Research Group on Rough Sets and Petri Nets
Chair of Computer Science, Rzeszów University
POLAND

WHETHER ROUGH SETS CAN BE USEFUL IN PROCESS MINING?

(Introduction to a panel discussion)

Andrzej Skowron & Zbigniew Suraj

Warsaw University, Poland
Rzeszów University, Poland

The Sixth International Conference on Rough Sets and Knowledge Technology, RSKT 2011,

October 9-12, 2011, Banff, Canada

OUTLINE

- Motivations
- Problems and Known Solutions
- Important Questions
- Available Software
- Concluding Remarks & Further Work

MOTIVATIONS

- **In a lot of cases, data are generated by concurrent processes.**

- **Discovering concurrent system models is essential from the point of view of understanding the nature of modeled systems as well as explaining their behaviors.**

MOTIVATIONS

- **Elaborated methods during the research can be used, i.e., in the synthesis and analysis of concurrent systems, data analysis, forecasting, etc.**

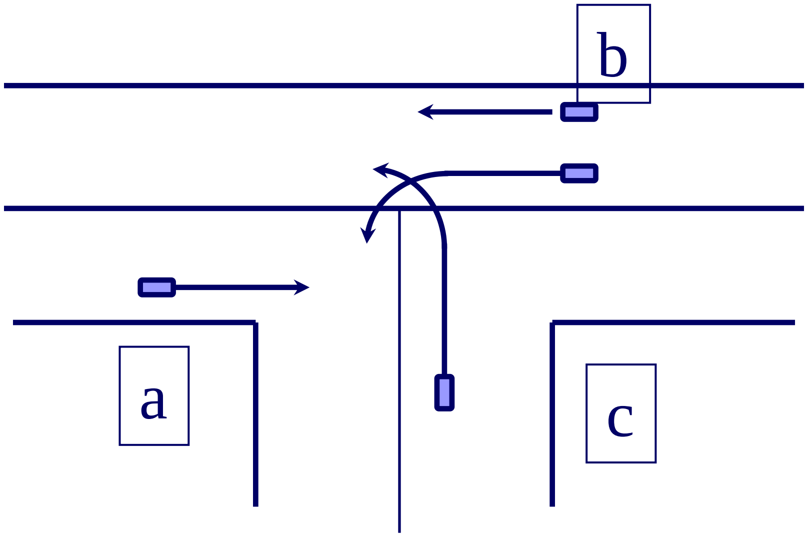
PROBLEMS AND KNOWN SOLUTIONS.

A SURVEY

HOW TO SPECIFY OF CONCURRENT SYSTEMS BY INFORMATION SYSTEMS?

- Pawlak, Z.: Concurrent versus sequential the rough sets perspective. ***Bulletin of the EATCS 48*** (**1992**) 178—190

EX. 1: LIGHT CONTROL



Three local processes

	a	*b*	*c*
u_1	1	1	0
u_2	0	2	0
u_3	0	0	2

Three global states

u_1, u_2, u_3 - global states

a, b, c - movement directions

0, 1, 2 - light colors (interpreted respectively as: red, green, green arrow)

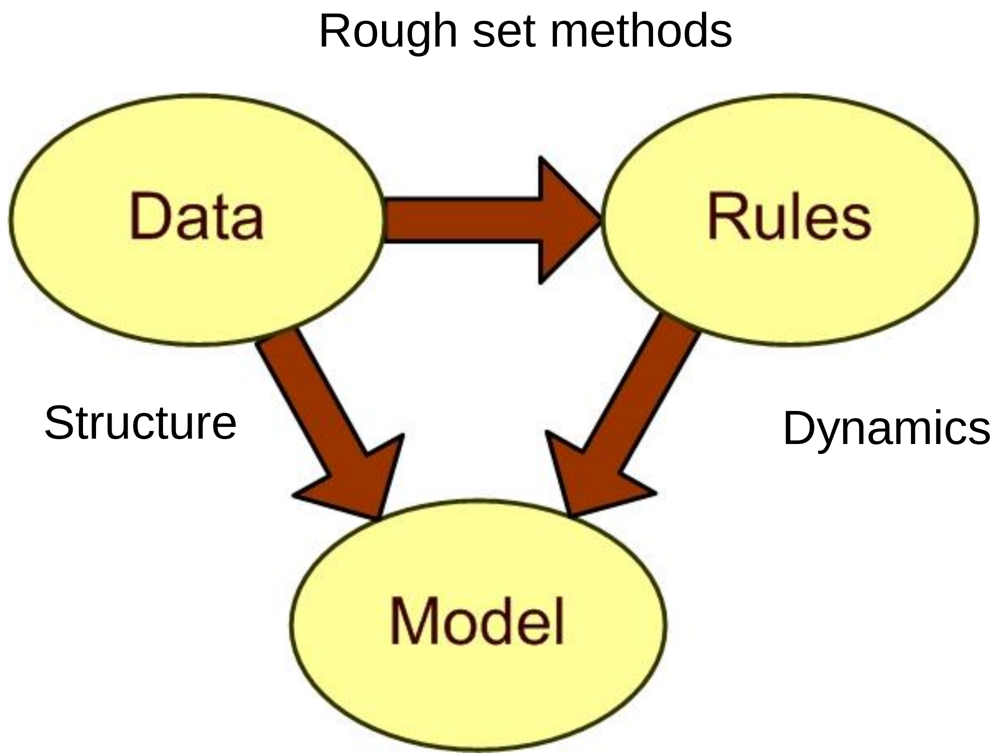
HOW TO DESIGN CONCURRENT SYSTEMS FROM SPECIFICATIONS BY INFORMATION SYSTEMS?

- Skowron, A., Suraj, Z.: Rough sets and concurrency. ***Bull. Pol. Acad. Sci. 41(3)*** (**1993**) 237—254
- Suraj, Z.: Rough set methods for the synthesis and analysis of concurrent processes. ***Studies in Fuzziness and Soft Computing 56***, Springer (**2000**) 379—488
- Pancerz, K., Suraj, Z.: Rough Sets for Discovering Concurrent System Models from Data Tables, ***Rough Computing. Theories, Technologies, and Applications***, IGI Global, **2008**, 239—268

A DIAGRAM OF PROBLEM SOLVING

DATA REPRESENTATION

KNOWLEDGE REPRESENTATION



PETRI NETS AS A CONCURRENT MODEL

MAIN IDEA

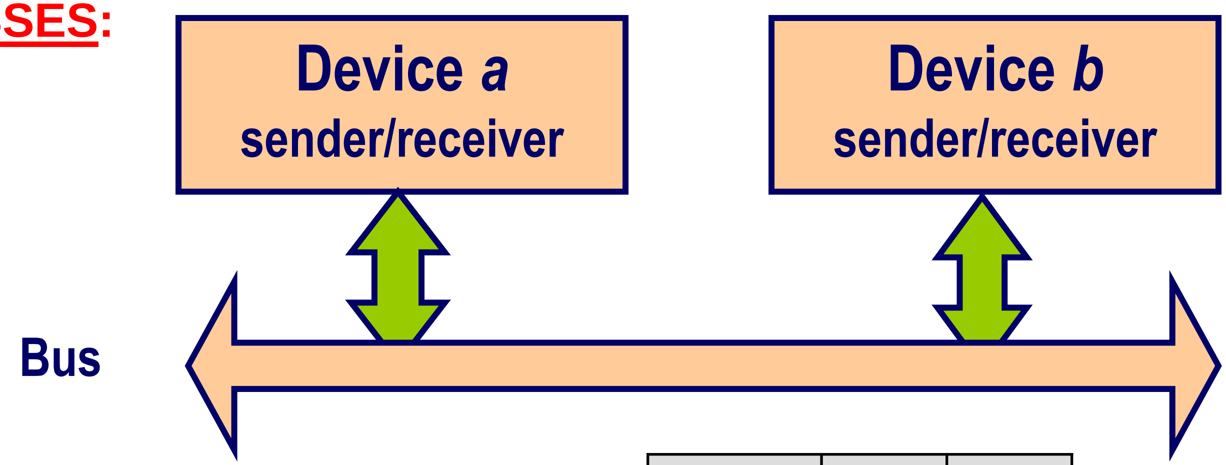
- Dependencies defined by attributes are conditions for coexistence of local states in global states
- One can use the existing methods for generating rules representing such dependencies (e.g., based on reducts or not)
- The set of rules is treated as knowledge representation for a given information system and defines its maximal extension, i.e., the set of global states consistent with all rules
- One can develop algorithms for designing, e.g., Petri nets defining maximal extensions of information systems (data tables)

FORMALLY:

- **Given information system A**
- **Define a theory Th(A) of A** (consisting of a set of rules describing dependencies in A)
- **Th(A) defines the maximal extension of A**
- **Construct a Petri net consistent with Th(A)**

EX. 2: COMMUNICATION SYSTEM

PROCESSES:



ACTIONS:

- **0 – sending**
- **1 – receiving**
- **2 – disconnecting**

U\A	*a*	*b*
u1	0	1
u2	1	0
u3	0	2
u4	2	0

DATA TABLE

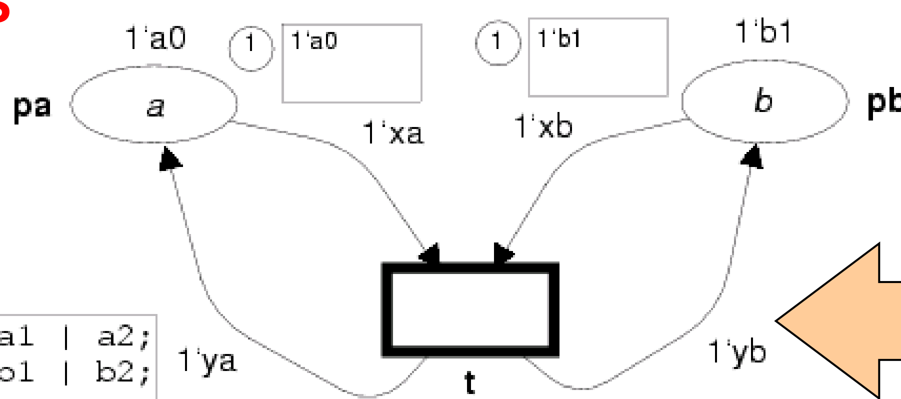
SOLUTION 1

U\A	a	b
u1	0	1
u2	1	0
u3	0	2
u4	2	0

IF *a1* THEN *b0*
IF *a2* THEN *b0*
IF *b1* THEN *a0*
IF *b2* THEN *a0*

(*a0* AND *b0*) OR (*a0* AND *b1*) OR (*a0* AND *b2*) OR (*a1* AND *b0*) OR (*a2* AND *b0*)

SYNCHRONOUS CONCURRENT MODEL:




```
color a = with a0 | a1 | a2;
color b = with b0 | b1 | b2;
var xa, ya : a;
var xb, yb : b;
```

[(ya=a0 andalso yb=b0) orelse (ya=a0 andalso yb=b1) orelse (ya=a0 andalso yb=b2) orelse (ya=a1 andalso yb=b0) orelse (ya=a2 andalso yb=b0)]

U\A	a	b
u1	0	1
u2	1	0
u3	0	2
u4	2	0
u5	0	0

hidden global state

U\A	a	b
u1	0	1
u2	1	0
u3	0	2
u4	2	0

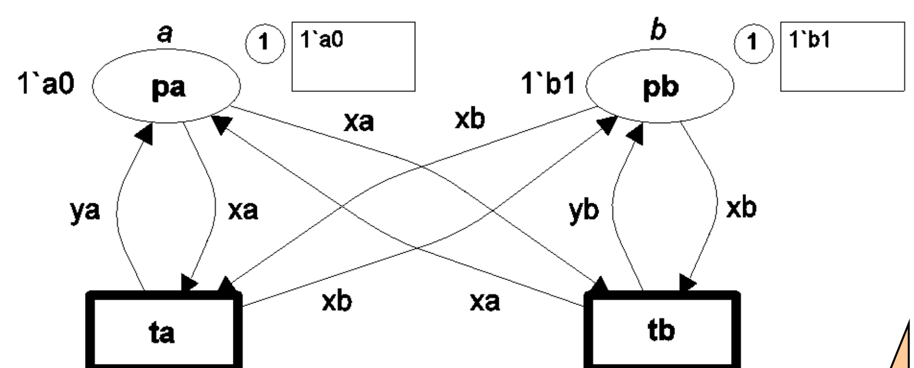
SOLUTION 2

IF *a1* THEN *b0*
IF *a2* THEN *b0*
IF *b1* THEN *a0*
IF *b2* THEN *a0*

(*a0* AND *b0*) OR (*a0* AND *b1*) OR (*a0* AND *b2*)
OR (*a1* AND *b0*) OR (*a2* AND *b0*)

ASYNCHRONOUS CONCURRENT MODEL:

Process a: **Process b:**



[not(((xb=b1) andalso (ya=a2))
orelse ((xb=b1) andalso (ya=a1))
orelse ((xb=b2) andalso (ya=a2))
orelse ((xb=b2) andalso (ya=a1)))]

[not (((xa=a1) andalso (yb=b2))
orelse ((xa=a1) andalso (yb=b1))
orelse ((xa=a2) andalso (yb=b2))
orelse ((xa=a2) andalso (yb=b1)))]

```
color a = with a0 | a1 | a2
color b = with b0 | b1 | b2
var xa, ya : a;
var xb, yb : b;
```

ADVANTAGES

- Complex Petri nets can be generated automatically from their specification by data tables
- Petri net can be adaptively modified with changes of data

IMPORTANT QUESTIONS

- **Which kinds of rules should be used (e.g., also non-deterministic, probabilistic)?**
- **How to characterize the expressibility of different rule sets?**
- **How to extend the approach by adding information on transition relation or temporal dependencies?**

1. Moshkov, M., Skowron, A., Suraj, Z.: Maximal Consistent Extensions of Information Systems Relative to Their Theories, ***Information Sciences*** 178/12 (2008), 2600--2620.
2. Delimata, P., Moshkov, M., Skowron, A., Suraj, Z.: ***Inhibitory Rules in Data Analysis***. *A Rough Set Approach*, Springer, 2009.

OTHER PROBLEMS RELATED TO THE SYNTHESIS PROBLEM

1. The Decomposition Problem

2. The Reconstruction Problem

3. The Prediction Problem

1. THE DECOMPOSITION PROBLEM

INPUT: **A given data** (an information system) **generated by a system of concurrent processes, and concurrent model constructed on the base the given data** (as a result of the synthesis algorithm).

OUTPUT: A family of components (subsystems) together with the links (rules) binding those components which are sufficient to build the original concurrent model such that model global states are consistent with the extracted knowledge from the given data.

2. THE RECONSTRUCTION PROBLEM

INPUT: A given data table representing the specification of concurrent process model obtained by the synthesis algorithm, and a new data representing a new specification of the modelled system.

OUTPUT: A plan (algorithm) of the reconstruction of a given concurrent process model satisfying the new specification represented by a new data.

3. THE PREDICTION PROBLEM

INPUT: A given concurrent model described by temporal data (ordered in time).

OUTPUT: A set of prediction rules which can be used to predict future changes of the model.

OUR SOFTWARE

Tool	Use
CPNetwork	Design and simulation of coloured Petri net models http://rsds.univ.rzeszow.pl (Option: **Software**)
ROSECON	Discovering concurrent models from data tables http://rsds.univ.rzeszow.pl (Option: **Software**)

CONCLUDING REMARKS

- **Constructed concurrent models can be useful for designers and analysts to:**

  - *determine some properties (concerning structure and behaviour) of modelled systems*

  - *extract new knowledge about modeled systems*

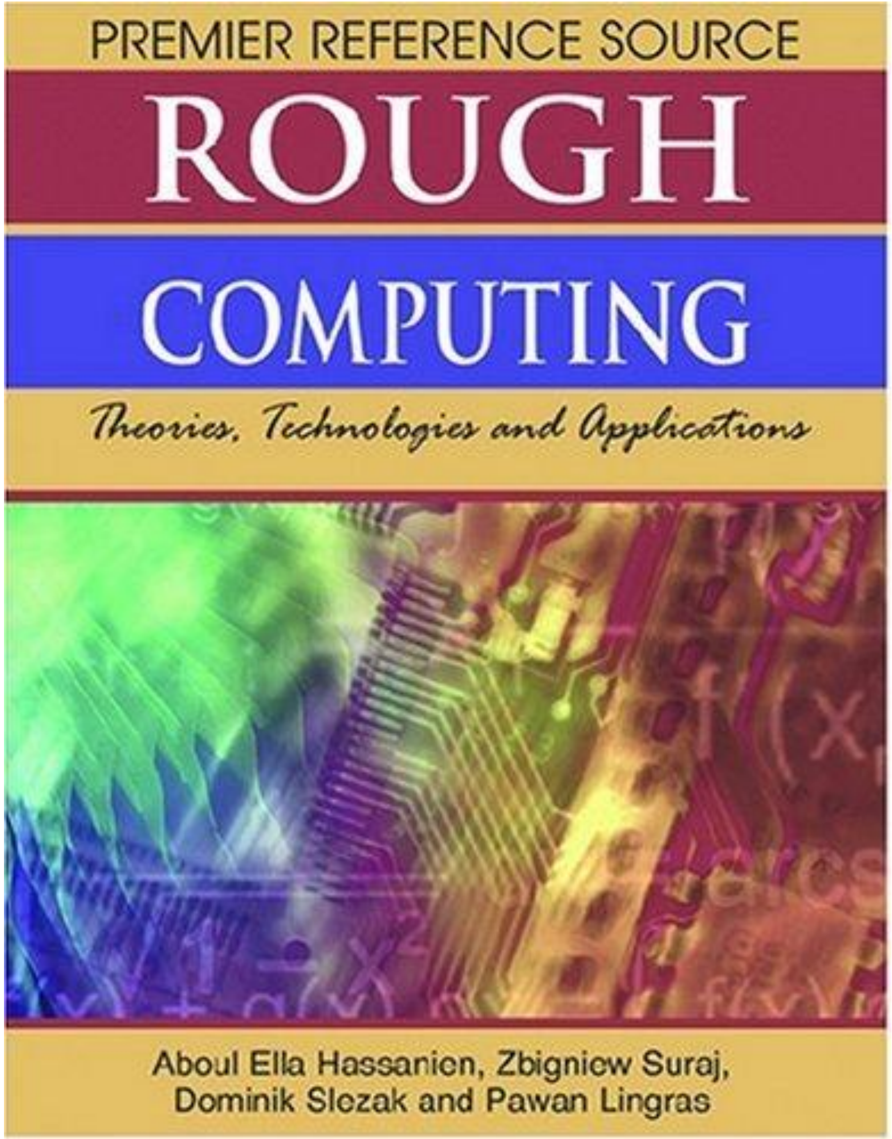
  - *verify their descriptions or specifications*

FURTHER WORK

- To consider the prediction problem of property changing net models in non-stationary data systems
- To discover modular and hierarchical concurrent models in data
- To discover of dynamical models from data based on rough granular calculus of changes and interactions
- To consider a new synthesis problem formulated as follows:

A NEW SYNTHESIS PROBLEM

- **Given the hierarchical information system A**
- **Define theory Th(A) of A (consisting of a set of rules describing temporal and spatial dependencies in A)**
- **Th(A) defines the maximal extension of A**
- **Construct a concurrent system consistent with Th(A)**



Information Science REFERENCE, Hershey – New York, 2008

The series *Studies in Computational Intelligence* (SCI) publishes new developments and advances in the various areas of computational intelligence – quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life science, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems and hybrid intelligent systems. Critical to both contributors and readers are the short publication time and world-wide distribution – this permits a rapid and broad dissemination of research results.

This monograph is devoted to theoretical and experimental study of inhibitory decision and association rules. Inhibitory rules contain on the right-hand side a relation of the kind "attribut does not equal value". The use of inhibitory rules instead of deterministic (standard) ones allows us to describe more completely information encoded in decision or information systems and to design classifiers of high quality.

The most important feature of this monograph is that it includes an advanced mathematical analysis of problems on inhibitory rules. We consider algorithms for construction of inhibitory rules, bounds on minimal complexity of inhibitory rules, and algorithms for construction of the set of all minimal inhibitory rules. We also discuss results of experiments with standard and lazy classifiers based on inhibitory rules. These results show that inhibitory decision and association rules can be used in data mining and knowledge discovery both for knowledge representation and for prediction. Inhibitory rules can be also used under the analysis and design of concurrent systems.

The results obtained in the monograph can be useful for researchers in such areas as machine learning, data mining and knowledge discovery, especially for those who are working in rough set theory, test theory, and logical analysis of data (LAD). The monograph can be used under the creation of courses for graduate students and for Ph.D. studies.

ISSN 1860-949X

ISBN 978-3-540-85637-5

9 783540 856375

springer.com

Delimata et al.

Inhibitory Rules in Data Analysis

Studies in Computational Intelligence 163

Pawel Delimata
Mikhail Ju. Moshkov
Andrzej Skowron
Zbigniew Suraj

Inhibitory Rules in Data Analysis

A Rough Set Approach

Springer

THANK YOU!