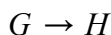
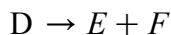
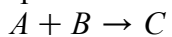


Enumerate reaction networks composed with heteromultimer and single transformations

November 2014

This particular code is to enumerate all possible reaction networks (with certain dimensions, $m \times n$ where n is species number m is reaction number) composed of three types of interactions:



Those are heterodimerization, disassociation and single transformation. However, we did not include $I \rightarrow 2J$ and $2K \rightarrow L$ those two type of elementary reaction here, we should implement those in near future.

With three types of elementary reactions above, we could construct a set of reaction networks, then we could use DSR graphs and/or bipartite graph to characterize those networks whether they are multistationary and has closed competition loop (as well as interchangeable competitors).

The main purpose of this document is to explain the procedures that how to construct and enumerate all possible reaction networks when given fixed reaction number and species number.

If a chemical reaction network has m reactions driven by n chemical species, we could have a stoichiometric matrix $N_{m \times n}$ with dimension

But before to go through such checking, we need to preclude situations that clearly not a complex balanced reaction network, by which mean it obeys the following three constraints:

0. Only allow elementary reactions described above, which is the starting point to construct the matrix; (NB: we don't consider birth-death process like $\emptyset \rightarrow X$ and $Y \rightarrow \emptyset$)

a). list all possible reaction vectors N_i ($i = 1 \dots n$) and select m of those into matrix (sequence does not matter)

The total number of reaction vectors for $G \rightarrow H$ is $\binom{n}{2} \cdot 2$ (since we need to consider the sides of two species in a reaction);

The total number of reaction vectors for $A + B \rightarrow C$ and $D \rightarrow E + F$ are both $\binom{n}{3} \cdot 3$ (same here);

We get $r = \binom{n}{2} \cdot 2 + \binom{n}{3} \cdot 3 + \binom{n}{3} \cdot 3$ number of reaction vectors, we store it in matrix $R_{r \times n}$, then we construct all the stoichiometric matrix by choosing m reaction vectors from $R_{r \times n}$ into $N_{m \times n}$, therefore we have total number of $\binom{r}{m}$ matrices to construct. Each constructed matrix $N_{m \times n}$ will go through balanced checking, mass conservation checking to become a valid stoichiometric matrix to go through further bistability check and competition check.

b). (Optional) We could further reduce the number of matrices when constructing them, in the set of constructed matrices there are huge number of matrices are isomorphic, which means any matrix in the set

with column permutation is another matrix in the set. (I did not prove this, I am thinking because I enumerated all possibility in each reaction vector which means no matter how to permute the columns (species) in a matrix, after permutation the matrix always falls in the same set). Now the set is closure for column (species) permutation, so does the set of $N_{(m-1) \times n}$, then if we construct the set of $N_{m \times n}$ from $N_{(m-1) \times n}$, we just need to add one in three reaction vectors (because with any column permutation we always get an isomorphic graph, the number of n th reaction vector is 3:

$[1, -1, 0, 0, \dots, 0]$, $[1, 1, -1, 0, \dots, 0]$, $[1, -1, -1, 0, \dots, 0]$, the position of 1 and -1 are not important), in this treatment, we can reduce the number of matrices from $\binom{r}{m}$ to $\binom{r}{m-1} \cdot 3$.

I am not sure about if this step is correct

c). Further, we could reduce the number to $\binom{r}{m-2} \cdot 43$ for $n = 5$, or $\binom{r}{m-2} \cdot 46$ for $n \geq 6$. 43 and 46 are numbers of the unique reaction patterns for two reactions between n species. I listed these reaction vectors manually, and implemented it in the code.

1. Mass conservation;

Based on the stoichiometric matrix $N_{m \times n}$, we can construct a vector of mass values \mathbf{m} , m_i is the mass value of species S_i . Then we have the equation $N\mathbf{m} = \mathbf{0}$, because in each reaction the mass of left (reactants) is equal to mass of right (products). We need to make sure \mathbf{m} is strictly positive.

a). firstly we check the rank of N (or linearly dependent), if $\text{Rank}(N) < m$, then it is linearly dependent, otherwise reject the matrix.

b). then calculate the nullspace basis of N . Then if the i th element in all basis is 0 or negative then m_i is 0 or negative. (This is not clear, may need some prove. When Maple compute the nullspace basis, it always return basis with 1s in e_j which means in the solution space $\sum x_j e_j$, x_j must be strictly positive.)

Actually a) and b) can be combined: if N is linearly independent, NullSpace will be empty. We could just exclude by examine nullspace basis.

Reference: Gevorgyan, A., Poolman, M. G., & Fell, D. A. (2008). Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24(19), 2245-2251.

2. Complex balanced: each species has at least one in flow and one out flow (this is very easy to check);

Exclude all matrices with any species (column) has no reactions involved, or all outflow (negative) or all inflow (positive)

Then we need to:

4. check competition: there are at least one species has two -1 and there is another -1 in each of the according reactions;

a). Get the N_- which only have the negative elements in N . In negative matrix, check the RowSum get indices I of -2 and check the ColumnSum get indices J of $Cs_i \leq -2$, if there are two indices $i, h \in I$ and one index $j \in J$ with which $N_{hj} = N_{ij} = -1$, $i = 1 \dots m$, then there is competition (of course the two competitors should be different, $c_i \neq c_j$).

5. check loop: take indices of competitors, do the network searching, find the loop from one to another and then from the other to this one.

this is fairly easy to understand, I use breadth-first search.

- first check if there are any species have more than two outflows (negative) then check if there are indeed two reactions with two species interact with another species. Then find the index of competitors.
- use breadth-first search to search routes between competitors.
- Note that to complete the competition loop (from bipartite graph), route from one competitor to the other need to avoid the competition reaction.

Cluster matrix into four categories: bistable with closed competition loop, bistable without closed competition loop, monostable with closed competition loop, monostable without closed competition loop.

All the procedure are implemented in the code. Any suggestions and corrections are more than welcome.

Initializations

```

[> restart :
[> interface( rtablesiz e = 400 ) :
[> with( ListTools ) :
[> with( LinearAlgebra ) :
[> with( VectorCalculus ) :
[> with( GraphTheory ) :
[> with( combinat ) :
[> with( ArrayTools ) :
[> Envsig n um 0 := 0 :

```

► **Functions for multistationality checking (execute before proceeding)**

► **Functions for constructing stoichiometric matrix and examine the existence of competition and closed loop.**

Testing

Here we test all functions:

$$\begin{aligned} & \text{> } V := \text{constrM}(5, 5) \text{ \# just count right matrices } \sim 2360s \text{ (40 mins) (9.29s to 2359.41s)} \\ & \quad \quad \quad V := [0, 0, 0, 0, 0, 0, 0, 9229, 3532880, 80, 86] \end{aligned} \quad (3.1)$$

```
> V := constrM(5, 5)
# also count injective extended matrices and export bistable matrices ~ 2716s
#(45 mins) (4621.73 s)
```

to 7337.76 s)

$$V := [65, 578, 87, 3977, 4707, 4522, 9229, 3532880, 80, 86] \quad (3.2)$$

> $V := \text{constrM}(5, 5)$

both check injectivity extended and competition loop ~2763.17s (46 mins) (15s to 2778.17s)

$$V := [65, 578, 87, 3977, 4707, 4522, 9229, 3532880, 80, 86] \quad (3.3)$$

> $V := \text{constrM}(5, 5)$

$$V := [65, 578, 87, 3977, 4707, 4522, 9229, 3532880, 80, 86] \quad (3.4)$$

> $A := \text{ImportMatrix}("5\text{species/injective/right_1479_injective1.csv"})$

$$A := \begin{bmatrix} 1 & 0 & -1 & -1 & 0 \\ -1 & -1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \quad (3.5)$$

> $\text{existcompetitionloop}(\text{Transpose}(A))$

$$\begin{bmatrix} 0 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} -2 & -3 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{matrix} 2 \\ 1 \\ 3e_{x1} + 4e_{x2} + e_{x3} + 2e_{x4} + 5e_{x5} \\ 2 \\ 2 \\ 2e_{x1} + 3e_{x2} + 2e_{x3} + 4e_{x4} + e_{x5} \\ 3 \end{matrix} \quad (3.6)$$

> $A := \text{ImportMatrix}("5\text{species/bistability/bistable_1.csv"})$

$$A := \begin{bmatrix} 1 & 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 \end{bmatrix} \quad (3.7)$$

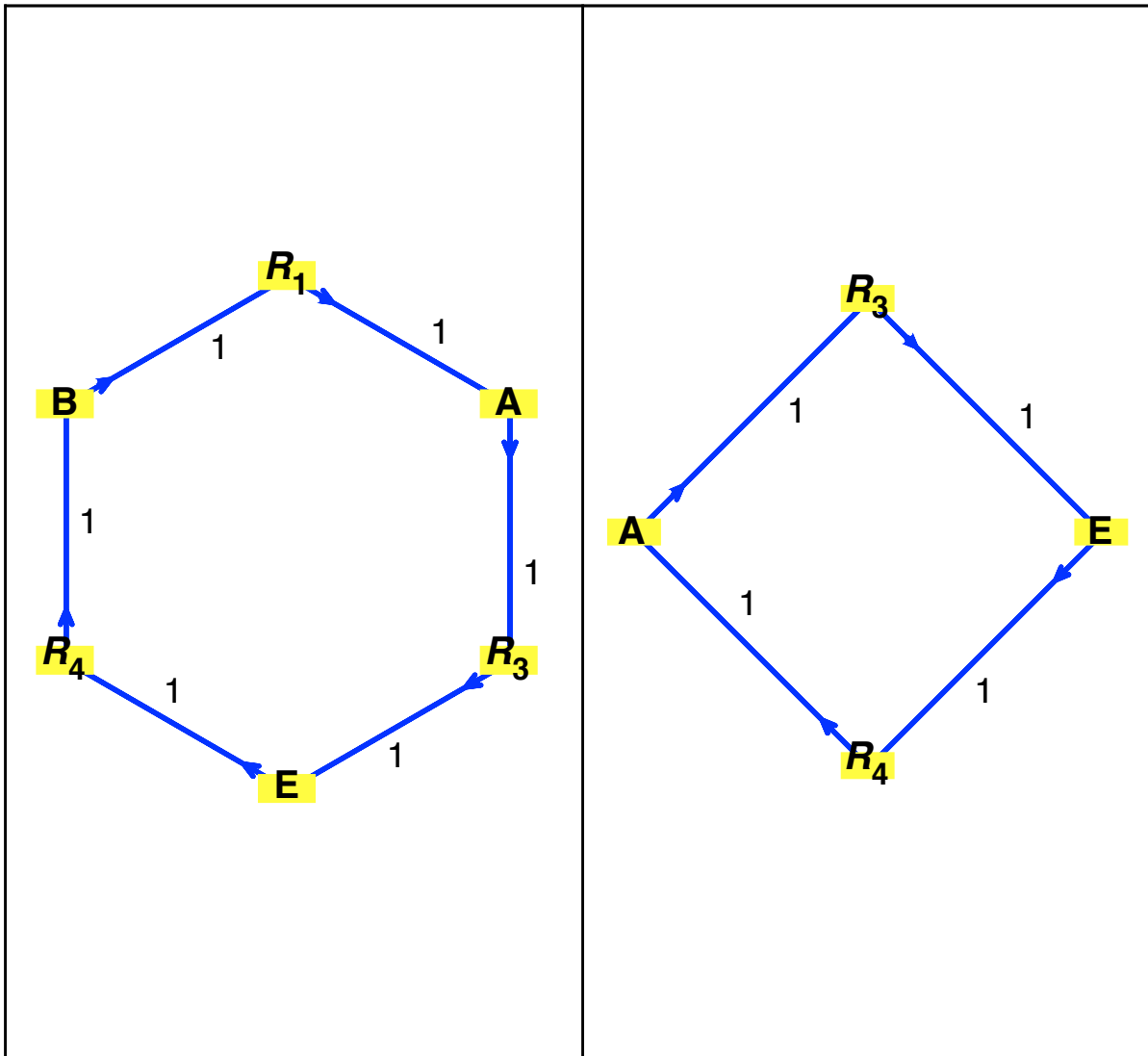
> $\text{existcompetitionloop}(\text{Transpose}(A))$

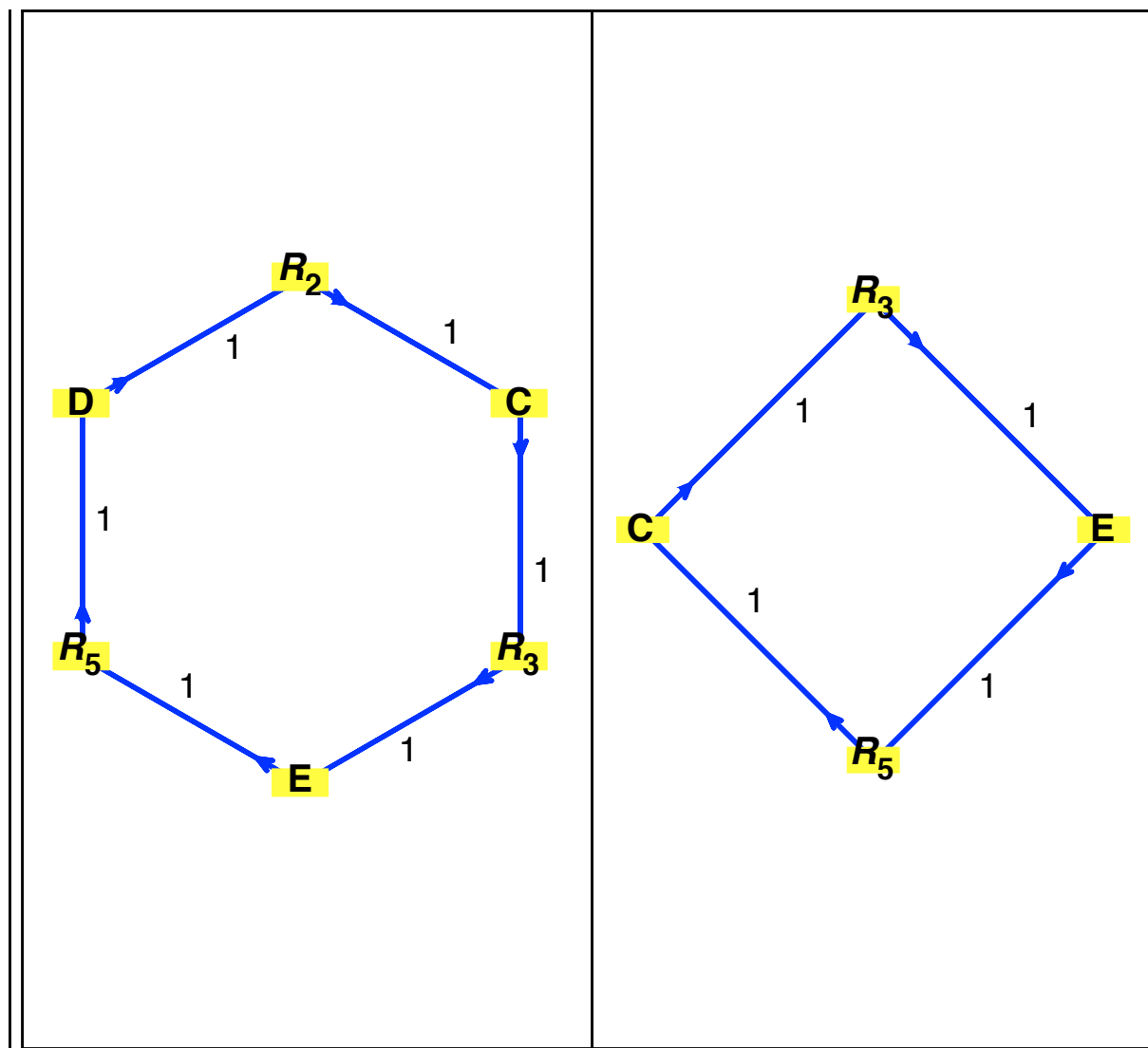
$$0 \quad (3.8)$$

```

> Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A) : selected
:= findloops( )
selected := [[[[R1, S1], 1], [[S1, R3], 1], [[R3, S5], 1], [[S5, R4], 1], [[R4, S2], 1], [[S2,
R1], 1]], [[[[R3, S5], 1], [[S5, R4], 1], [[R4, S1], 1], [[S1, R3], 1]], [[[[R2, S3], 1],
[[S3, R3], 1], [[R3, S5], 1], [[S5, R5], 1], [[R5, S4], 1], [[S4, R2], 1]], [[[[R3, S5], 1],
[[S5, R5], 1], [[R5, S3], 1], [[S3, R3], 1]]]]
(3.9)
> speciessord := ["A", "B", "C", "D", "E"] :
> drawloops(selected, speciessord)

```





Here is an example that without two species competing another species that could give rise bistability. It potentially implies that the competition is not necessary happening in between two species but also possible happening between two reactions.

```
> A := ImportMatrix("5species/bistability/bistable_10.csv")
```

$$A := \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \quad (3.10)$$

```
> existcompetitionloop(Transpose(A))
```

3

(3.11)

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

$$s := 3$$

(3.12)

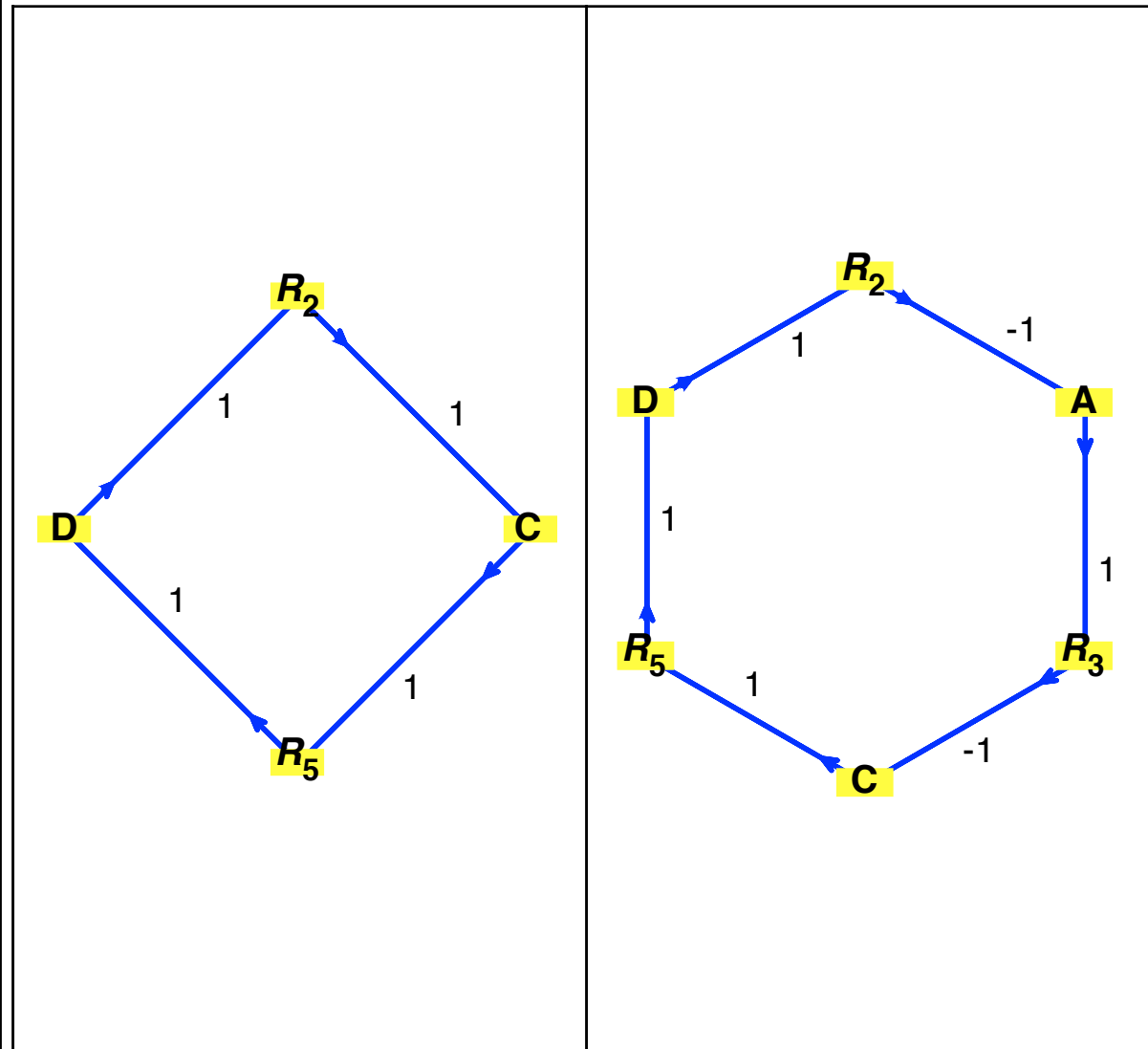
> *selected* := *findloops*()

selected := $\left[\left[\left[R_2, S_3 \right], 1 \right], \left[\left[S_3, R_5 \right], 1 \right], \left[\left[R_5, S_4 \right], 1 \right], \left[\left[S_4, R_2 \right], 1 \right] \right], \left[\left[R_2, S_1 \right], -1 \right], \right.$ (3.13)

$\left. \left[\left[S_1, R_3 \right], 1 \right], \left[\left[R_3, S_3 \right], -1 \right], \left[\left[S_3, R_5 \right], 1 \right], \left[\left[R_5, S_4 \right], 1 \right], \left[\left[S_4, R_2 \right], 1 \right] \right] \right]$

> *speciessord* := ["A", "B", "C", "D", "E"]:

> *drawloops*(*selected*, *speciessord*)



Here is an example with biological/chemical meaning, and also has the competition and closed loop also interchangeable loops.

Now we change a little bit to this network:

```

> A[1, 3] := 0 : A[1, 4] := 0 :
> A[4, 3] := -1 : A[4, 4] := 1 :
> A

```

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \quad (3.14)$$

```

> Z := findZ(A) : n := Dimension(A)[1] : m := Dimension(A)[2] : s := Rank(A)
s := 3

```

(3.15)

```

> selected := findloops( )
selected := [ ]

```

(3.16)

```

> speciessord := ["A", "B", "C", "D", "E"] :
> drawloops(selected, speciessord)
> isinjectiveextended(A)
1

```

(3.17)