SYSTEMS ANALYSIS
LECTURE 7
TOOLS

Zuzana Bělinová

Lecture overview

- Decision tables
- Cluster analysis
- Graph transmission

Decision tables

Decision tables

- Tool for modeling sets of logical terms
- Similar to if-then rules
- Work with several conditions at the same time
- Advantages
  - Possibility of modelling complex alternative processes
  - Suitable for algorithms with many decisions
  - Easily computerized

Decision tables - structure

- 4 quadrants

List of conditions	Combinations of conditions
List of actions	Combination of actions

- DTs describe which actions are to be done under particular combination of conditions
- In the „Combination of conditions“ quadrant different **possible** combinations of conditions are stated and to each combination there are in the appropriate column in the quadrant „combination of actions“ - markes actions to be done

Decision tables - creating

- 1. Identify conditions and their values
- 2. Identify possible actions
- 3. Enter combinations of conditions
- 4. Define actions for each combination of conditions
- 5. Verify the table and if possible, simplify the table

Example

DS: Should I attend the lecture?				
Was the last lecture useful?	0	0	1	1
Do I have other duties?	0	1	0	1
Stay home and do nothing	1	0	0	0
Work on other duties	0	1	0	1
Attend the lecture	0	0	1	1

- One combination can have more actions
- One action can be done for more combinations of conditions

Types of decision tables

- According the inputs
  - With limited-entry (binary input)
  - With extended entry (each condition has more possible results)
    - It is possible to put down the different values (intervals) in the combinations of conditions
    - Or every condition in the first quadrant has as many formulations, as there are possible inputs
  - With mixed inputs

Example – table with extended entry

DS: Evening activity				
How much many I have at disposal?	<50	50-200	200-500	>500
Stay at home	1	0	0	0
Go to pub	0	1	0	1
Go to a theater	0	0	1	0
Invite friends	0	0	0	1

DS: Evening activity				
I have at disposal <50 Kč	1	0	0	0
I have at disposal 50-200 Kč	0	1	0	0
I have at disposal 200-500 Kč	0	0	1	0
I have at disposal >500 Kč	0	0	0	1
Stay at home	1	0	0	0
Go to pub	0	1	0	1
Go to a theater	0	0	1	0
Invite friends	0	0	0	1

Example – table with contradictory entry

- e.g. choosing signal plan for traffic control

Day: public holiday	1	0	0	0	….	0
Monday	-	1				
Tuesday	-		1			
Wednesday	-			1		
….	-					
Sunday	-					
Use signal plan A	1					
B		1				
C			1	1		

Types of decision tables

- According cohesion with other DT
  - Open – there are more DT connected (using links from one to another)
  - Closed – one self-sustaining table
- Full tables – with limited binary entry, number of rules is 2^n
  - More often there are less rules because of redundancy or contradictory

Example – open table

DT 1				
Do you have ticket?	0	0	1	1
Do you have cell phone?	0	1	0	1
Go on foot	1	0	0	0
Send SMS	0	1	0	0
Go to DT 2	0	1	0	0
Return from DT 2	0	1	0	0
Get on the vehicle	0	1	1	1

DT 2		
Have you received the SMS ticket?	0	1
Wait - Repeat DT 2	1	0
Return to DT1 and continue	0	1

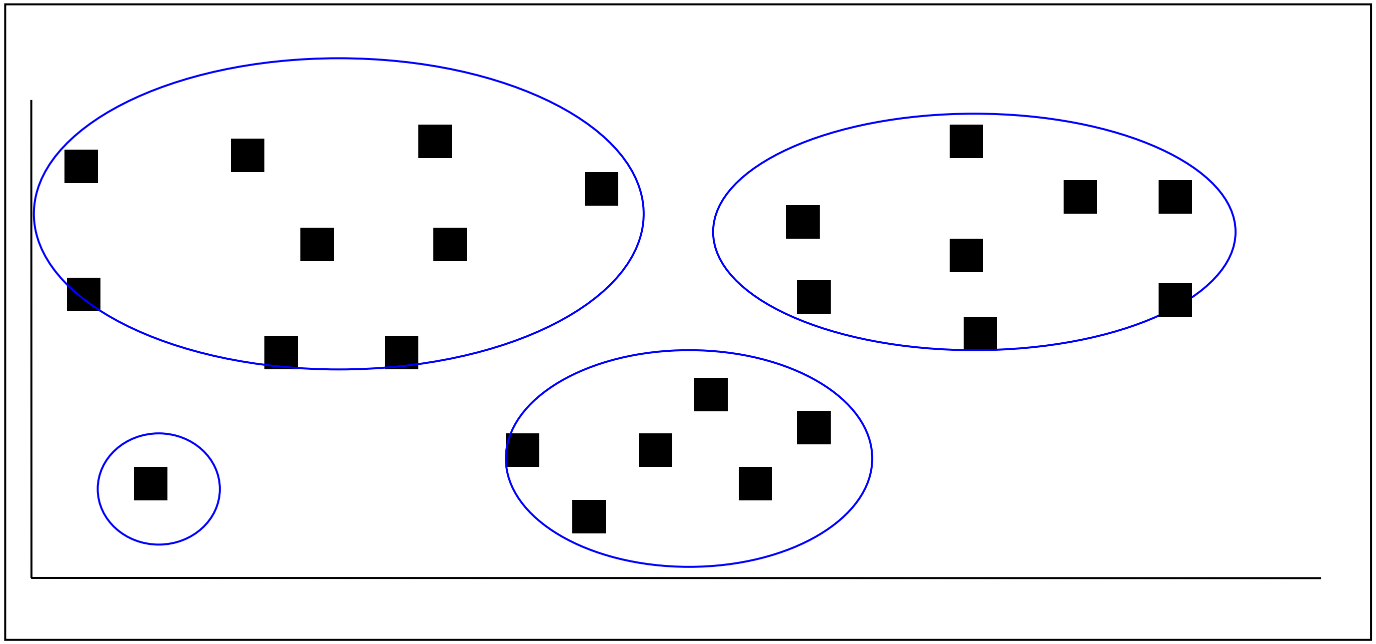
Cluster analysis

Cluster Analysis - CA

- Method for clustering similar objects
- Combines mathematical and expert attitudes

- Usage in the Systems analysis, e.g.
  - System identification
  - Semantic decompositon

Example for 2D position



Typical procedure

- Choice of objects (elements) objektů / prvků
- Choice of measured characteristices and assignment of weights to these characteristices
- Choice of measure of similarity of elements – metrics
- Choice of clustering method
- If needed, determination of number of clusters(either based on expertise or using full hierarchical analysis and setting of rules for choice of optimal level)
- Calculation
- Results interpretation

Cluster analysis methods

- E.g.:
  - K-means
  - ISODATA
  - …

Graph transmission – Mason-Truxal rules

Graph transmission

- Procedure of transferring of system of linear equation into a graph (the system matrix must be partially empty) – izomorfism between oriented graph and system of linear equations

- It serves for simple solving of system transmission (proportion of output and input ($x_i \rightarrow x_j$)

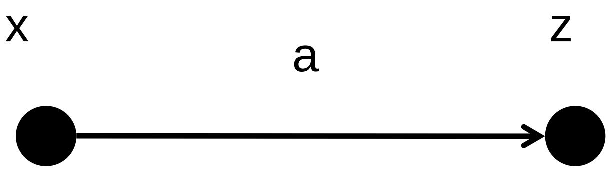
Graph transmission

- Usage
  - regulaion
  - Control technique
  - Cybernetics
- Topological or algebraical form

- Topological form – Mason-Truxal rules

How to transfer the equations into graph

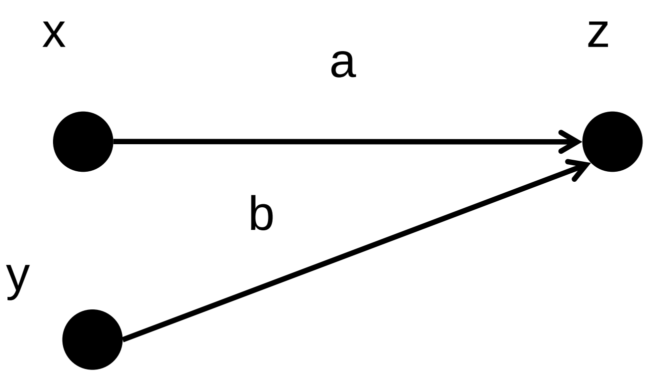
- z=a*x



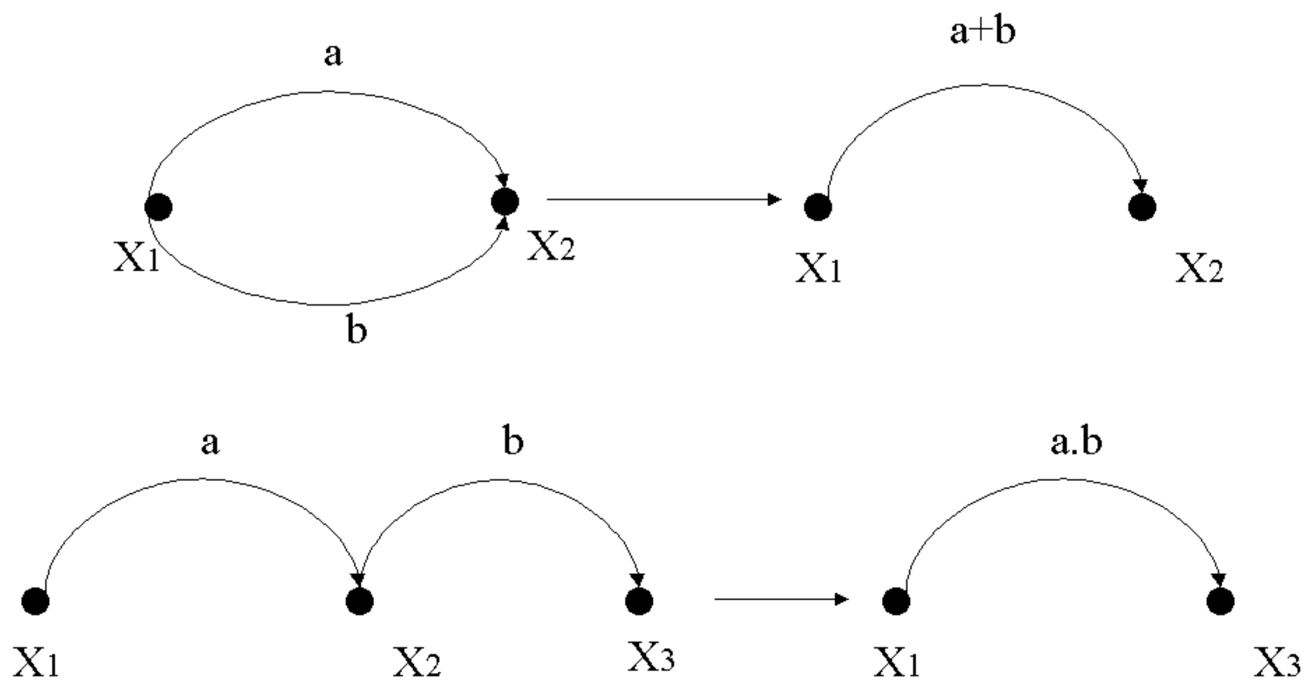
x, y, z – variables

a, b - constants

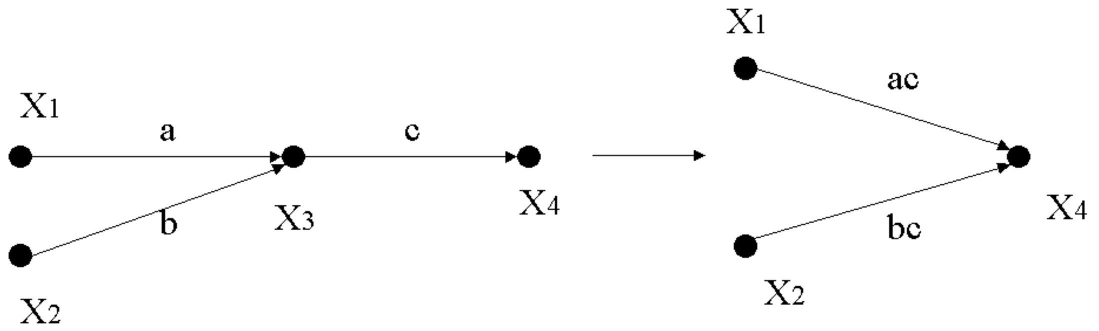
- z=a*x+b*y

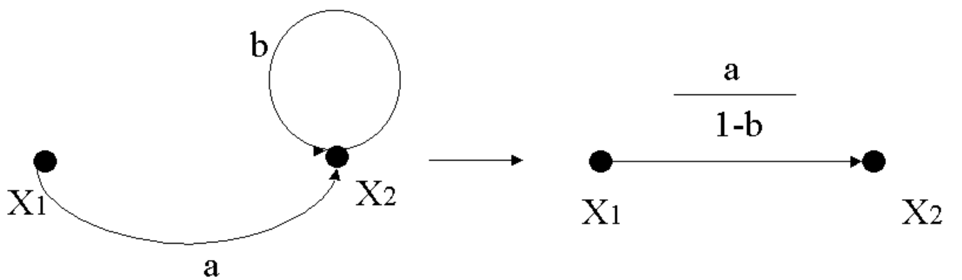


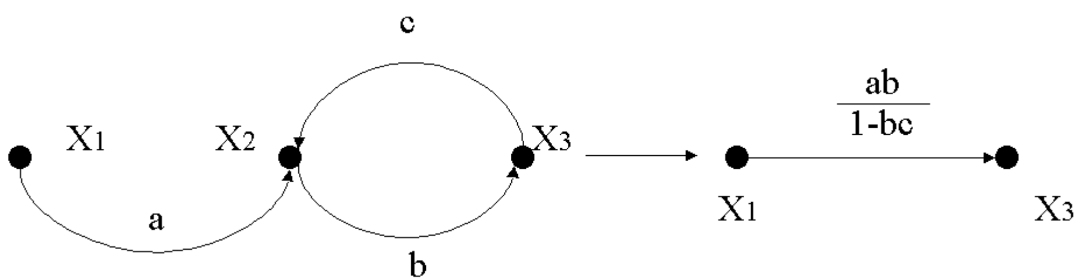
The Mason-Truxal rules I



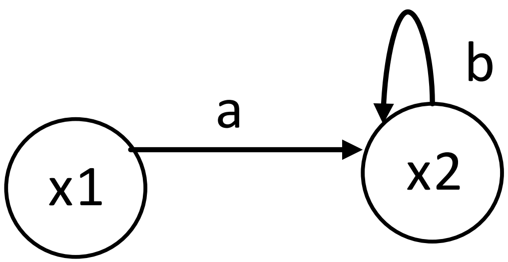
The Mason-Truxal rules II







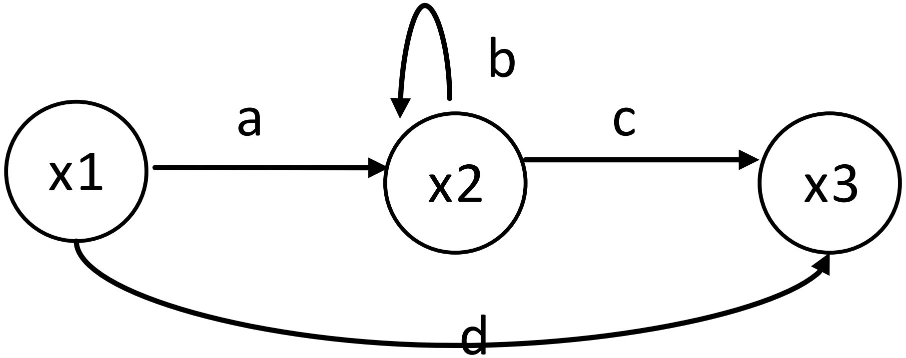
Feedback rules - deduction



- x2=a*x1+b*x2 x1
- x2=a*x1+b*x2 x2-b*x2=a*x1
- x2(1-b)=a*x1
- x2/x1=a/(1-b)

Example

$x_2 = a*x_1 + b*x_2$
$x_3 = c*x_2 + d*x_1$



$$\frac{x3}{x1} = \left(\frac{a}{1-b} * c\right) + d$$

- Verification

$$x_2 = \frac{a * x_1}{1-b}$$

$$x_3 = c * \frac{a * x_1}{1-b} + d * x_1$$

$$\frac{x3}{x1} = \frac{a * c}{1-b} + d$$

Graph transition - terms

- **Path** – continuous succession of branches, no node is encountered more than once
- **Loop** – simple closed path,
- **Self loop –** loop with no other node than the origin=destination node

Graph transmission – basic rules

- Path transmission = product (multiplication) of all branch transmissions along a single path
- Transmission of parallel paths = sum of the path trnasmissions of all the possible paths between two nodes
- Absorbing node (not a loop node) – it must be included in all the direct paths going through it
- Absorbing several self-loops of one node into one – sum of the self-loops
- Absorbing self-loop with transmission b – transmission **of every incoming branch** is divided by (1-b)

Graph transmission – procedure

- Absorbing nodes with just one input or output
- Absorbing all paralel branches (with the same orientation)
- Absorbing self-loops
- Absorbing loop node – all direct and self paths must going through the node must be assessed

References

- Graph transmission
  - D. Eppinger Murthy V. Nukala Daniel E. Whitney: Generalized Models of Design Iteration Using Signal Flow Graphs Steven. MIT, Sloan School of Management, 1996. Accesible from: https://core.ac.uk/download/pdf/4380258.pdf (04/2019)