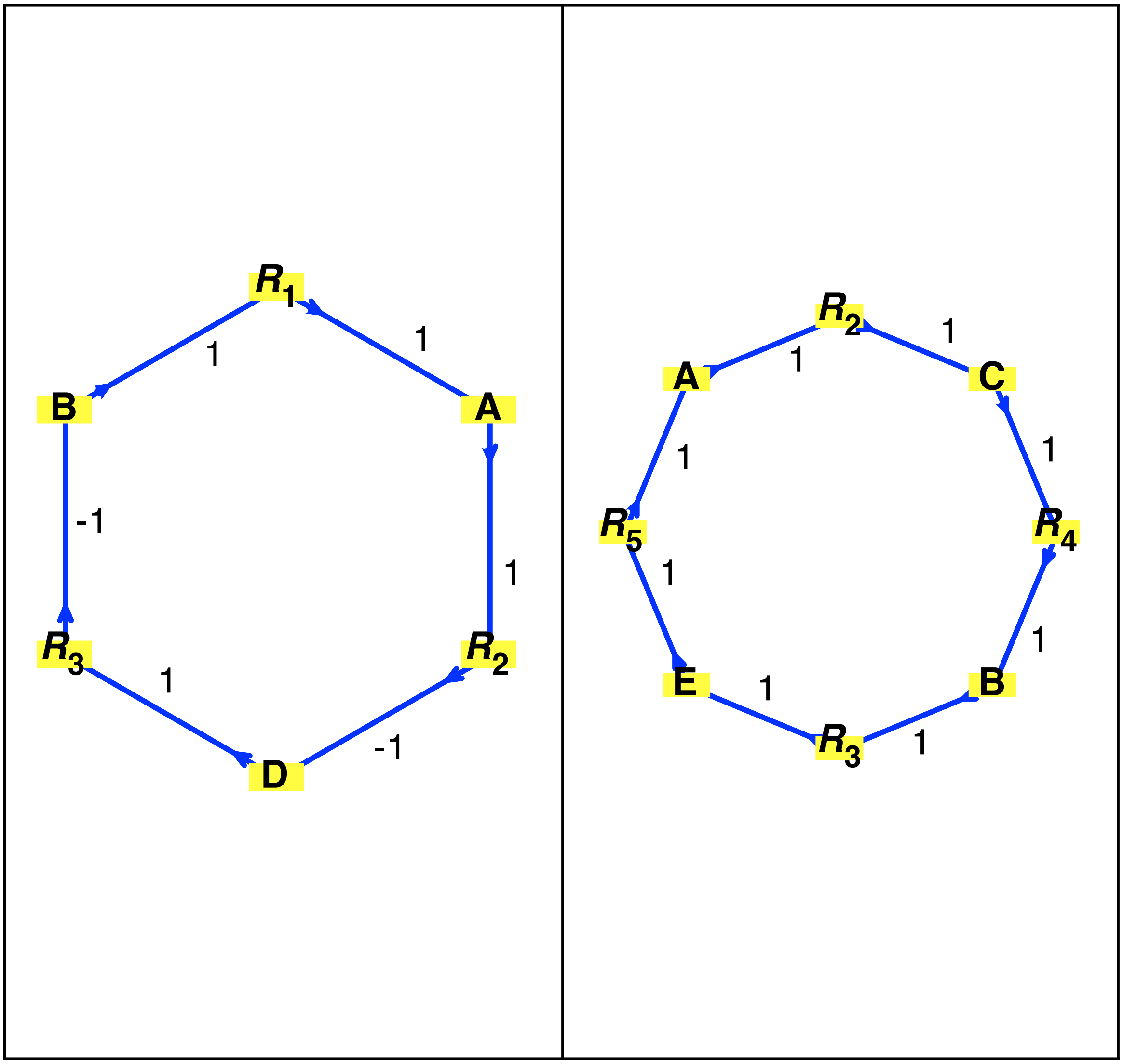
$C \rightarrow A + B$

$E \rightarrow A + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_1796.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

Meet condition 1. NB: A, B, D are three states, and all go to A.

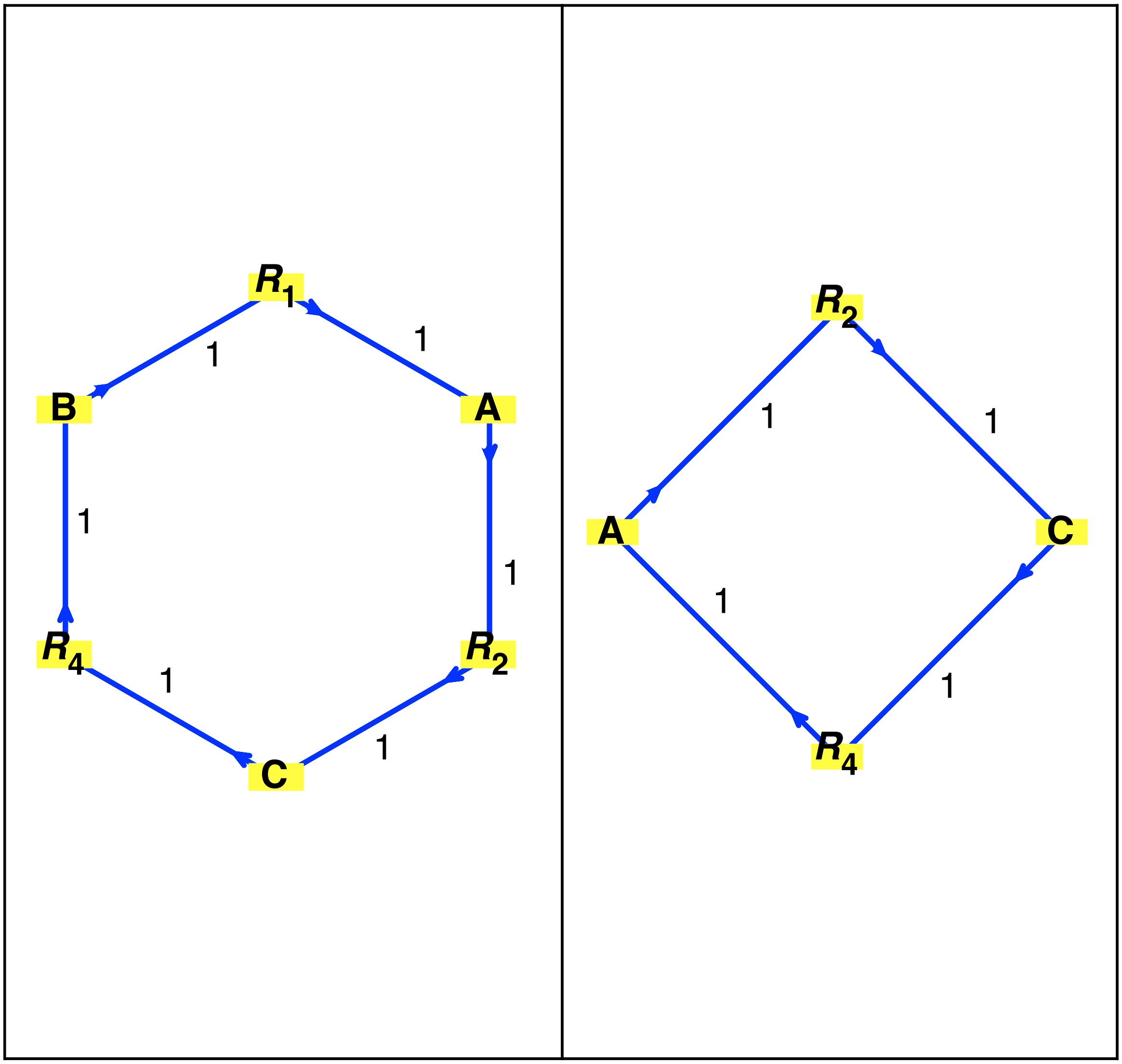
NB!!! all 1473, 1474, 1518, 1796 contains the same pattern: the three intersecting loops in 1518

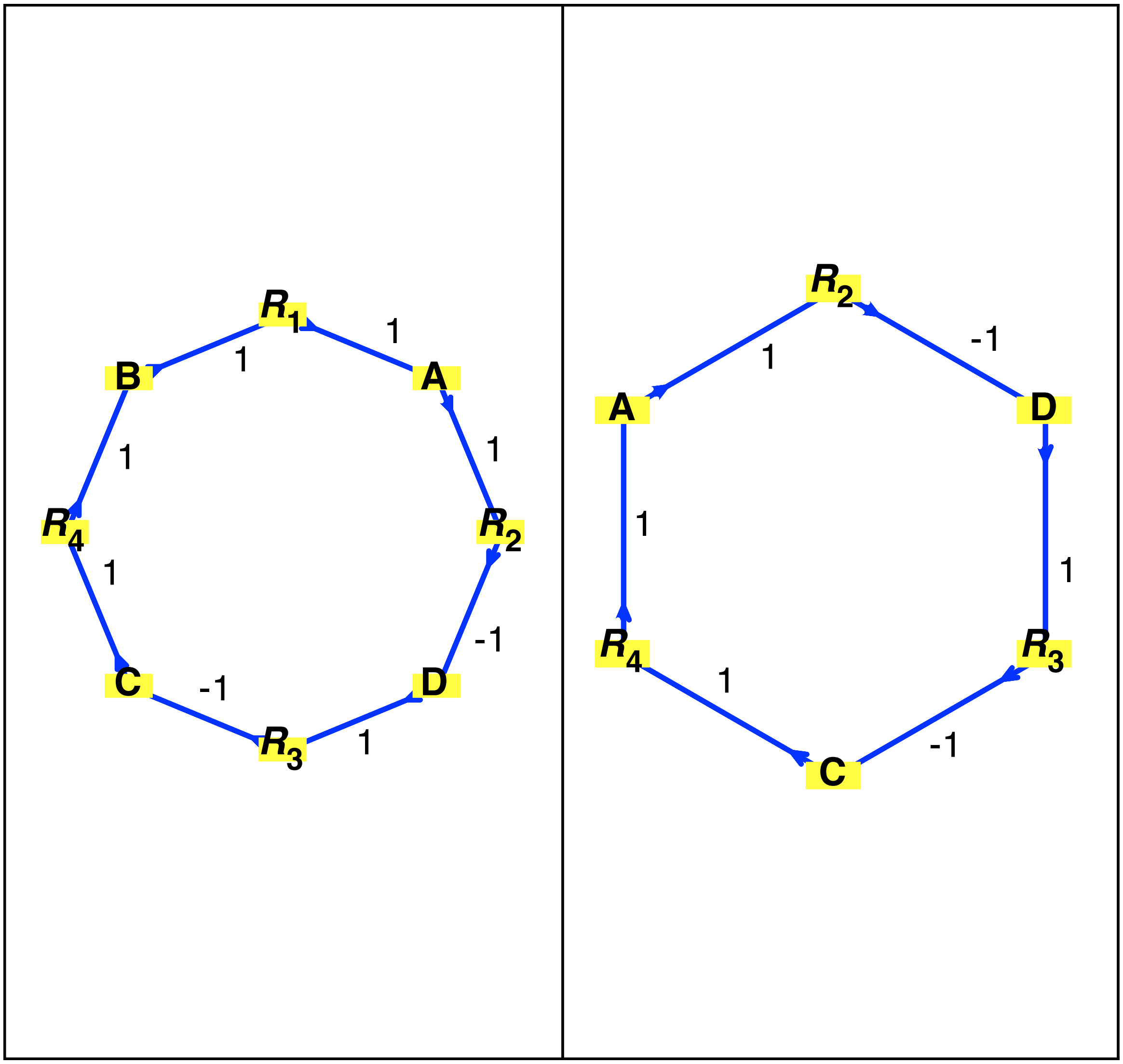
Network 1814 (meet neither, not realistic)

$B \rightarrow A$
$A + \mathrm{D} \rightarrow C$
$C + \mathrm{D} \rightarrow E$
$C \rightarrow A + B$
$E \rightarrow C + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_1814.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

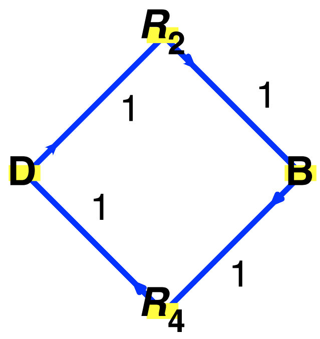
Meet both (?). NB: B, D, A are three states and all go to A.

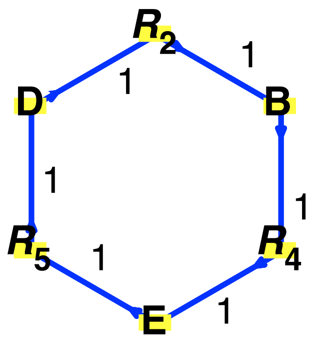
Network 2955 (meet neither, realistic but no futile cycle) three competing reactions

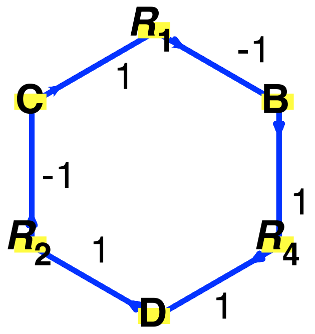
$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow B + C$
$B \rightarrow C + \mathrm{D}$
$C + E \rightarrow \mathrm{D}$

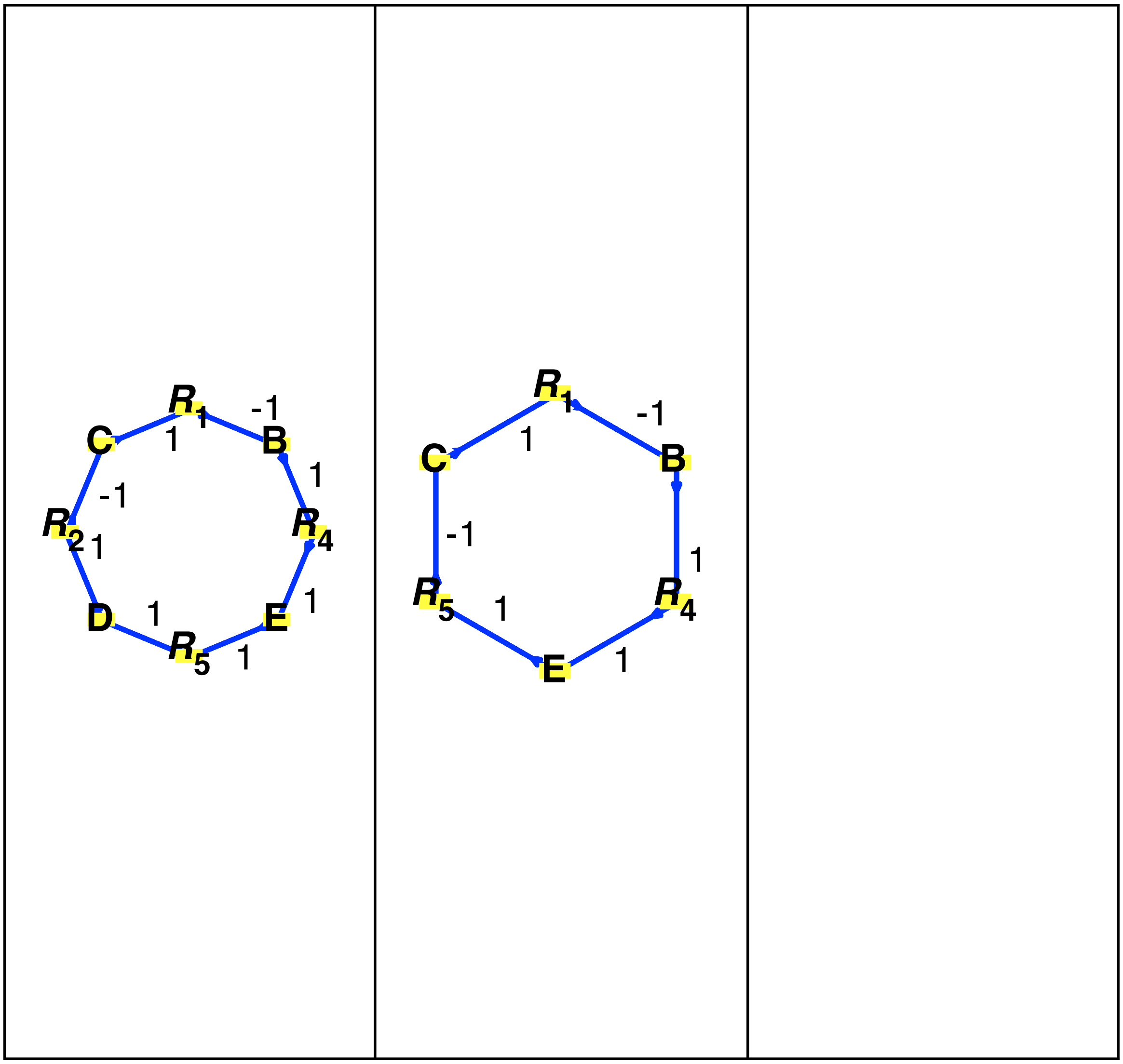
```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2955.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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









>

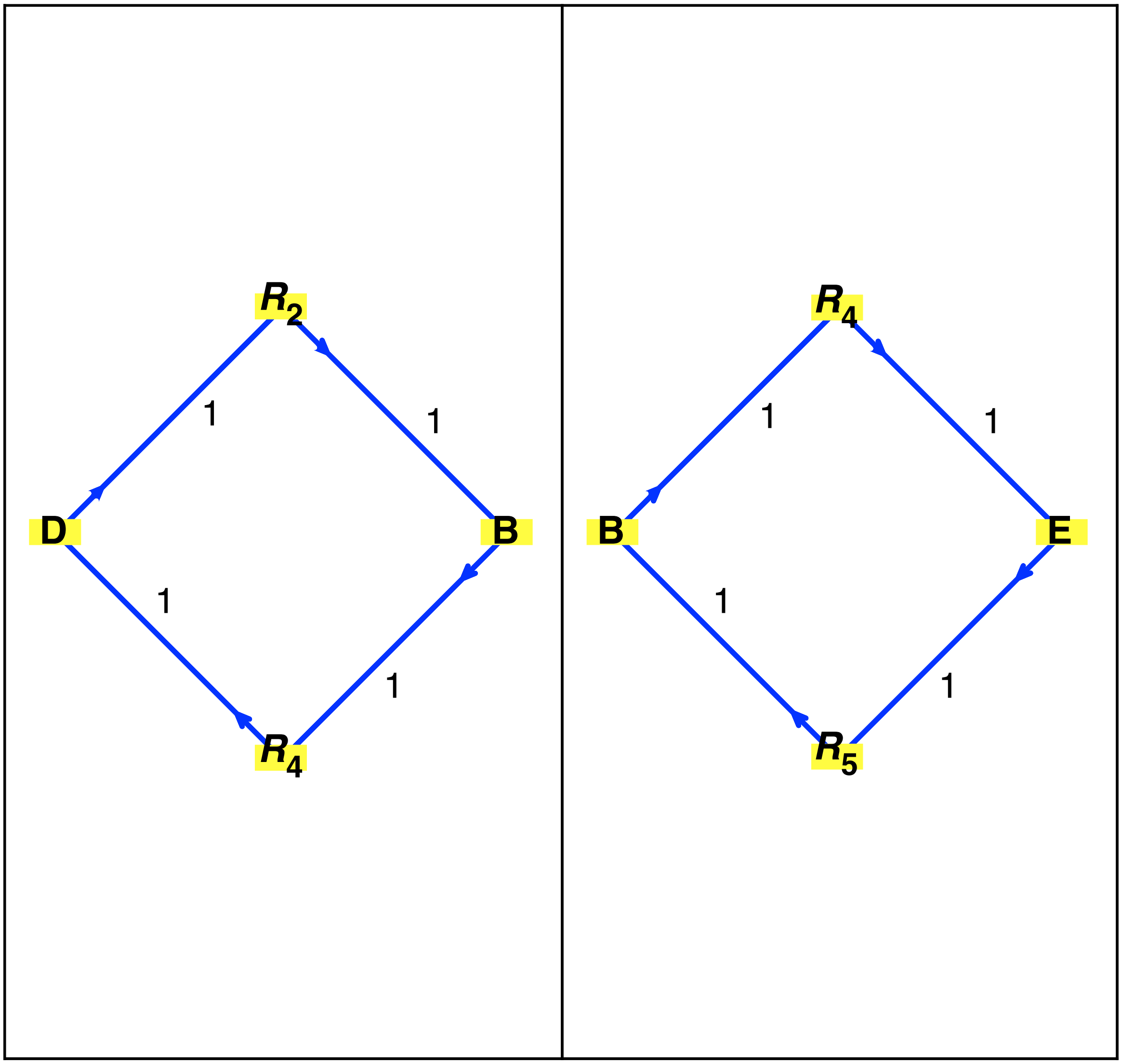
Meet condition 1 (?). NB: this is just complex formation, no futile cycles at all.

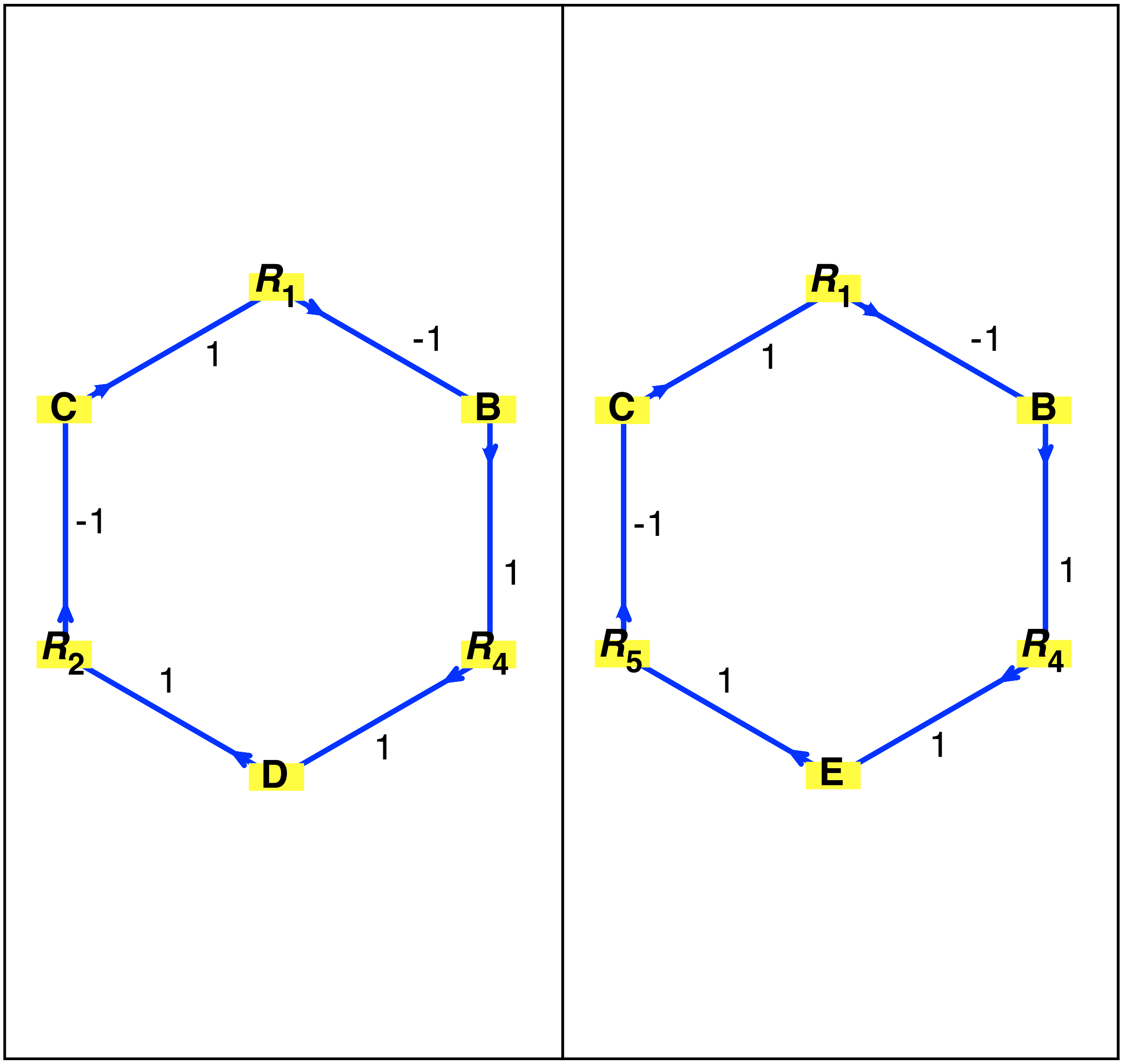
Network 2957 (meet neither, not realistic)

$B + C \to A$
$C + \mathrm{D} \to B$
$A \to B + C$
$B \to \mathrm{D} + E$
$C + E \to B$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2957.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

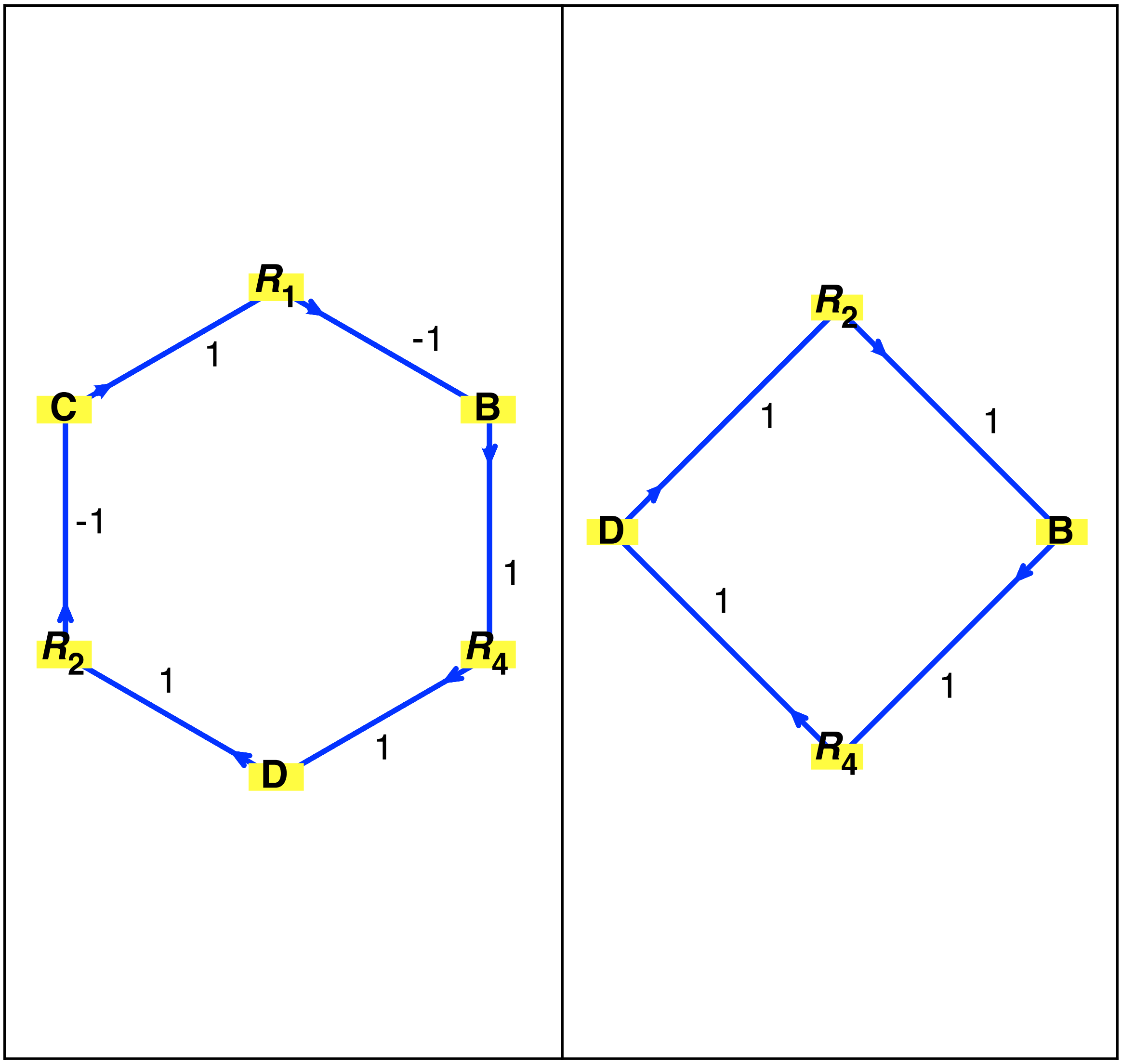
Meet both (?). NB: D is same as E!

Network 2958 (meet both condition 1 & 2, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow B + C$
$B \rightarrow \mathrm{D} + E$
$\mathrm{D} + E \rightarrow B$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2958.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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



>

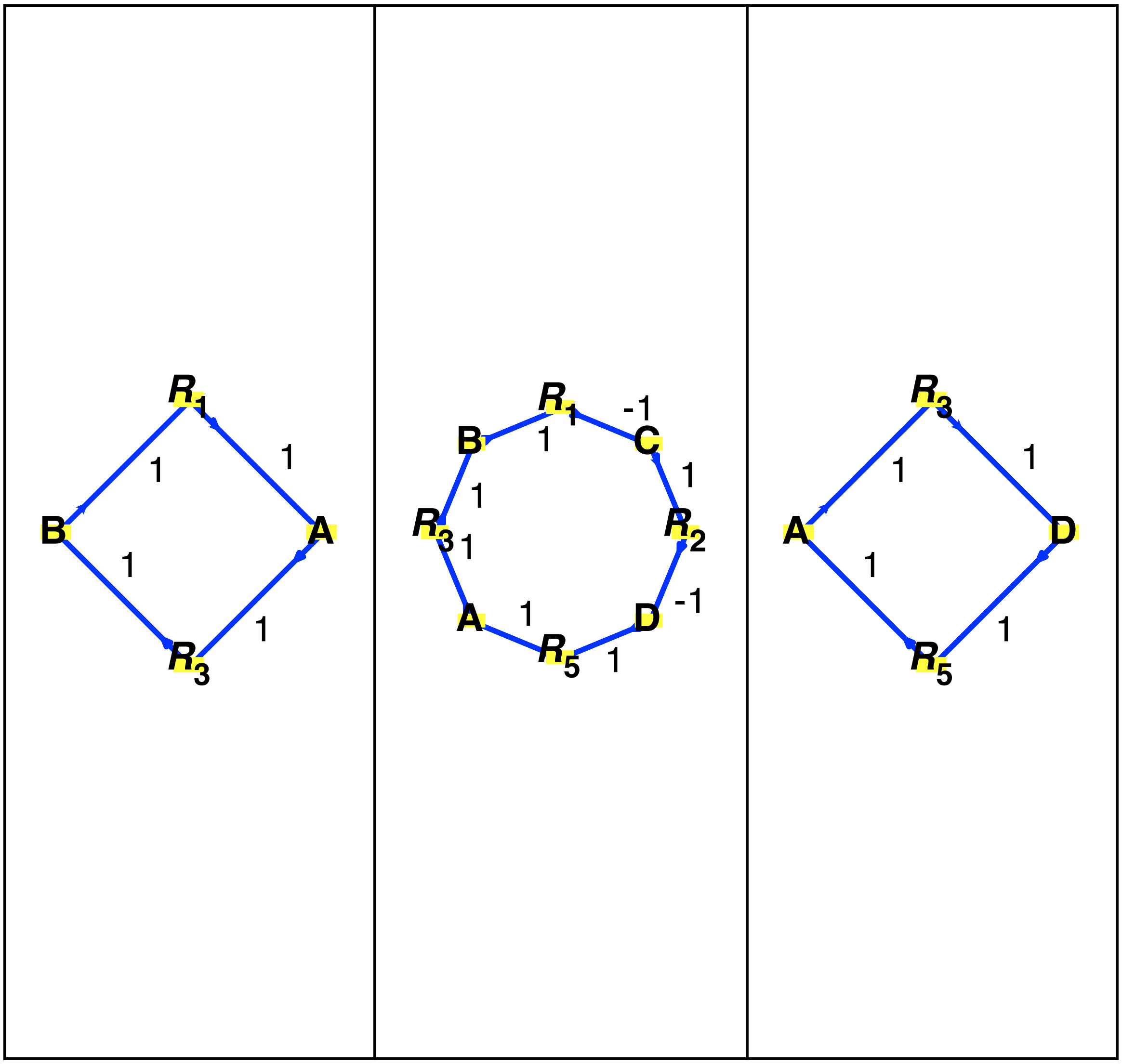
Meet both conditions. NB: E is same as C.

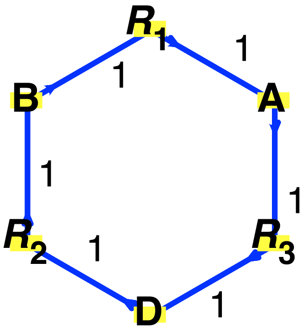
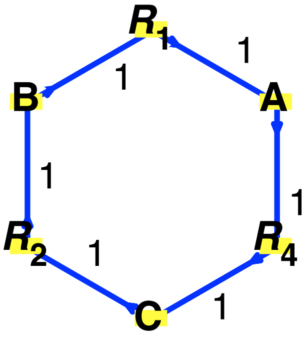
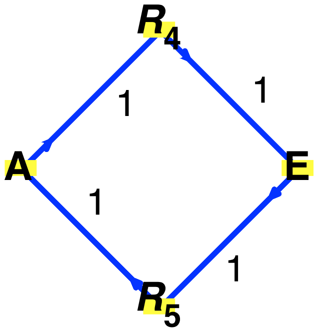
Network 2965 (meet neither, not realistic) three competing reactions

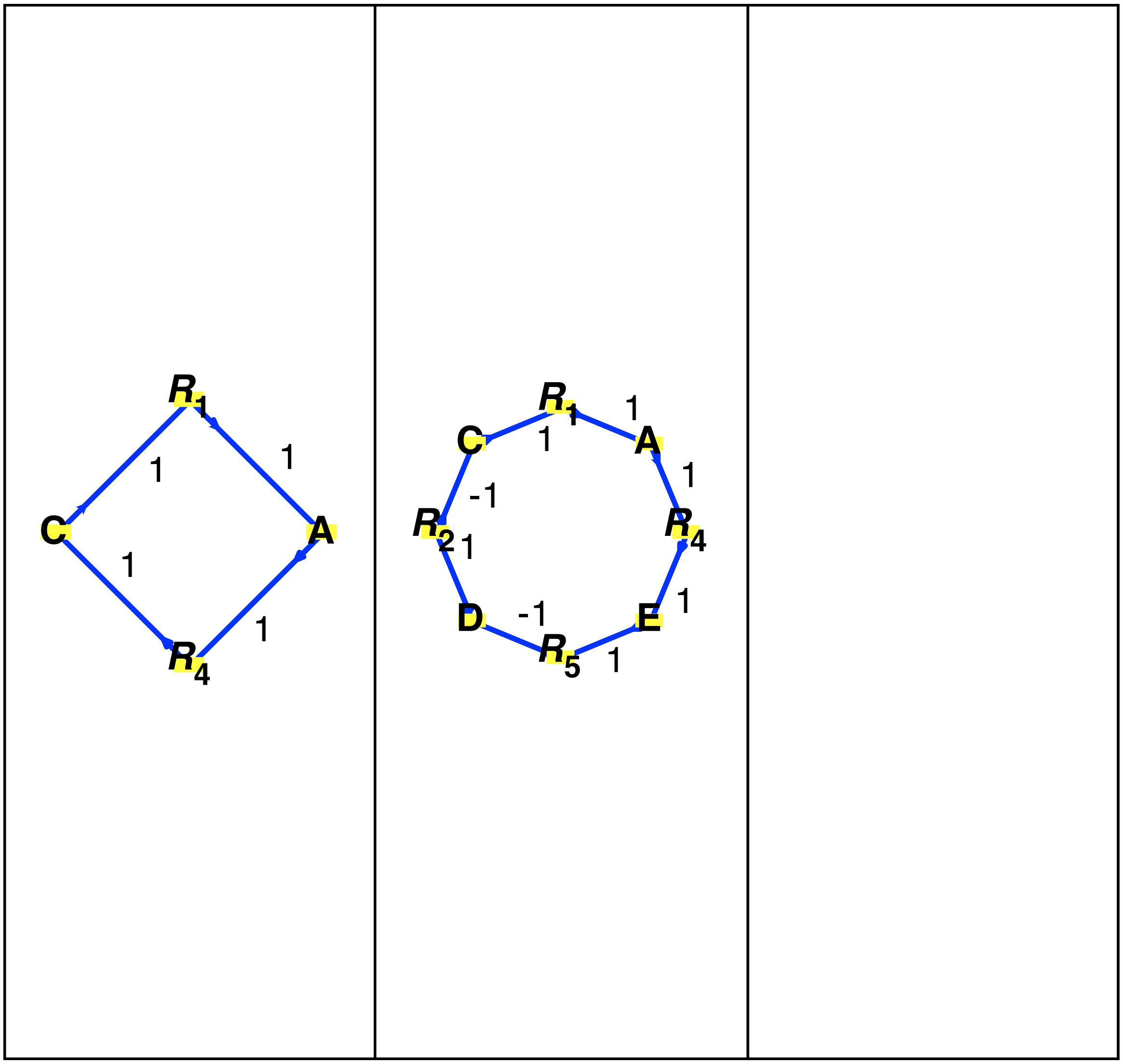
$B + C \rightarrow A$
$C + D \rightarrow B$
$A \rightarrow B + D$
$A \rightarrow C + E$
$D + E \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2965.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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



R_1 1 A 1 B 1 R_3 1 R_2 1 D 1	R_1 1 A 1 B 1 R_4 1 R_2 1 C 1	R_4 1 1 A E 1 R_5 1



```
>
```

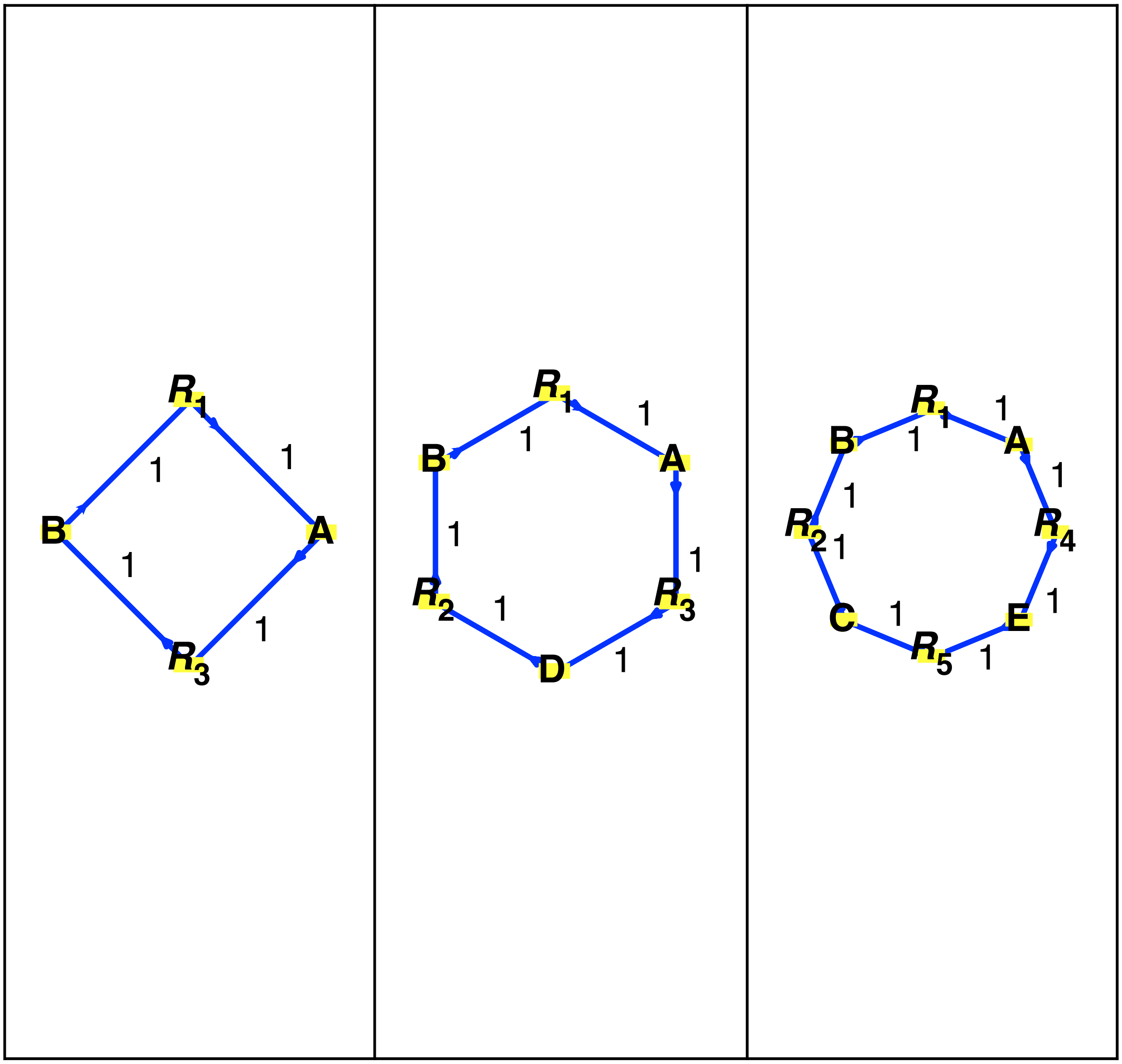
Meet both (?). NB: A is trimer, B & E are dimers of C & D, B catalyze C to D, C catalyze B to E.

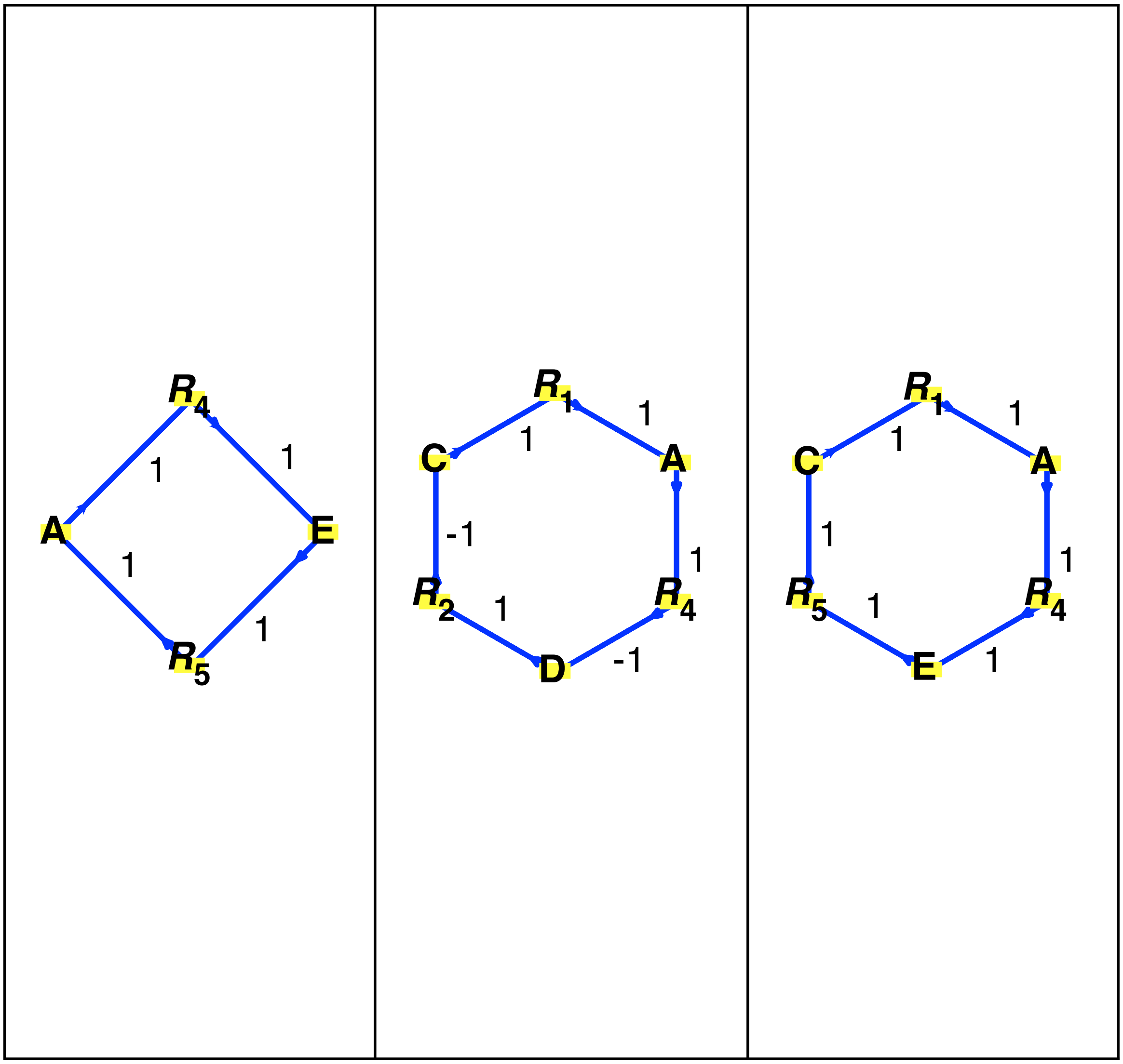
Network 2969 (meet neither, not realistic) three competing reactions

$B + C \to A$
$C + \mathrm{D} \to B$
$A \to B + \mathrm{D}$
$A + \mathrm{D} \to E$
$E \to A + C$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2969.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

Meet both (?). NB: D is tetramer, A is trimer, B is dimer of C & D, B catalyze C to D, A catalyze D to C.

Network 2970 (meet neither, maybe not realistic) three competing reactions

$B + C \rightarrow A$

$C + \mathrm{D} \rightarrow B$

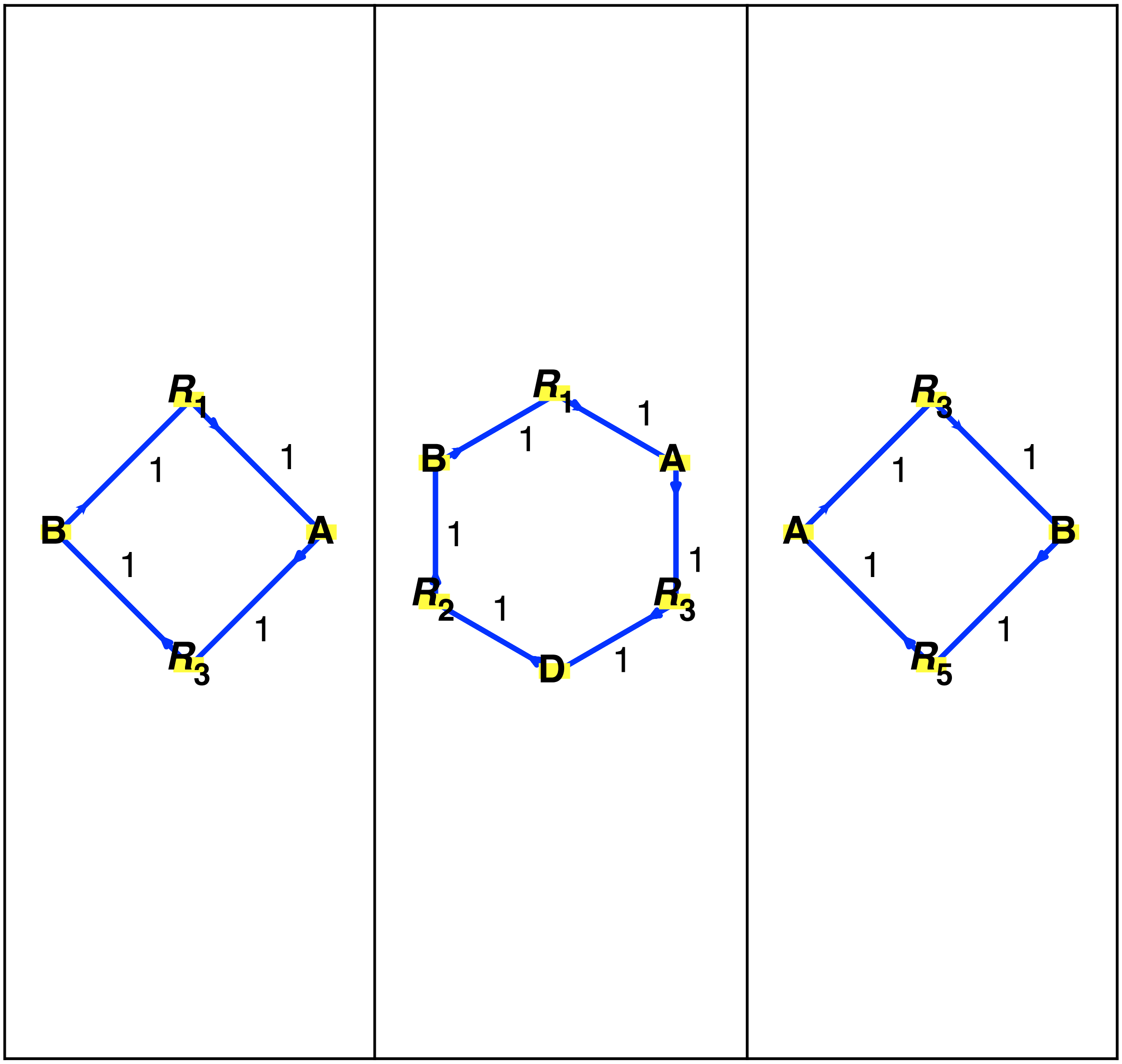
$A \rightarrow B + \mathrm{D}$

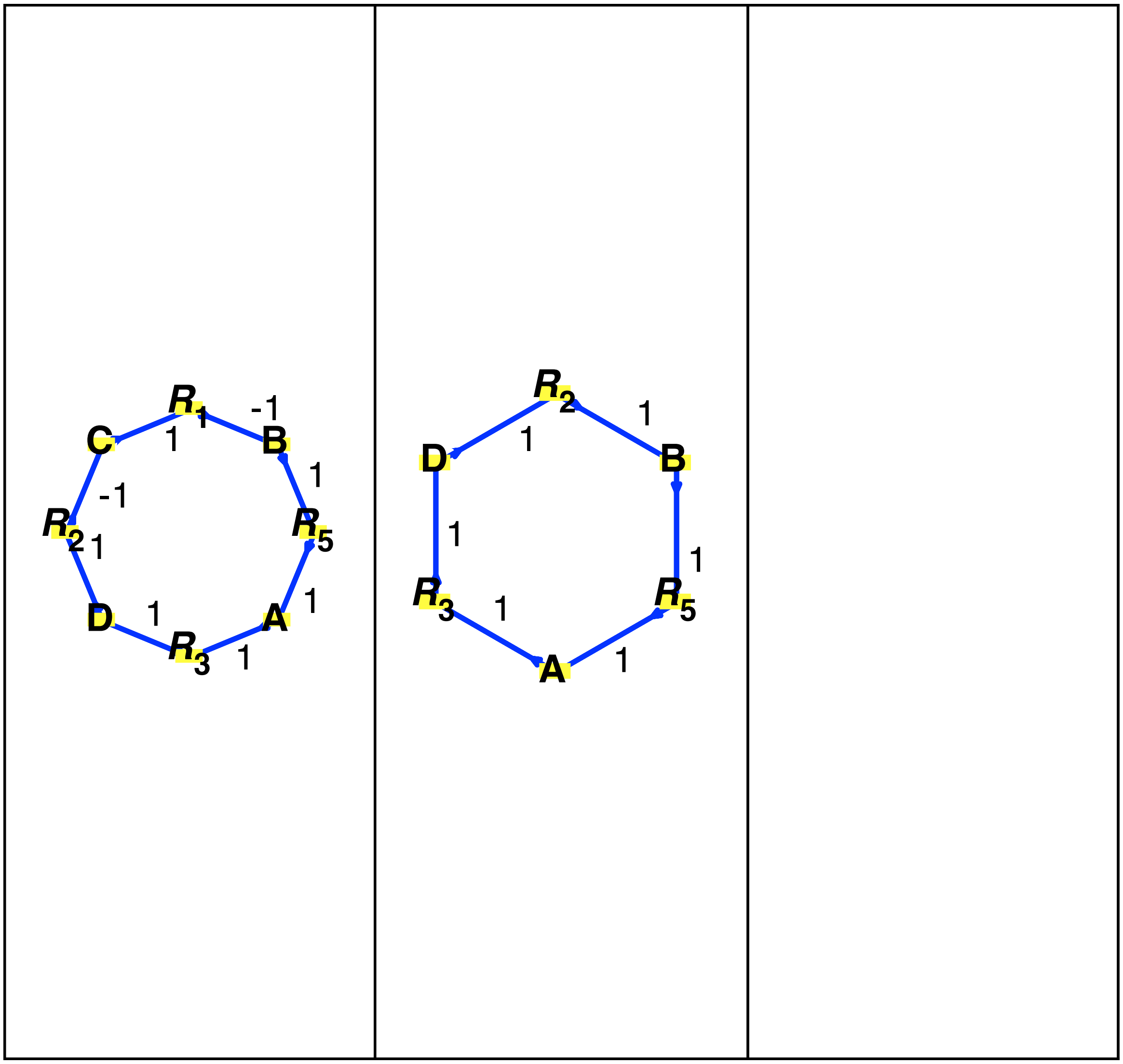
$B \rightarrow C + E$

$E + B \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2970.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

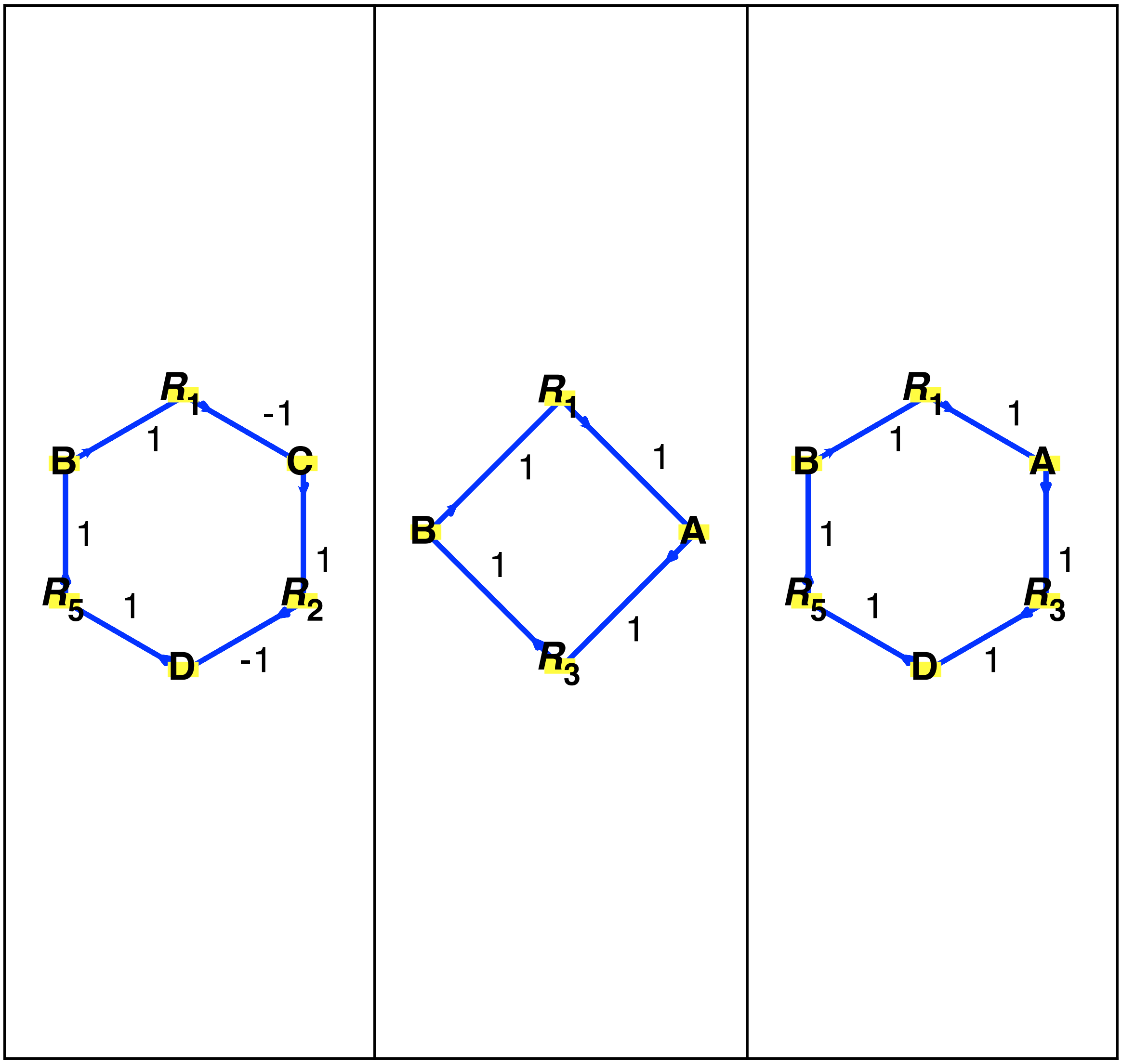
Meet both (?). NB: A is trimer, B is dimer of C, D & E, B catalyze C to D, C catalyze D to E.

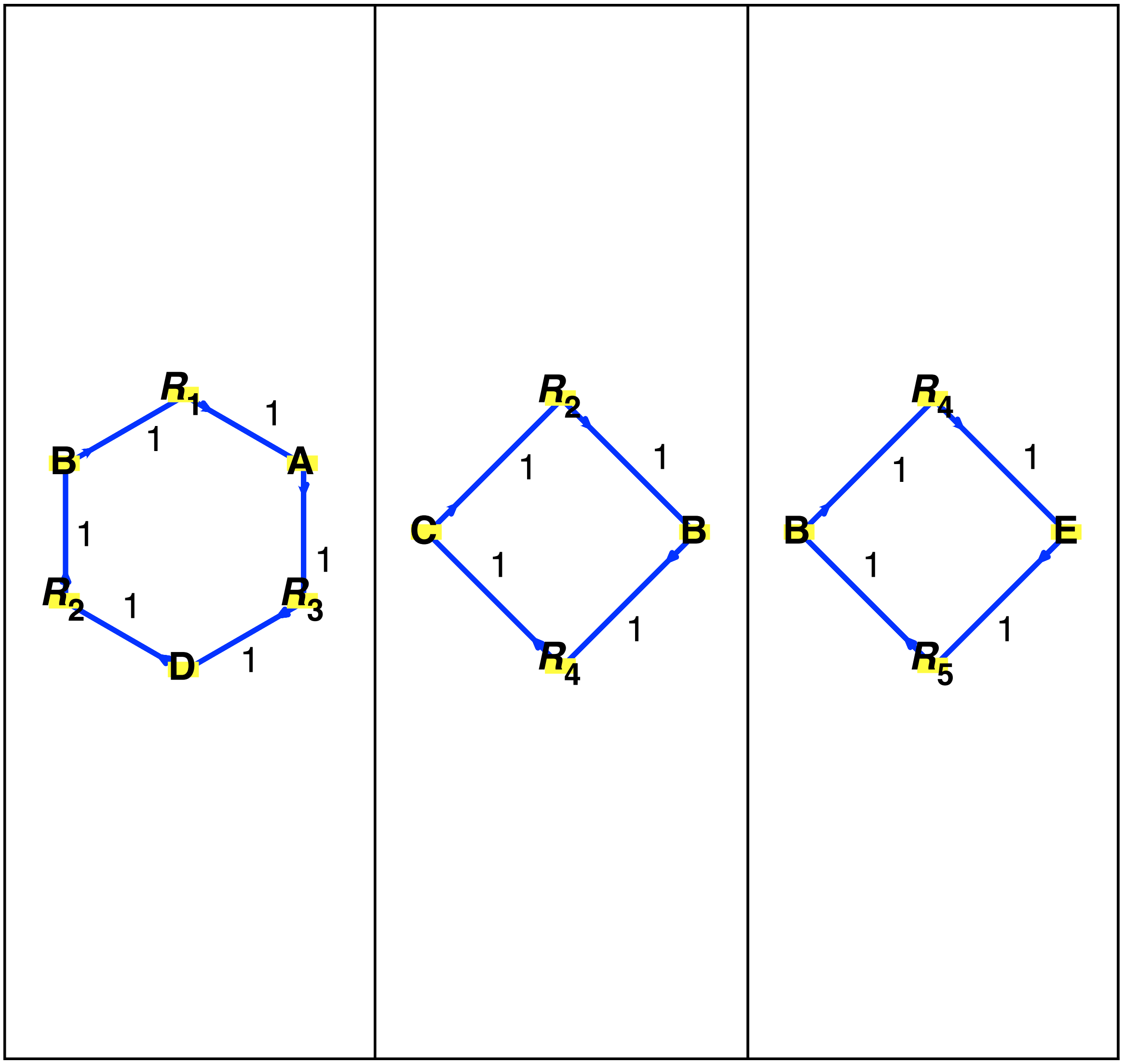
Network 2972 (meet neither, maybe not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow B + \mathrm{D}$
$B \rightarrow C + E$
$E + \mathrm{D} \rightarrow B$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2972.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

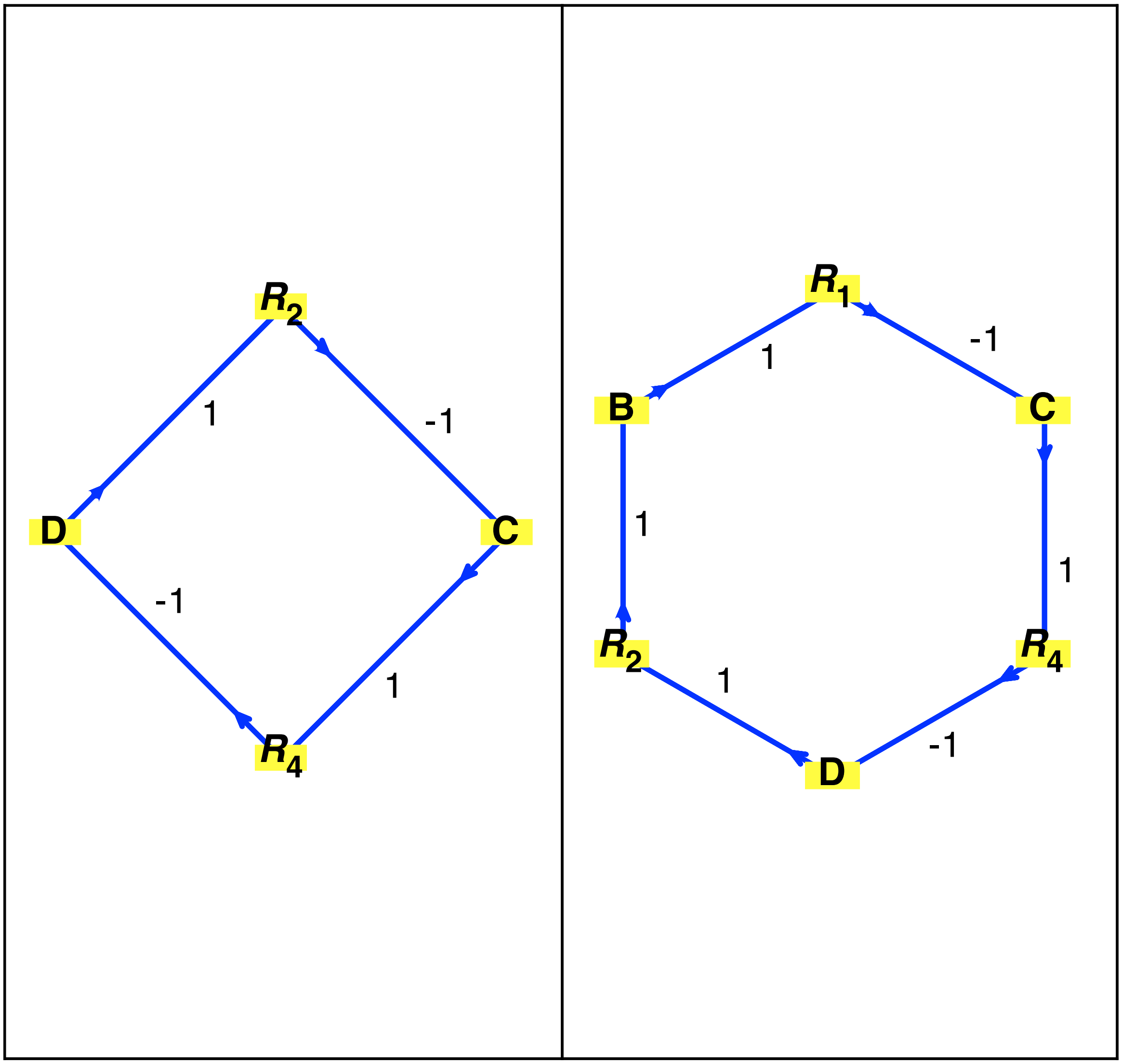
Meet both (?). NB: A is trimer, B is dimer of C, D & E, B catalyze C to D, C catalyze D to E.

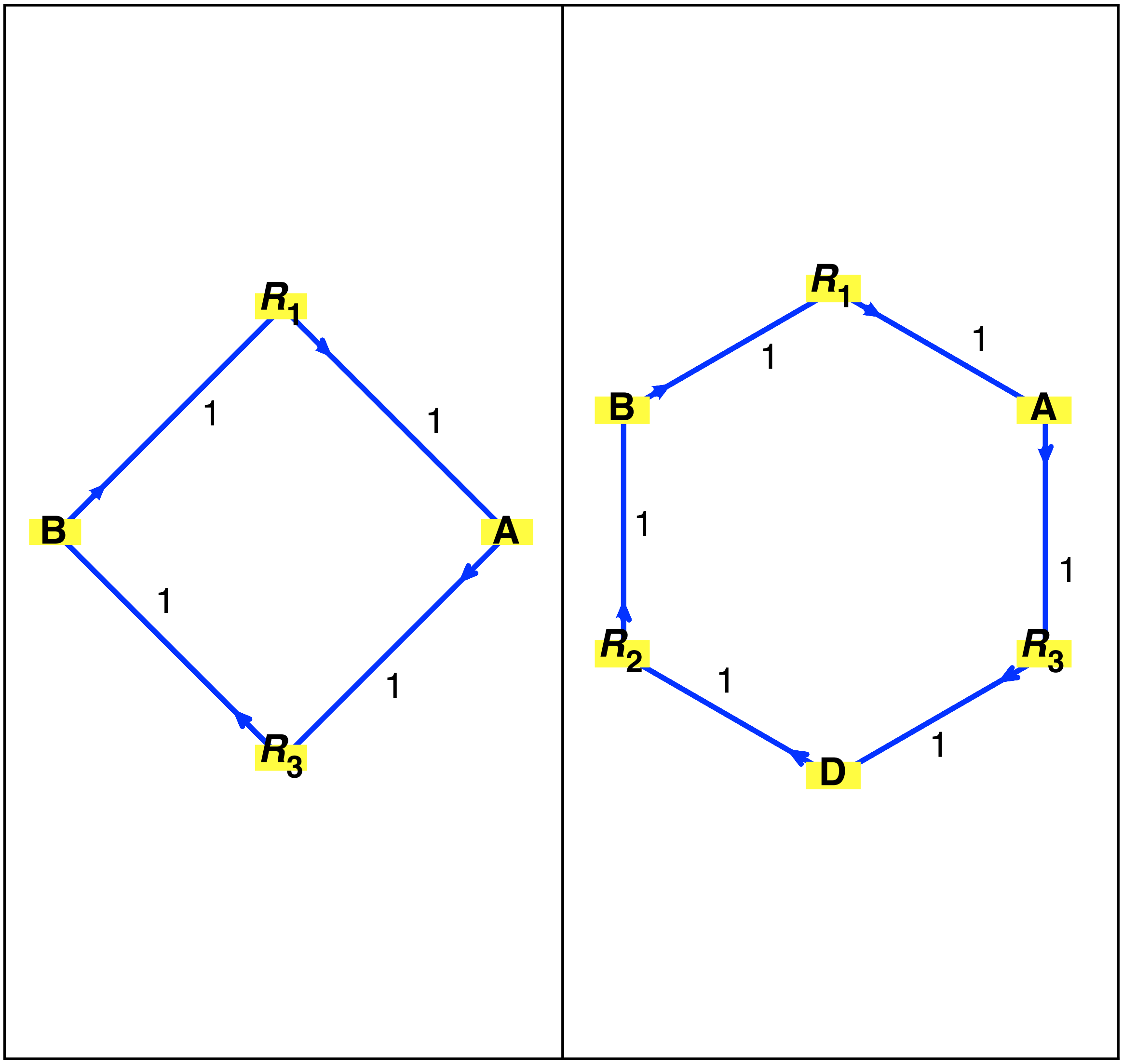
Network 2975 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow B + \mathrm{D}$
$C + \mathrm{D} \rightarrow E$
$E \rightarrow C + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2975.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

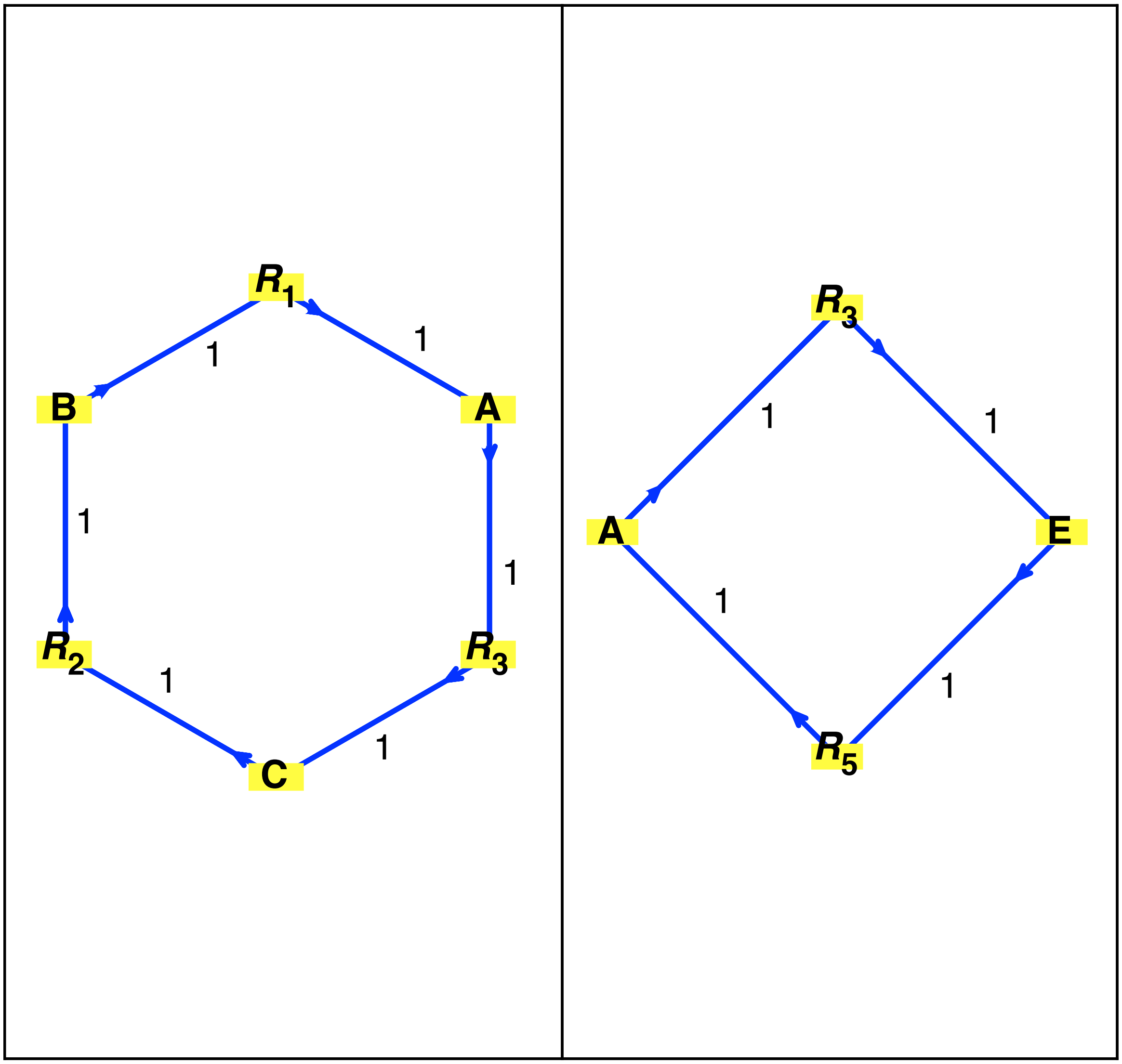
Meet both (?). NB: E is same as B.

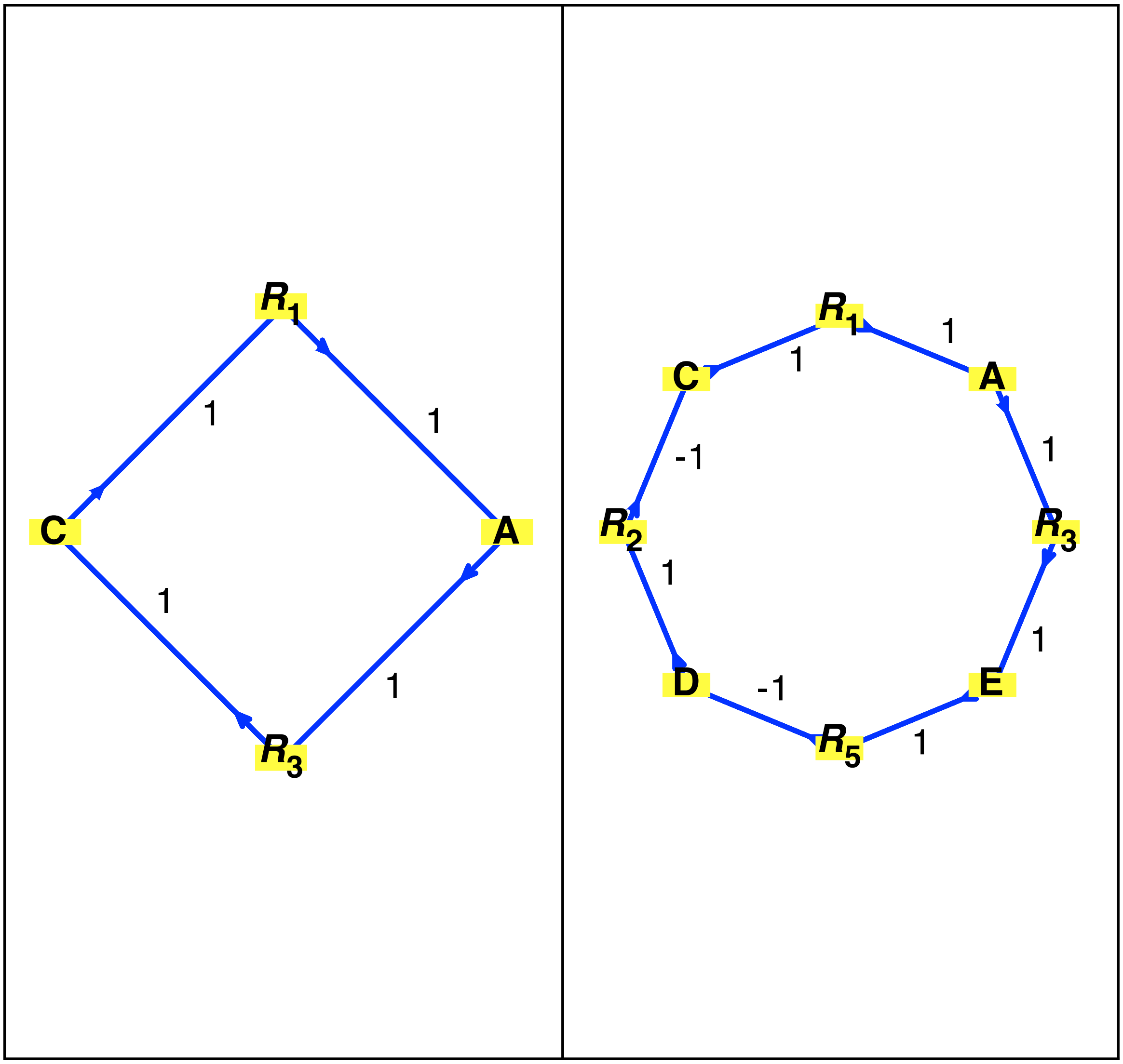
Network 2983 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow C + E$
$A \rightarrow \mathrm{D} + E$
$E + \mathrm{D} \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2983.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

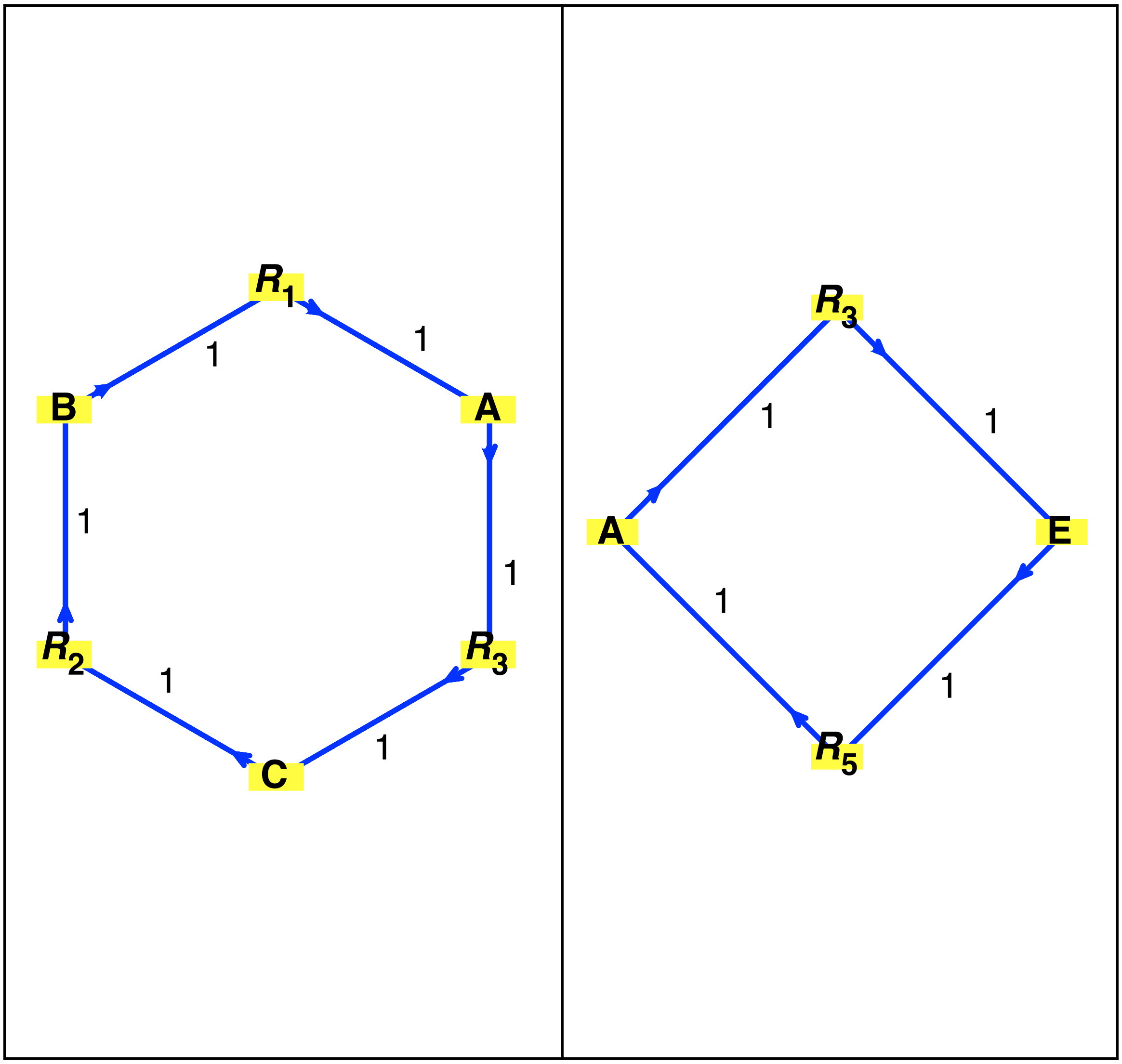
Meet both (?). NB: C is same as D.

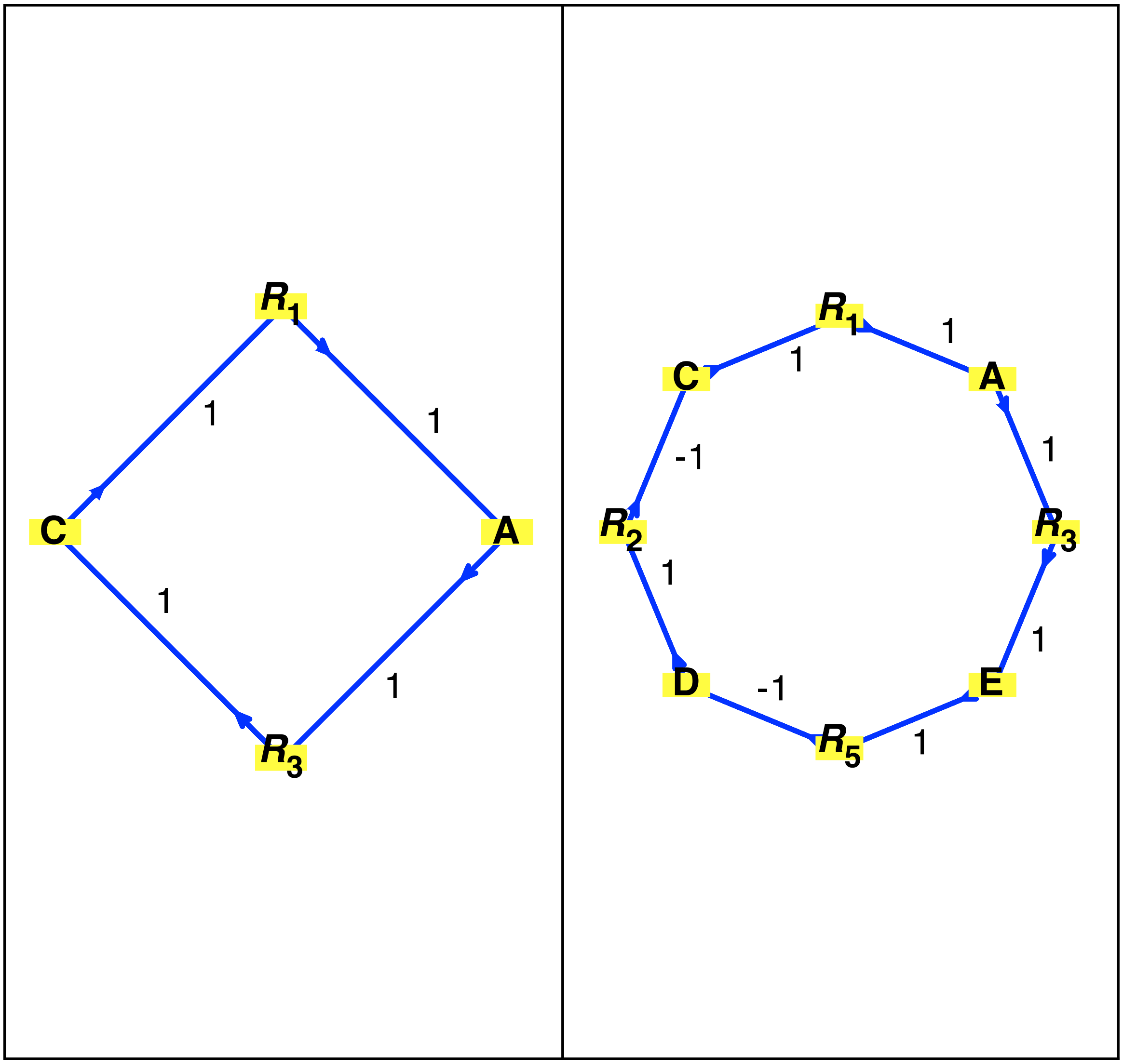
Network 2986 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow C + E$
$B \rightarrow C + \mathrm{D}$
$E + \mathrm{D} \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2986.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

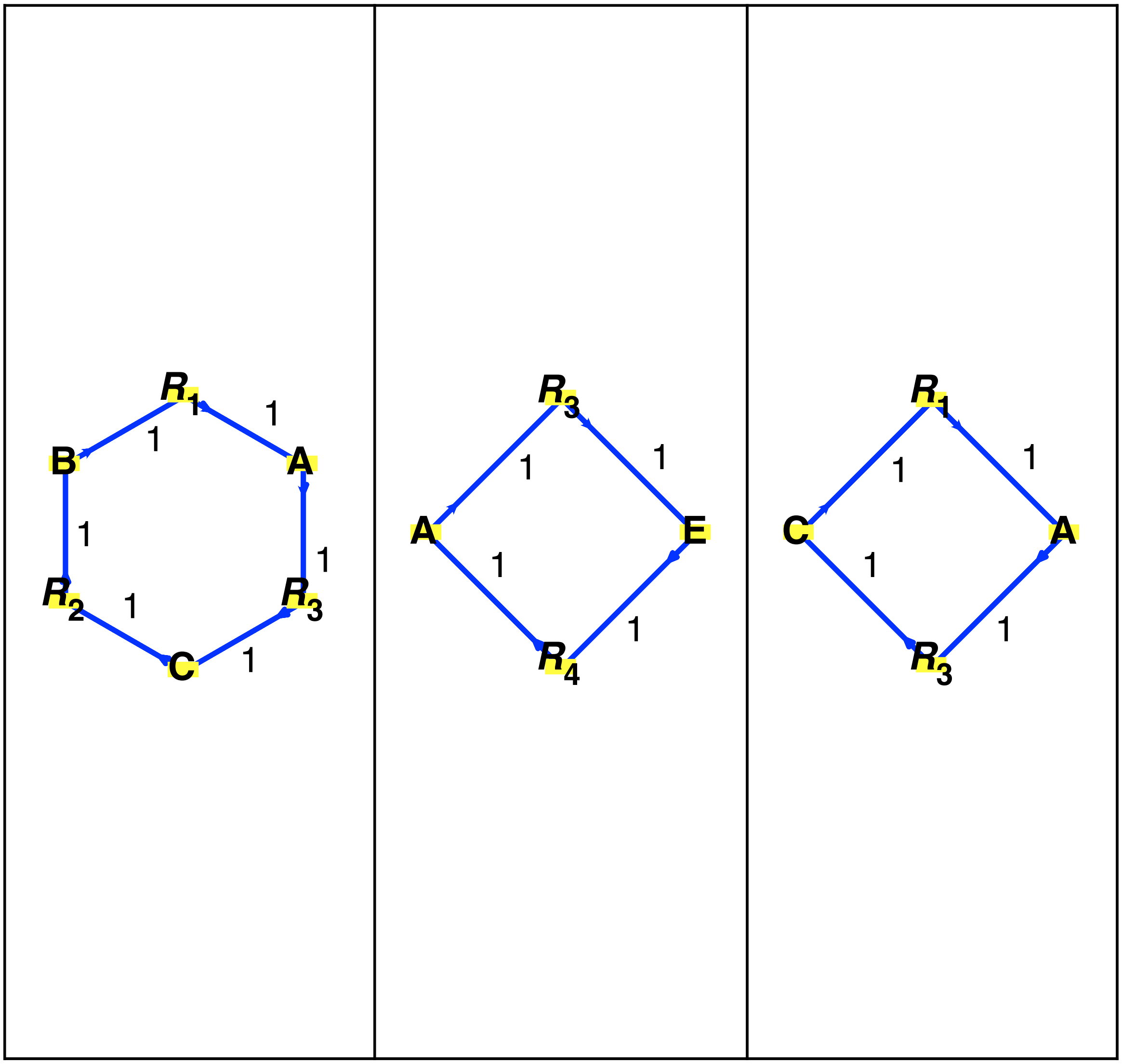
Meet both (?). NB: A is trimer, B & E are dimers of C & D, C catalyze B to E.

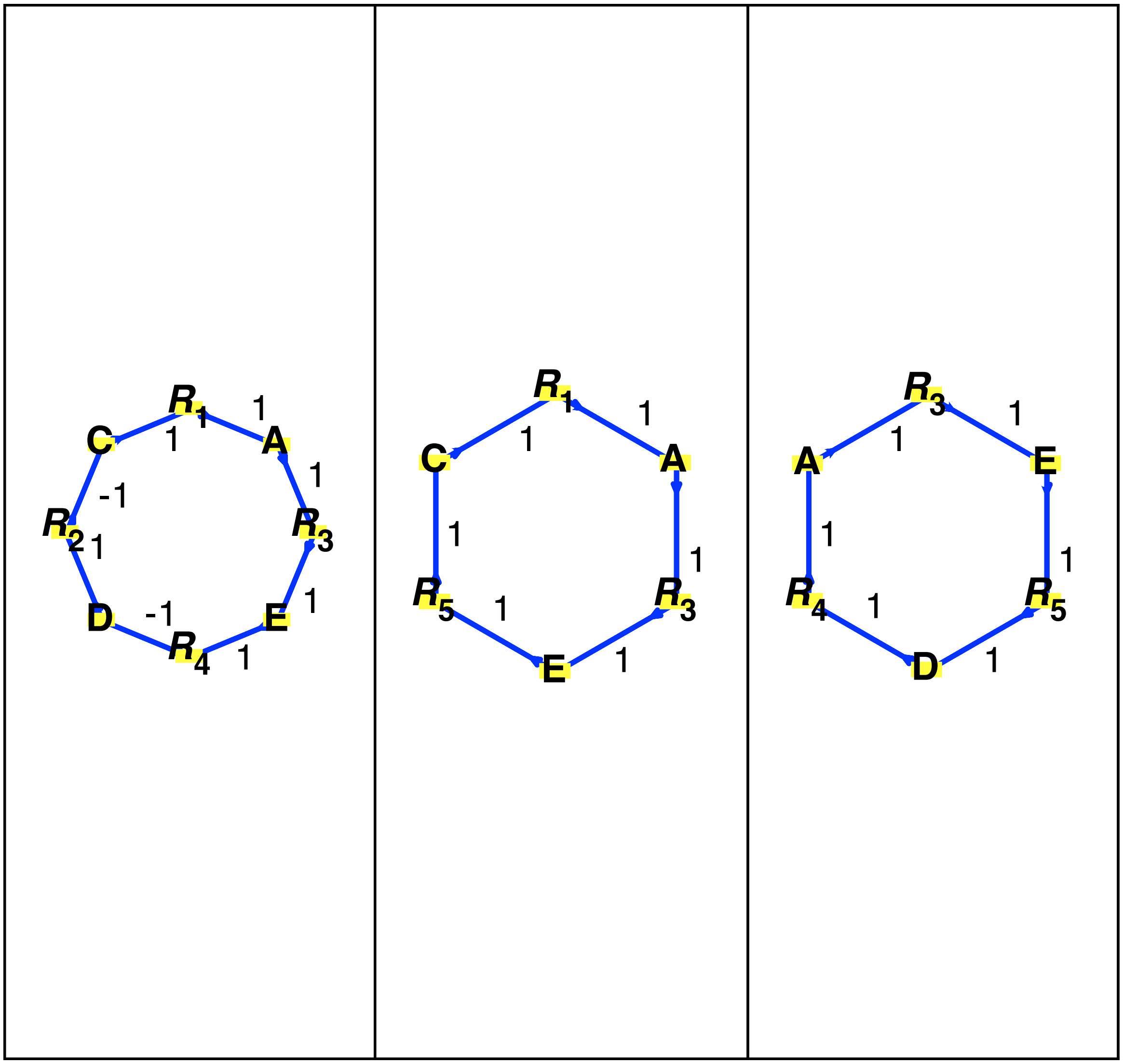
Network 2992 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A \rightarrow C + E$
$E \rightarrow C + \mathrm{D}$
$E + \mathrm{D} \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_2992.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

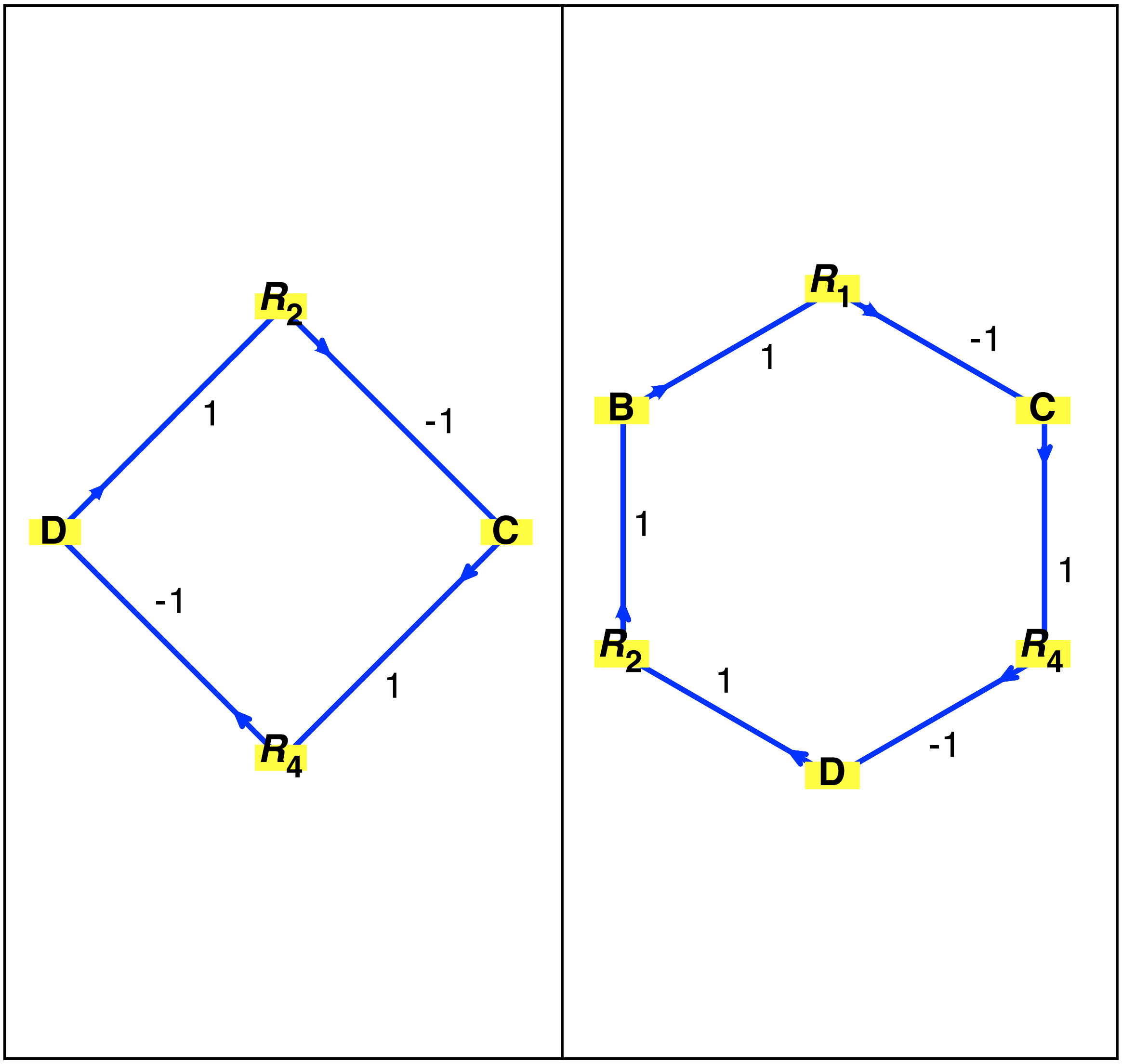
Meet both (?). NB: A is trimer, B & E are dimers of C & D, C catalyze B to E.

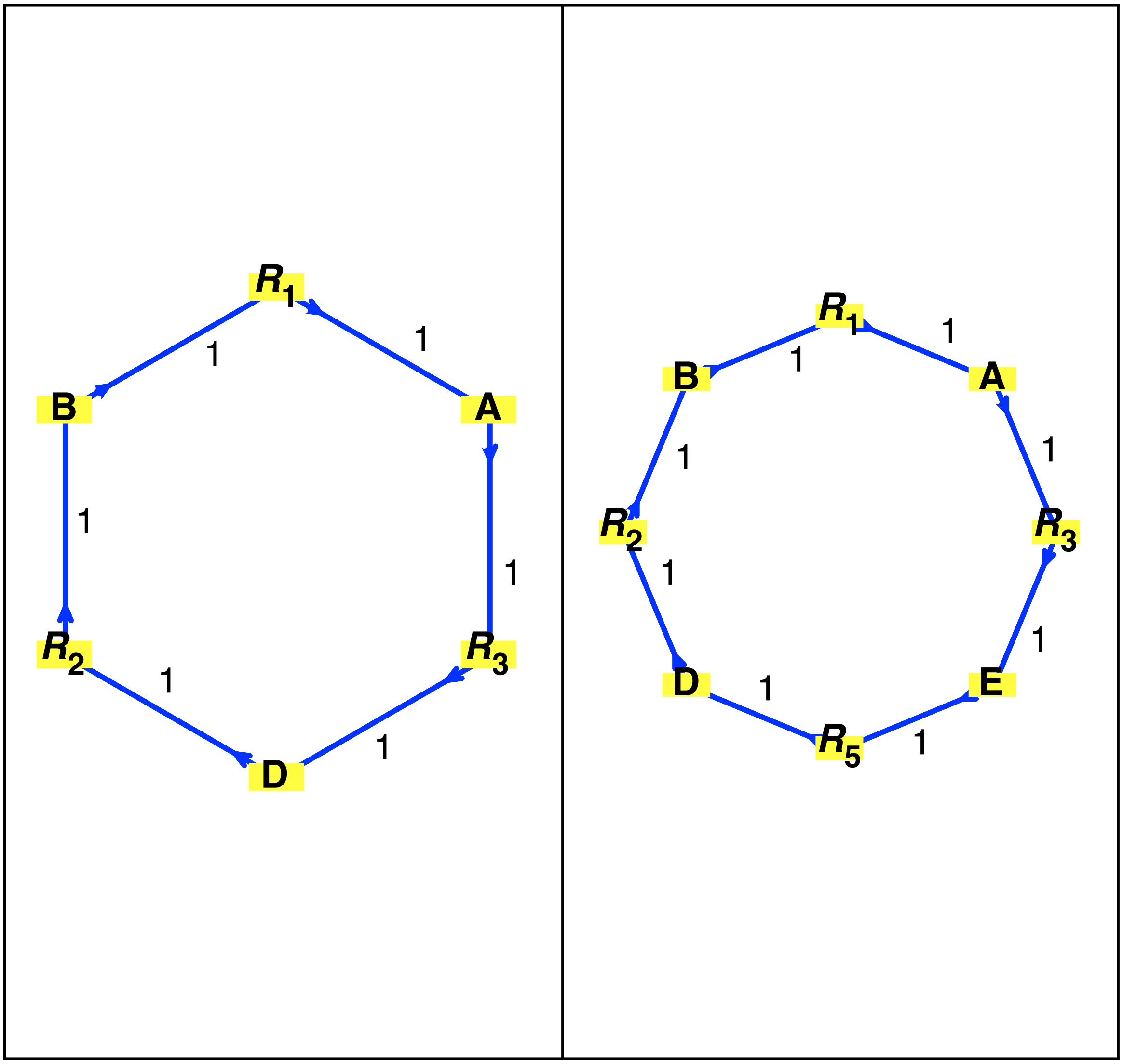
Network 3007 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + D \rightarrow B$
$A \rightarrow D + E$
$C + D \rightarrow E$
$E \rightarrow C + D$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3007.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

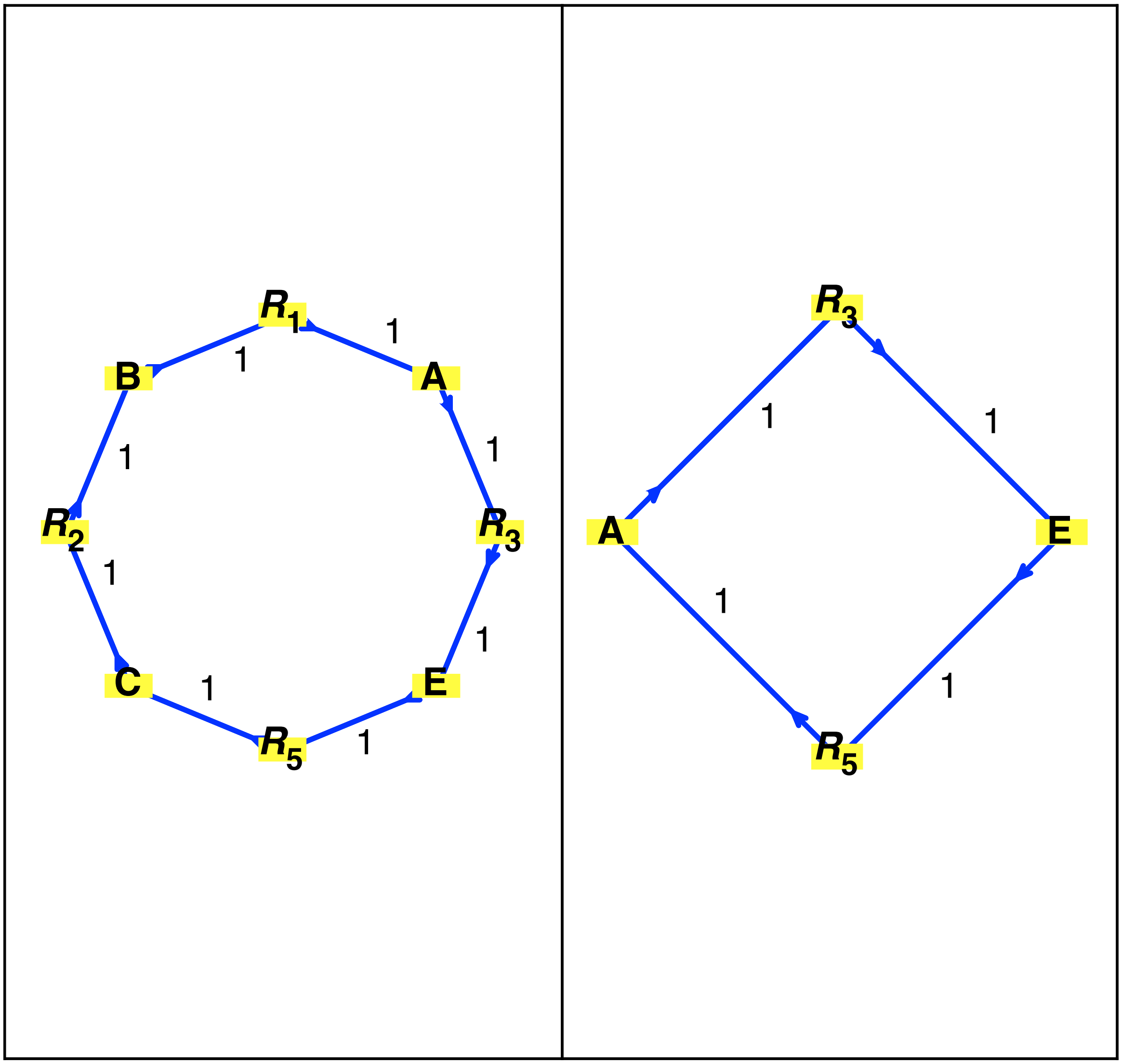
Meet both (?). NB: B is same as E, A is trimer, B & E are dimers of C & D.

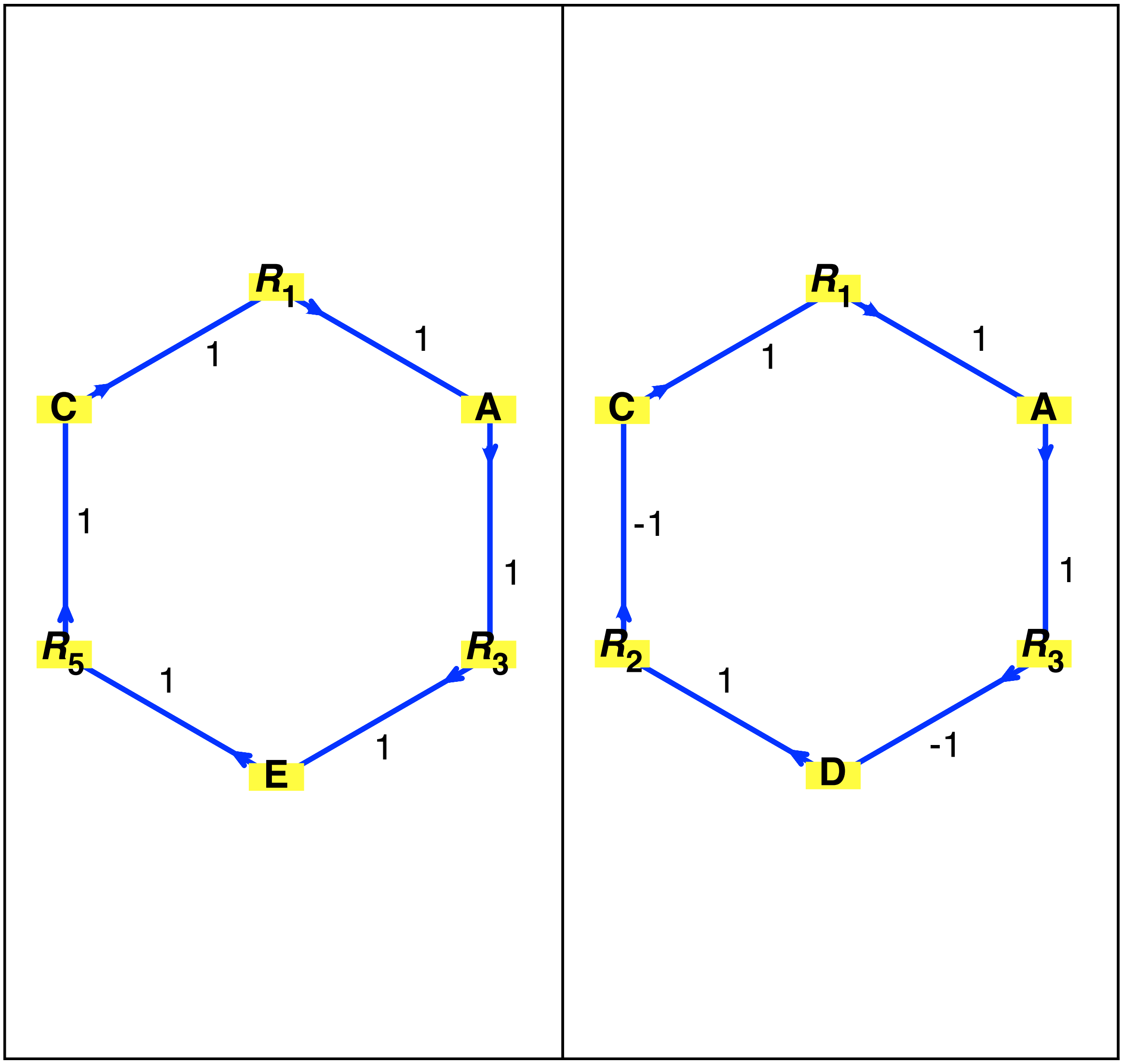
Network 3015 (meet neither, maybe not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow B$
$A + \mathrm{D} \rightarrow E$
$B \rightarrow C + \mathrm{D}$
$E \rightarrow A + C$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3015.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

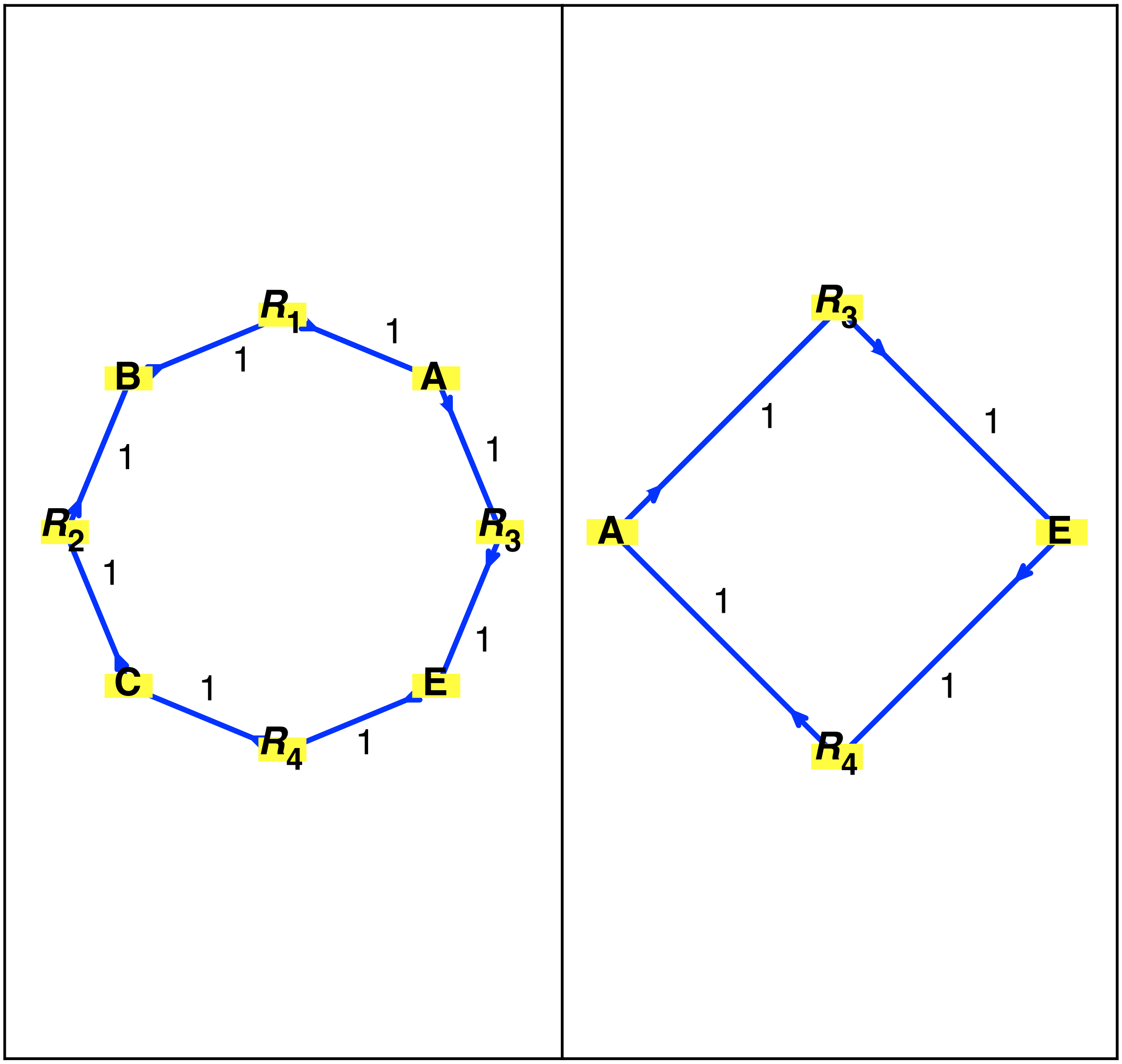
Meet both (?). NB: E is tetramer, A is trimer, B is dimer of C & D.

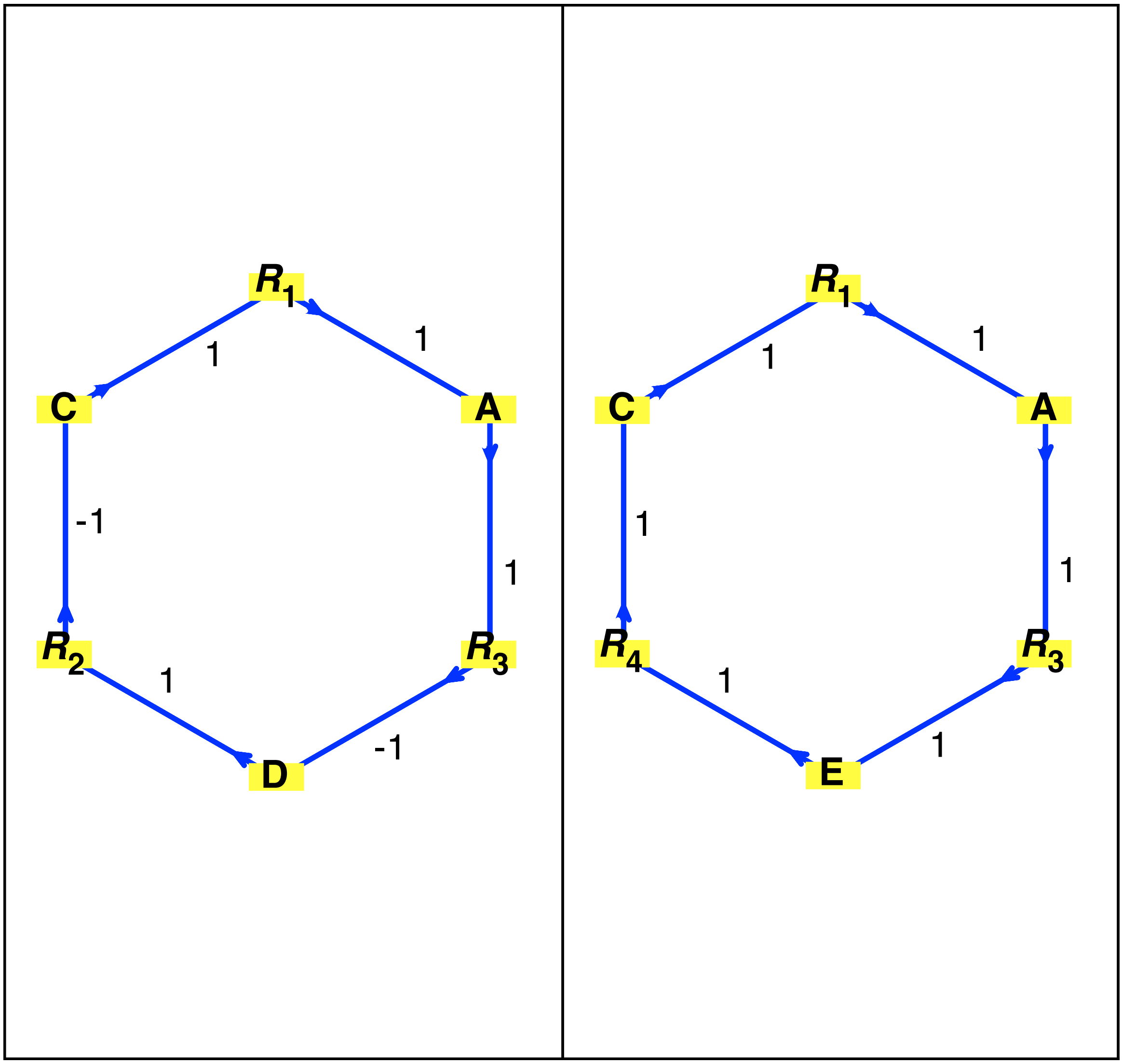
Network 3017 (meet neither, maybe not realistic) three competing reactions

$B + C \to A$
$C + \mathrm{D} \to B$
$A + \mathrm{D} \to E$
$E \to A + C$
$E \to A + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3017.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

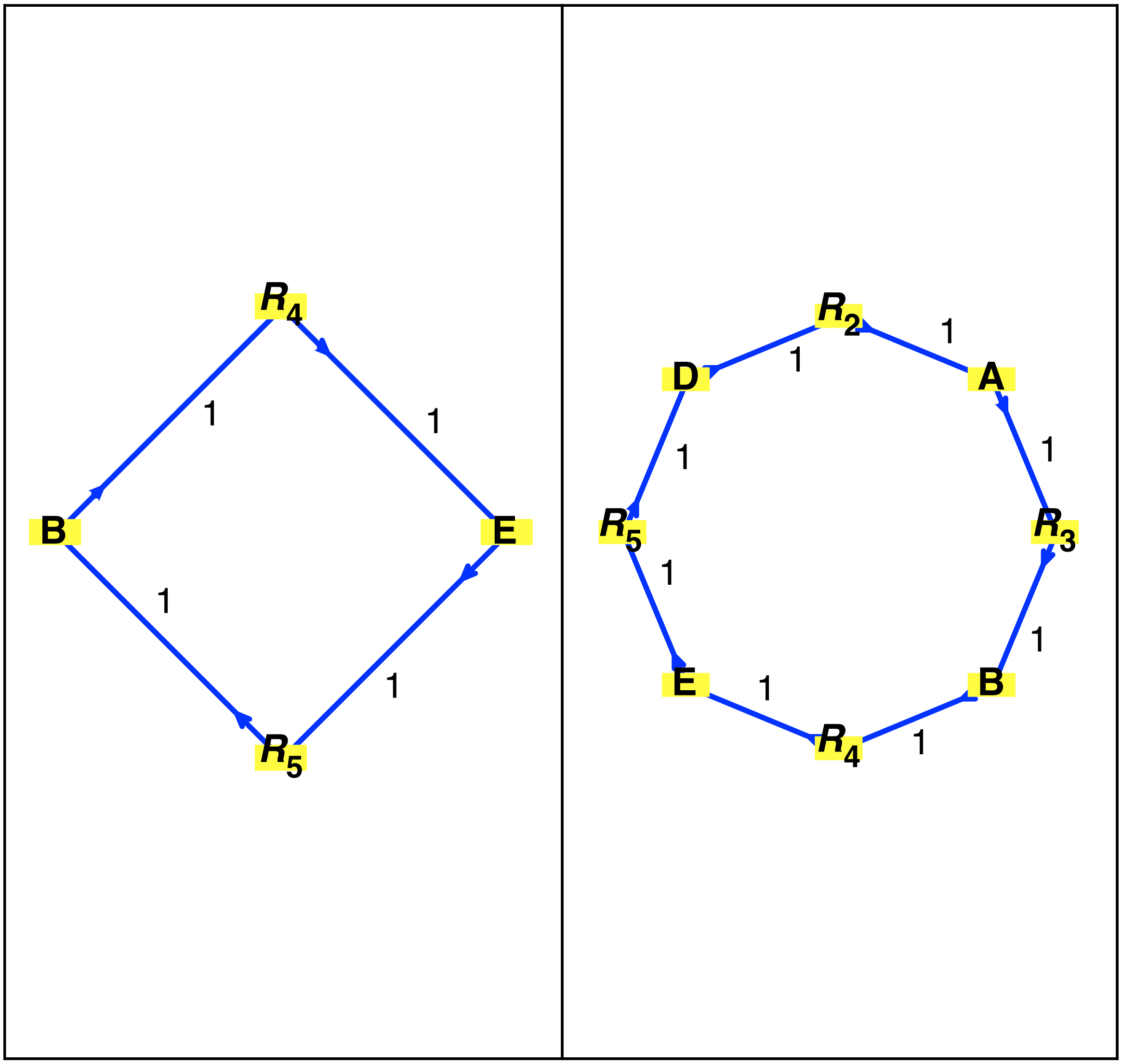
Meet both (?). NB: E is tetramer, A is trimer, B is dimer of C & D. A catalyze D to C

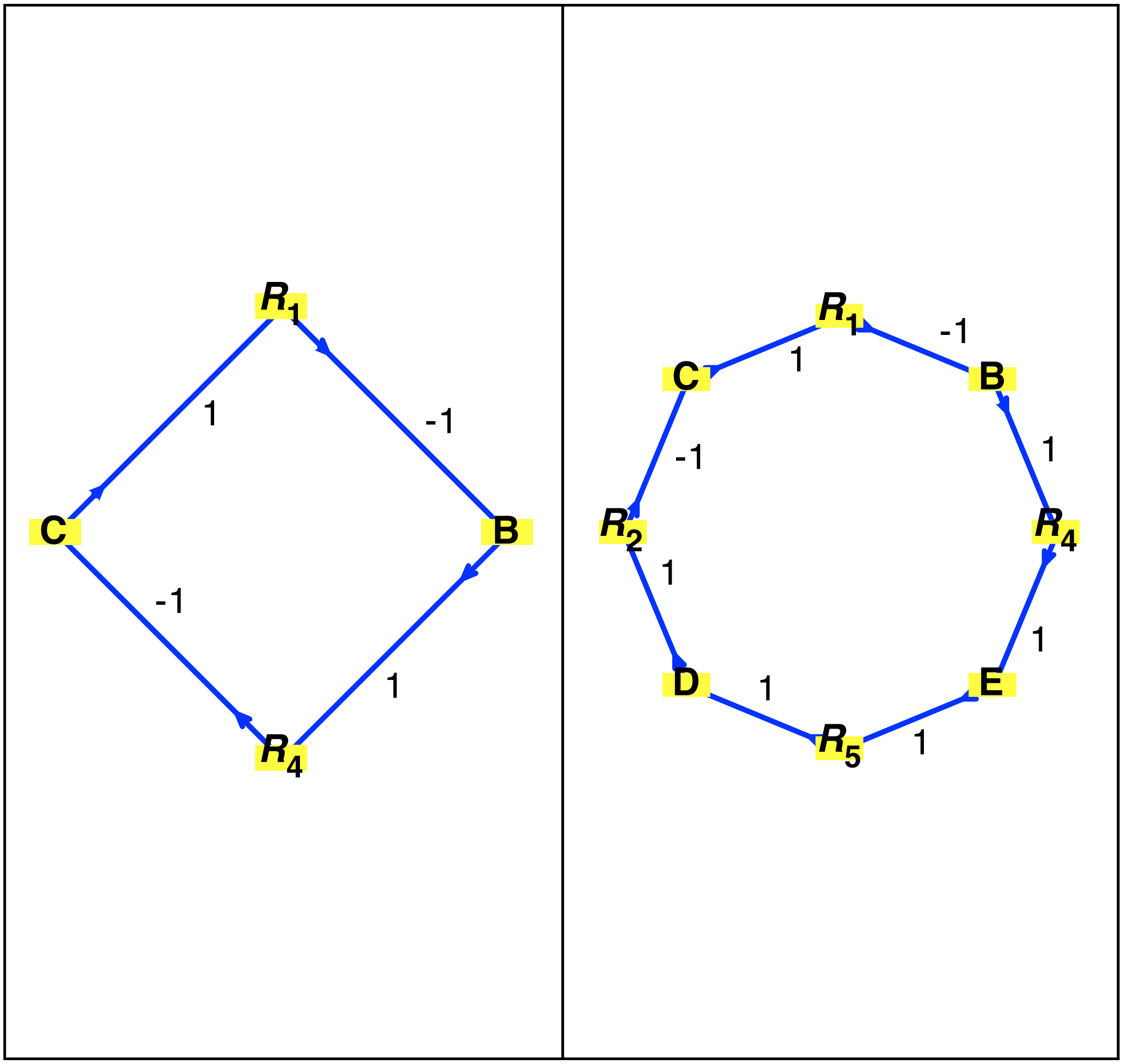
Network 3110 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow A$
$A \rightarrow B + C$
$B + C \rightarrow E$
$E \rightarrow B + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3110.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

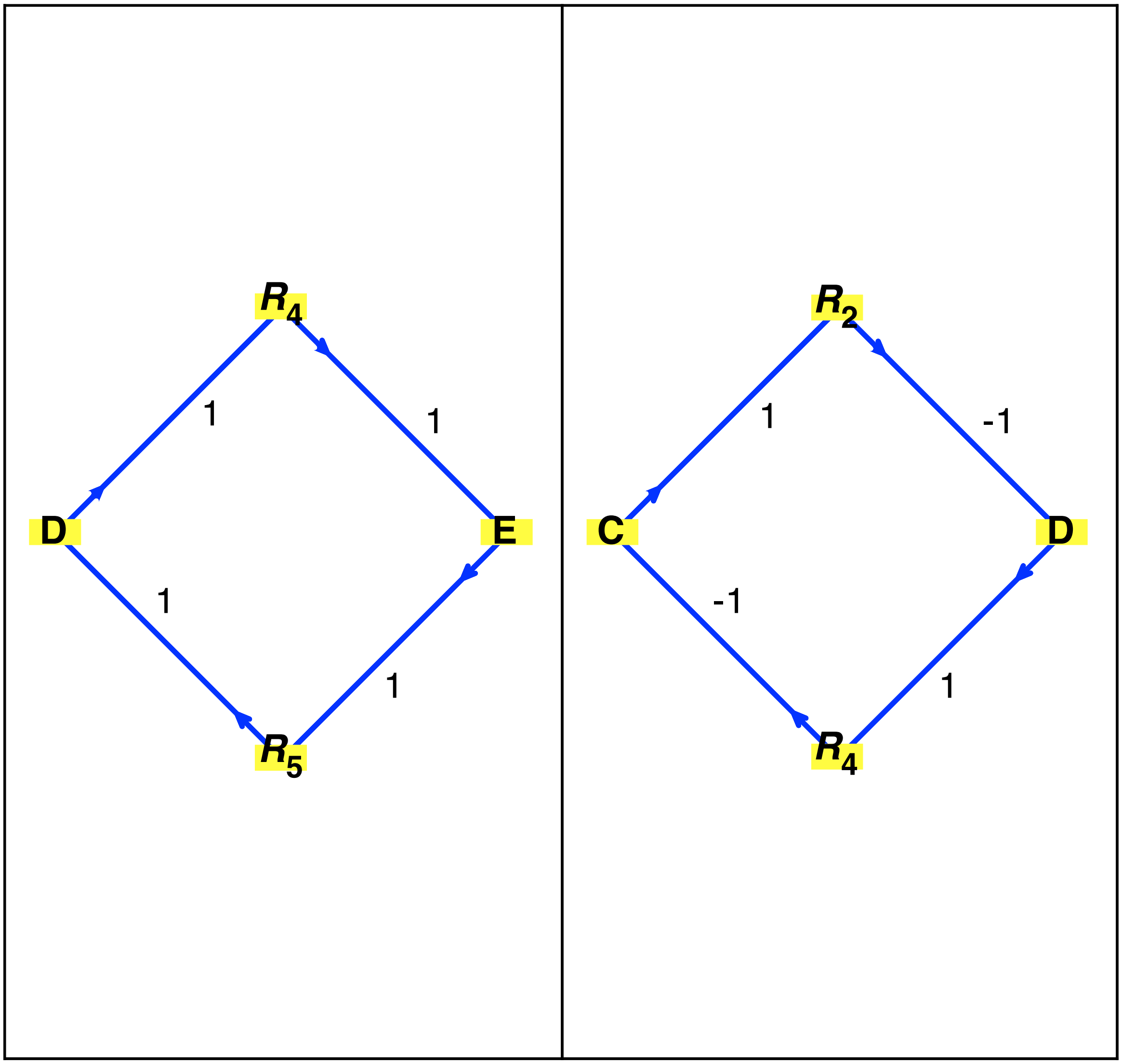
Meet both (?). NB: D is same as B. C, B, D are three states. A & E are dimer of B, C & D. C catalyze D to B, B catalyze C to D.

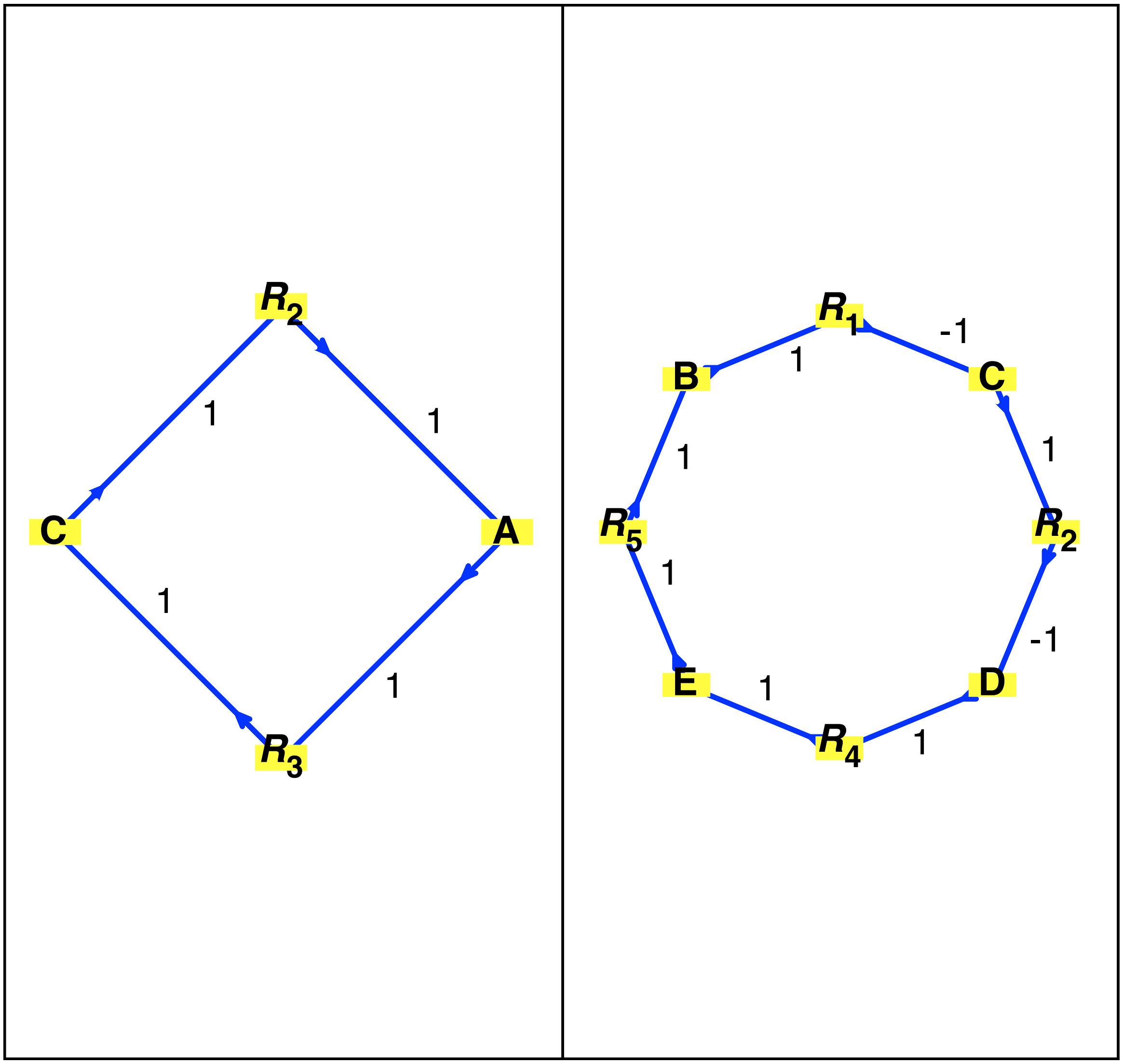
Network 3118 (meet neither, not realistic) three competing reactions

$B + C \to A$
$C + \mathrm{D} \to A$
$A \to B + C$
$C + \mathrm{D} \to E$
$E \to B + \mathrm{D}$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3118.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





```
>
```

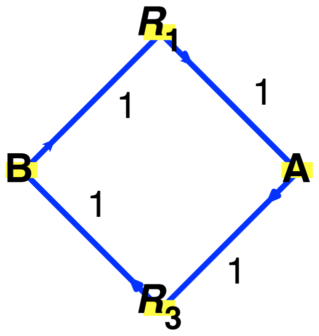
Meet both (?). NB: A is same as E. B is same as D. C, B, D are three states. A & E are dimer of B, C & D. C catalyze D to B, D catalyze C to B.

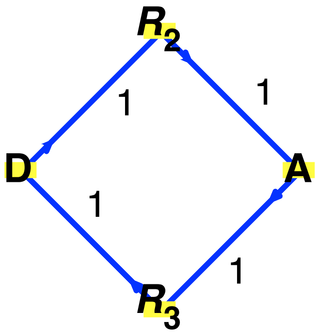
Network 3123 (meet neither, not realistic) three competing reactions

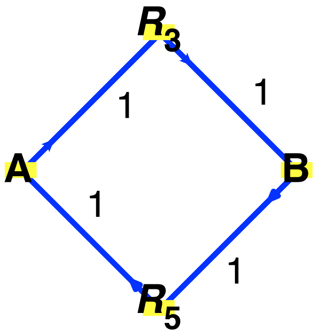
$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow A$
$A \rightarrow B + \mathrm{D}$
$A \rightarrow C + E$
$E + B \rightarrow A$

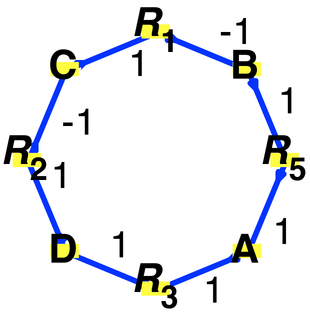
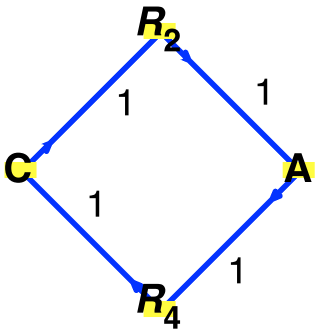
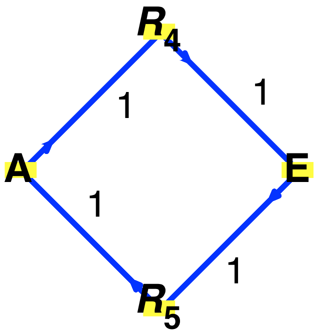
```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3123.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

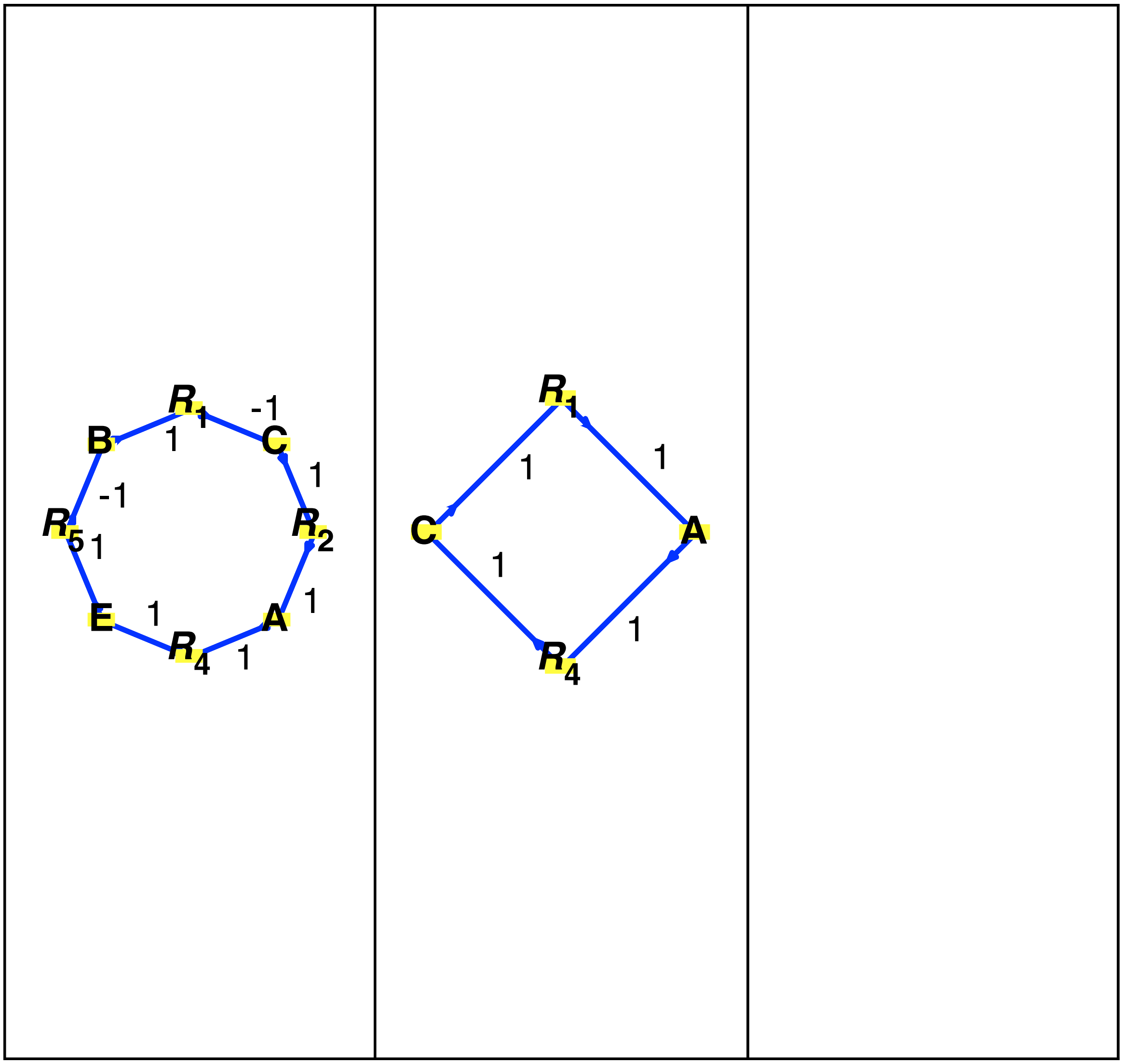
$$A := \begin{bmatrix} 1 & 1 & -1 & -1 & 1 \\ -1 & 0 & 1 & 0 & -1 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$







R_1 -1 C 1 B 1 -1 R_2 1 R_5 1 D A 1 R_3 1	R_2 1 1 C A 1 1 R_4	R_4 1 1 A E 1 1 R_5



>

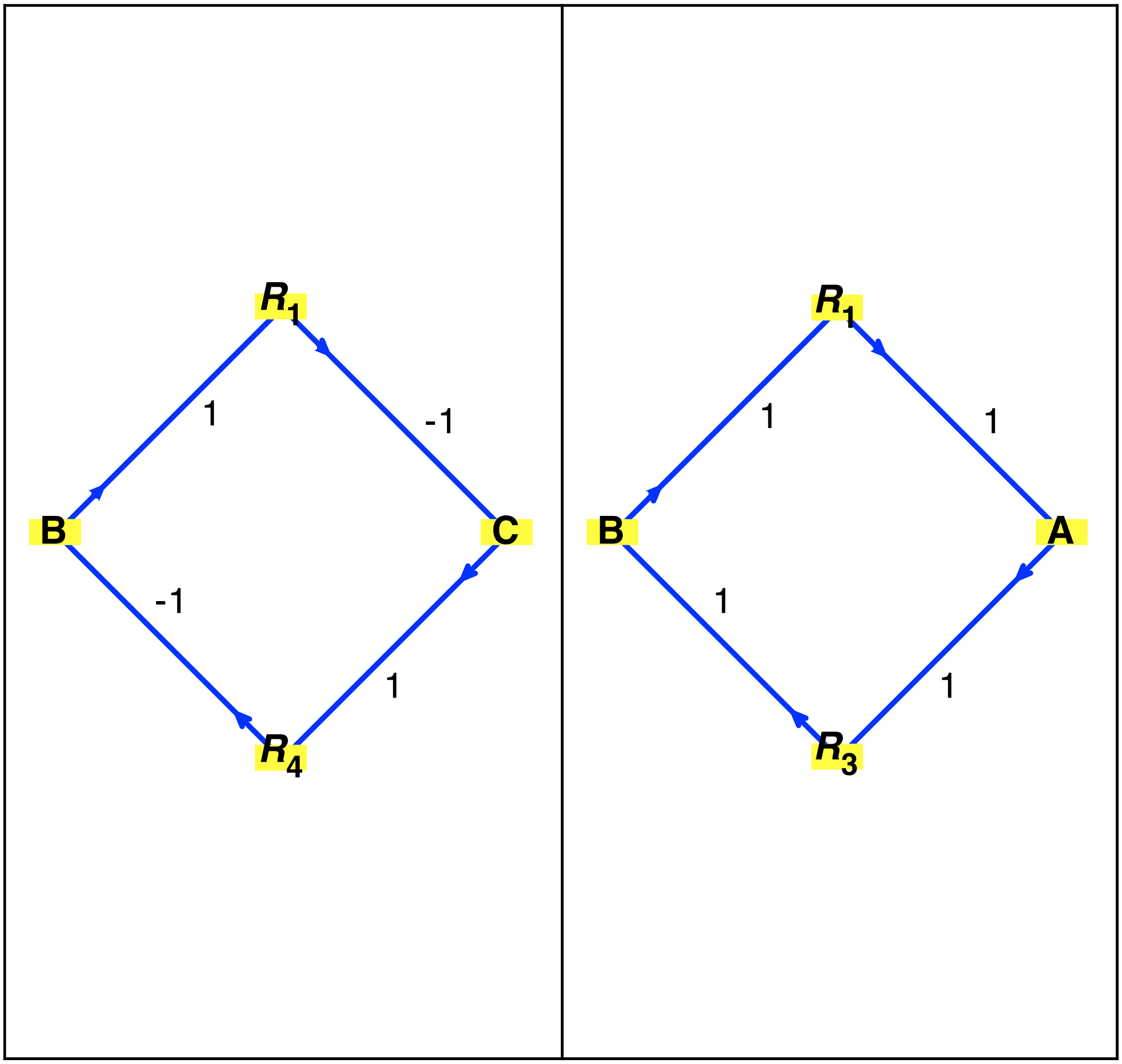
Meet both (?). NB: A dimer of B, C, D & E.

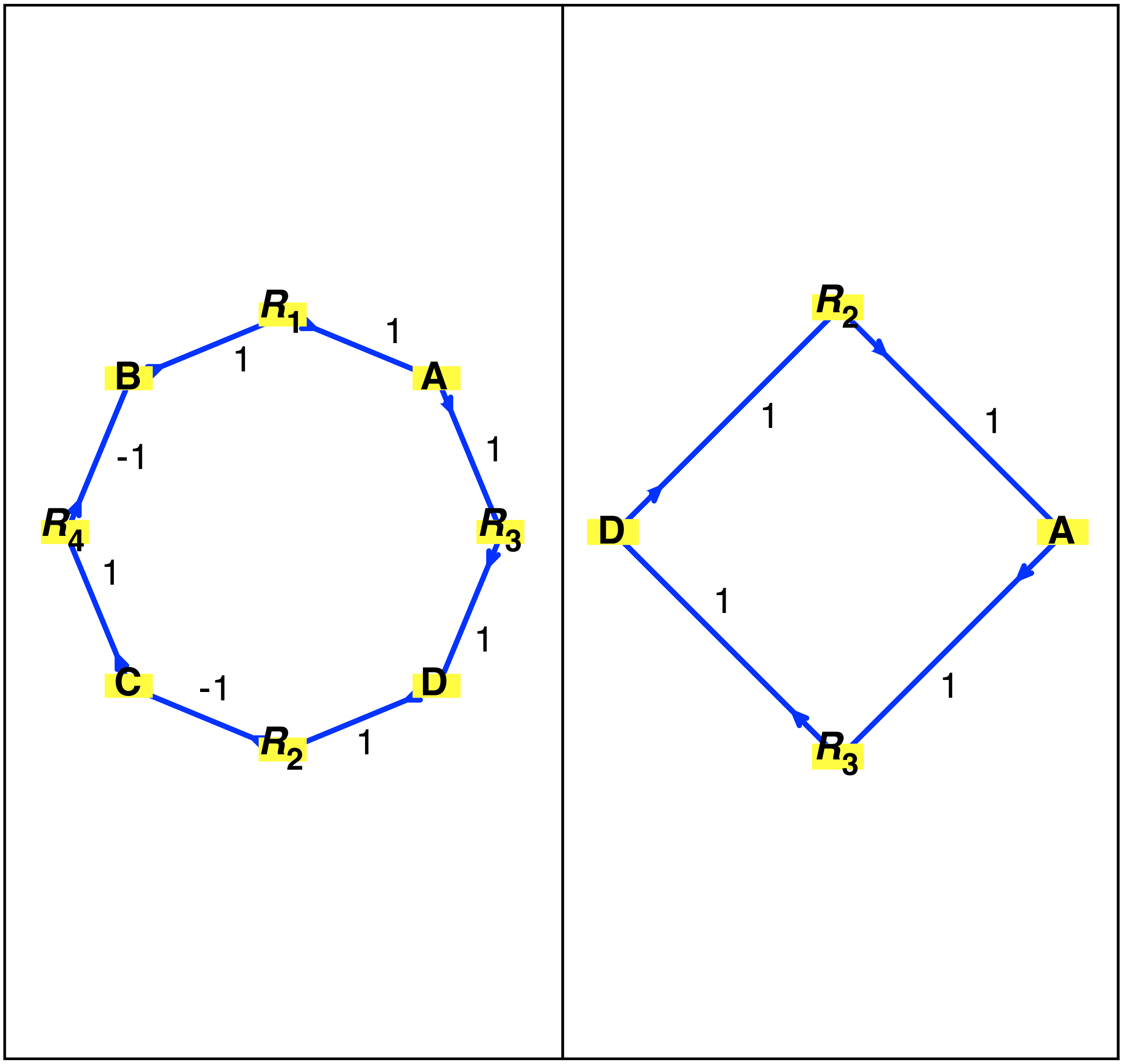
Network 3128 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$
$C + \mathrm{D} \rightarrow A$
$A \rightarrow B + \mathrm{D}$
$B + C \rightarrow E$
$E \rightarrow B + C$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3128.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

Meet both (?). NB: B is same as D, A is same as E.

Network 3555 (meet neither, not realistic) three competing reactions

$B + C \rightarrow A$

$\mathrm{D} \rightarrow B + C$

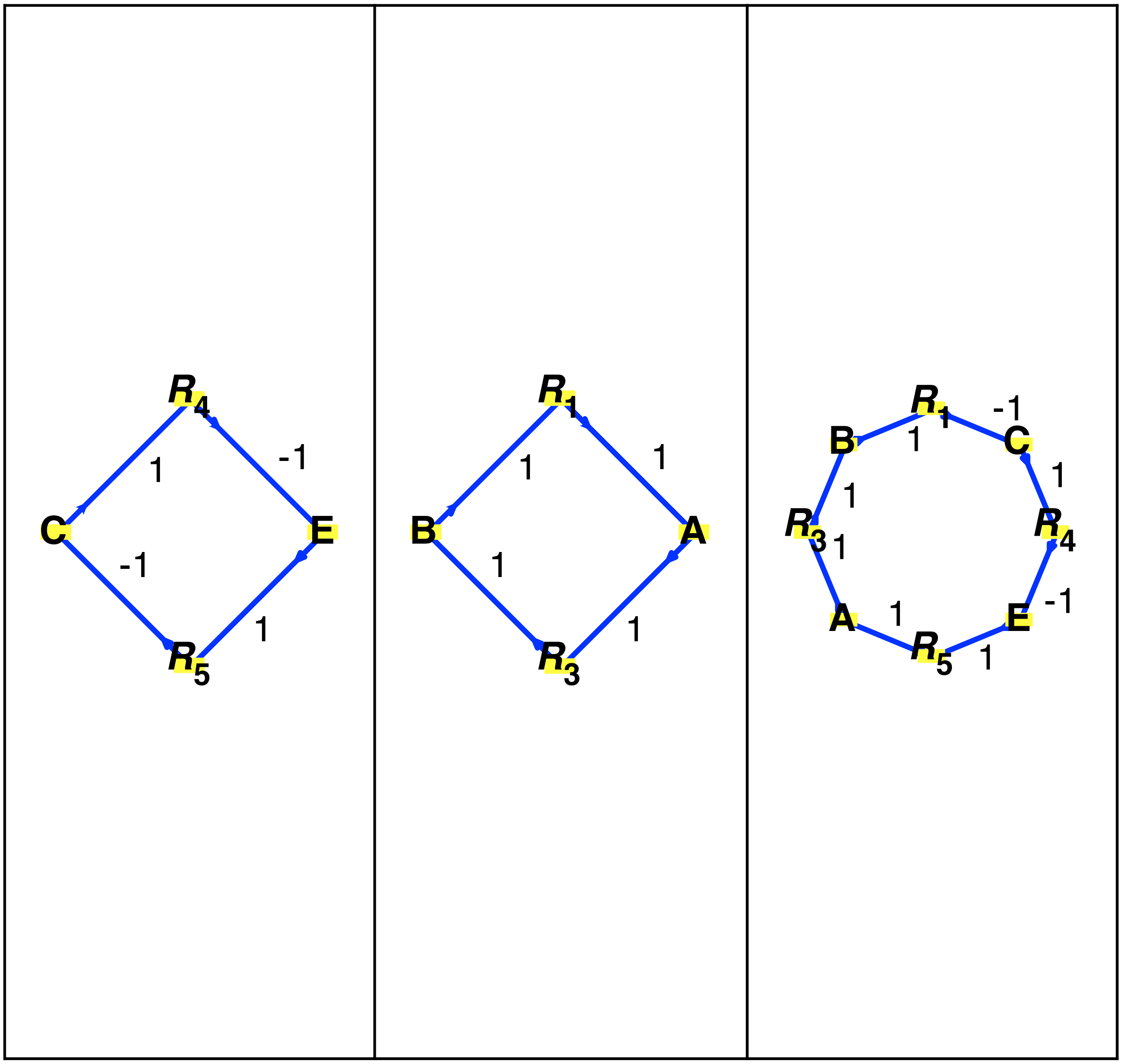
$A \rightarrow B + E$

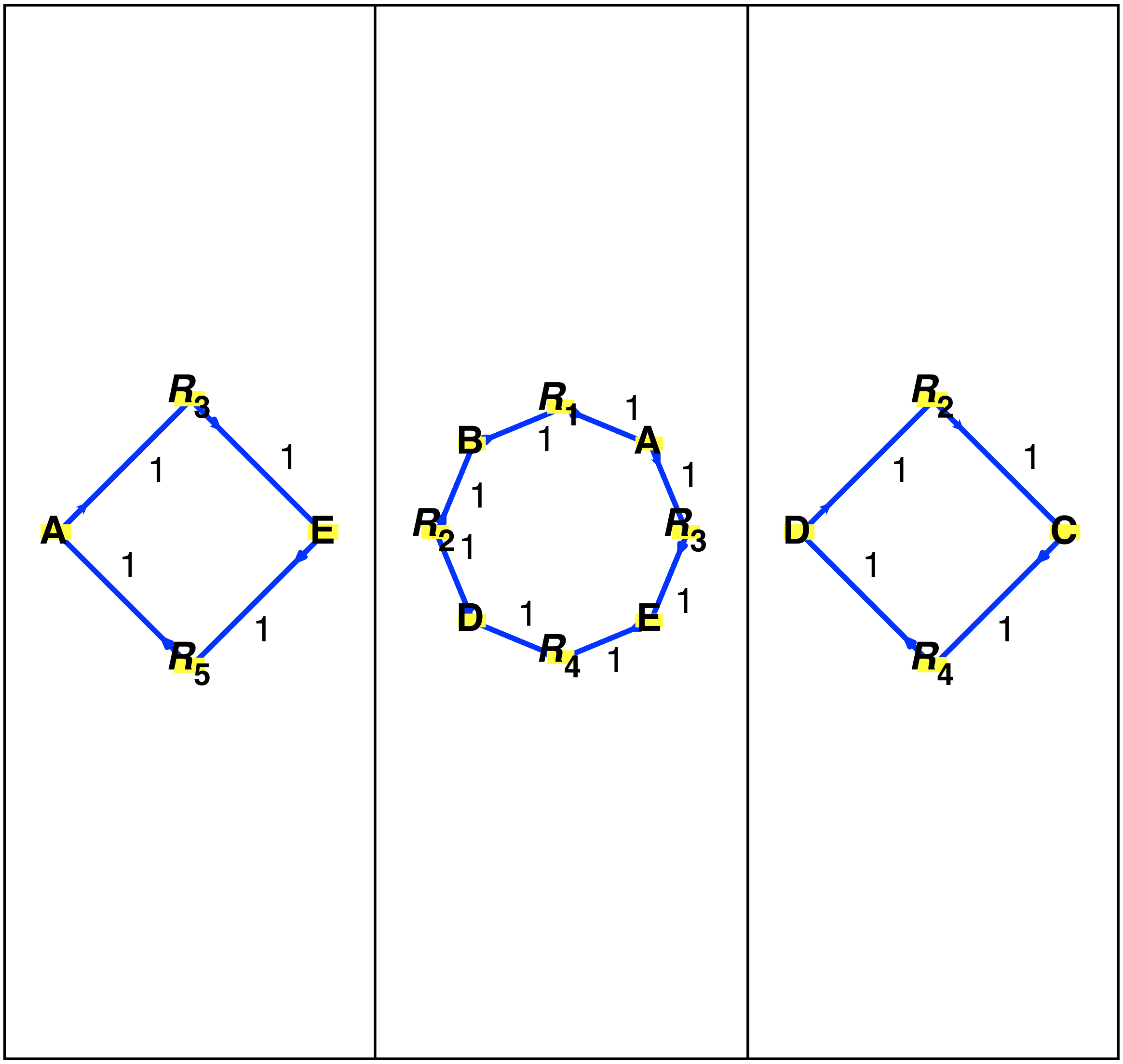
$C + E \rightarrow \mathrm{D}$

$C + E \rightarrow A$

```
> A := ImportMatrix(cat(uniquefolder, "/injectiveEx3_3555.csv")); Z := findZ(A) :
      selected := findloops(A, Z) : drawloops(selected, speciesord)
```

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





>

Meet both (?). NB: B is same as E, A is same as D.