CONCURRENT VERSUS SEQUENTIAL THE ROUGH SETS PERSPECTIVE

Zdzislaw Pawlak

Abstract

There are many models of concurrency. An elegant and successful one has been proposed by Petri (cf. Petri (1962)). The paper is an attempt to present a new approach to concurrency based on the rough sets philosophy.

1. Introduction

Suppose a finite set $A = \{a_1, a_2, \dots, a_n\}$ of elements called *agents* is given. With every agent $a \in A$ a finite set of its internal *states* V_a is associated. Each agent can be viewed as a kind of finite state machine (automaton, device etc.). In this note we will consider the following two seemingly similar problems.

1. Analysis. Suppose that agents of A are changing their states according to some rules. The changes are watched by an observer who does not know the rules. The results of the observation can be presented in a form of a table as shown in the example below.

U	a	b	c	d	e
1	1	0	2	1	0
2	0	0	1	2	1
3	2	0	2	1	0
4	0	0	2	2	2
5	1	2	2	1	0

Table 1

In the table the set of agents is $A = \{a,b,c,d,e\}$. Each row in the table contains record of observed states of the set of agents A, and each record is labelled by an element from the set U, in this case by numbers 1,2,3,4 and 5. For example record 3 reveals that during this observation agents a,b,c,d and e were in states 2,0,2,1 and 0, respectively.

The task of the observer is to find out, on the basis of his observations, the rules governing the behavior of the system. More specifically, his task is to find out whether the agents are changing their states independently or the changes are interrelated functionally. In what follows we will identify independence of events with concurrency. On the contrary if such a dependency dose exist we assume that the functional relationship between the agents states is due to the cause-effect principle (however in general this may be not necessarily the case) and therefore they must change their states *sequentially*.

Note also that discovering relations between observed data is the main objective of *machine discovery* (cf. Zytkow (1991)).

Two approaches here are possible, called *the Closed World Assumption* (*CWA*) and *the Open World Assumption* (*OWA*). In the first case we assume that the table contains all possible states combinations, the remaining ones being prevented by the intrinsic nature of the system. In other words the table contains the whole knowledge about the observed behavior of the system – whereas in the second case only a part of possible observations is contained in the table, i.e. it contains partial knowledge about the system behavior only. For the sake of simplicity we will consider in this note only the first case.

2. Synthesis. Tables as shown before can be also treated as a *specification* of the system required behavior. In this case the problem is whether such specification defines *concurrent* or *sequential* system and what are the rules describing the system behavior.

The above both mentioned problems can be solved by employing the concept of an *information system* and the *rough set* as formulated in Pawlak (1991).

Before we enter more specific consideration, first we give some basic definition and properties which will be needed in what follows.

2. Information Systems

Informally an information system is a table rows of which are labeled by *objects*, columns – by *attributes* and entries of the table are *values of attributes*. Formal definition goes as follows.

Information System is a pair $S = (U,A)$, where

U – is a nonempty, finite set called the *universe*,
A – is a nonempty, finite set of *attributes*.

Every attribute $a \in A$ is a total function $a: U \longrightarrow V_a$, where V_a – is the set of *values* of a, called the *domain* of a;

If $S = (U,A)$ and $X \subseteq U$, $B \subseteq A$, than $S = (X,A)$ and $S = (U,B)$ will be refereed to as *X–subsystem or B–subsystem respectively*.

We will identify information system $S = (U,A)$ with a *Finite States System* (*FSM*), elements of U are interpreted as *states* of the system, attributes are meant to denote individual components of the system (finite states machines, called *agents*) and values of attributes are understood as agents *states*.

It is obvious that every subset of attributes defines partition of elements of U, which is defined as follows.

Suppose we are give a system $S = (U,A)$. Every subset $B \subseteq A$, defines a binary relation $IND(B)$, called an *indiscernibility relation* and defined thus:

$$IND(B) = \{(x,y) \in U^2 : \text{for every } a \subset B,\ a(x) = a(y)\}.$$

Obviously $IND(B)$ is an equivalence relation and

$$IND(B) = \bigcap_{a \in B} IND(a)$$

By $U/IND(B)$ (in short U/B) we will denote the family of all equivalence classes of the relation $IND(B)$, i.e. partition generated by the set B, and $[x]_B$ denotes the equivalence class of $U/IND(B)$ containing element $x \in U$.

Partitions generated by attributes are basic tools used to define further notions needed in the presented approach.

3. Reduction of Attributes

As mentioned in the Introduction we identify concurrency with the independence of actions of agents. Hence we need a formal definition of independence of attributes. Minimal subset of $B \subseteq A$ which preserves classification generated by B will be called a reduct of B. It turns out that any reduct of B is the maximal set of agents which can act independently (concurrently). Next the necessary formal definitions are given.

We will say that an attribute $a \in B$ is *superfluous* in B, if $IND(B) = IND(B - \{a\})$; otherwise the attribute a is *indispensable* in B.

If all attributes $a \in B$ are indispensable in B, then B will be called *independent*.

Subset $B' \subseteq B$ is a *reduct* of B, iff B' is *independent* and $IND(B) = IND(B')$

Obviously any subset of an independent set of attributes is also independent.

If $S = (U, A)$ is a system and $B \subseteq A$ is a reduct of A then $S = (U, A)$ will be called *partially concurrent*.

The set of all indispensable attributes in B will be called the *core* of B, and will be denoted by $CORE(B)$.

The following theorem establishes important relationship between the core and the reducts.

Proposition 1

$$CORE(B) = \bigcap_{R \in RED(B)} R$$

where $RED(B)$ the family of all reducts of B. ■

To compute reducts and the core we will use the method proposed by Skowron (cf. Skowron et all. (1991)), which is defined below.

Let $S = (U, A)$ be given, where $U = \{x_1, x_2, \ldots, x_n\}$, and let $B \subseteq A$. By a *discernibility matrix of B* in S, denoted $M_S(B)$, or $M(B)$ if S is understood - we will mean n x n matrix defined thus:

$$c_{ij} = \{a \in B: a(x_i) \neq a(x_j)\} \text{ for } i,j = 1,2, \dots , n.$$

Intuitively entry c_{ij} is the set of all attributes which discern objects x_i and x_j.

The discernibility matrix $M(B)$ assigns to each pair of objects x and y a subset of attributes $\delta(x,y) \subseteq B$, which satisfies the following conditions.

i) $\delta(x,x) = \emptyset$

ii) $\delta(x,y) = \delta(y,x)$

iii) $\delta(x,z) \subseteq \delta(x,y) \cup \delta(y,z)$

It is easily seen that the core is the set of all single element entries of the discernibility matrix $M(B)$, i.e.

$$CORE(B) = \{a \in B: c_{ij} = (a), \text{ for some } i,j\},$$

whereas $B' \subseteq B$ is a reduct of B, if B' is the minimal (with respect to inclusion) subset of B such that

$$B' \cap c \neq \emptyset \text{ for any nonempty entry } c \ (c \neq \emptyset) \text{ in } M(B).$$

In other words reduct is a minimal subset of attributes which discerns all objects discernible by the whole set of attributes.

With every discernibility matrix $M(B)$ we can associate uniquely a *discernibility (boolean) function* $f(B)$, defined as shown below.

Let us assign to each attribute $a \in B$ a binary boolean variable $\bar{a}$, and let $\sum \delta(x,y)$ denote the boolean sum of all boolean variables assigned to the set of attributes $\delta(x,y)$, provided $\delta(x,y) \neq \emptyset$. The discernibility function can be defined now as

$$f(B) = \Pi \{\sum \delta(x,y) : (x,y) \in U^2 \text{ and } \delta(x,y) \neq \emptyset\}$$

The following Proposition gives an important property which enables us to compute easily all reducts of B.

Proposition 2 (Skowron et all. (1991))

All constituents in the minimal disjunctive normal form of function $f(B)$ are all reducts of B. ■

Thus in order to compute the "concurrent part" of the system we have to compute first discernibily matrix for the required subset of attributes, next discernibility function must be computed and finally the normal form of the function gives us all reducts. The example below depicts the procedure more exactly.

Example 1

Consider system as represented in Table 1. For this system we have the following discernibility matrix.

	1	2	3	4	5
1					
2	a,c,d,e				
3	a	a,c,d,e			
4	a,d,e	c,e	a,d,e		
5	b	a,b,c,d,e	a,b	a,b,d,e	

Table 2

After simplification (using the absorption law) we get the following discernibility function and its minimal disjunctive normal form

$$ab(c+e) = abc+abe.$$

Thus the core of the set $A = \{a,b,c,d,e\}$ is the set $\{a,b\}$ and there are two following reducts $\{a,b,c\}$ and $\{a,b,e\}$ of the set A .

This means that if the observed behavior of the system, is as shown in Table 1, we can not uniquely determine which part of the system is sequential.

4. Dependency of Attributes

Having defined the concurrent part of the system next we would like to recognize which subsystems are sequential. To this end we need the notion of dependency of attributes.

Intuitively speaking set of attributes $B \subseteq A$ depends on set of attributes $C \subseteq A$ $(C \Rightarrow B)$, if values of attributes in B are uniquely determined by values of attributes in C, i.e. if there exists a function which assigns to each set of values of C set values of B. Formally

$$C \Rightarrow B \text{ iff } IND(C) \subseteq IND(B).$$

If $C \Rightarrow B$ and $B \Rightarrow C$ we say that C and B are *equivalent*.

If on the right hand side of the dependency there is only one attribute we will call this kind of dependency *elementary*.

The next propositions give an important relationship between the notions of a reduct and the dependency.

Proposition 3

Let $S = (U,A)$ be given and let $B \subseteq A$. If B' is a reduct of B and $B-B' \neq \emptyset$, then $B' \Rightarrow B-B'$. ∎

The next two propositions are a direct consequence of the definition of dependency.

Proposition 4

$B \Rightarrow C$, implies $B \Rightarrow C'$, for every $\emptyset \neq C' \subseteq C$. ∎

In particular $B \Rightarrow C$, implies $B \Rightarrow \{a\}$, for every $a \in C$.

Proposition 5

If B' is a reduct of B, then neither $\{a\} \Rightarrow \{b\}$ nor $\{b\} \Rightarrow \{a\}$ holds, for any $a, b \in B', a \neq b$, i.e. all attributes in the reduct are pairwise independent. ∎

Theorems 3 and 4 enables us to find all dependencies among attributes and the example which is given next will serve as an illustration of the just defined ideas.

Example 2.

By Proposition 3 we get for system presented in Table 1 the following dependencies

$$\{a,b,c\} \Rightarrow \{d,e\} \text{ and } \{a,b,e\} \Rightarrow \{c,d\},$$

and consequently by Proposition 4 we have the elementary dependencies as below

$$\{a,b,c\} \Rightarrow \{d\}$$

$$\{a,b,c\} \Rightarrow \{e\}$$

and

$$\{a,b,e\} \Rightarrow \{c\}$$

$$\{a,b,e\} \Rightarrow \{d\}.$$

The intuitive meaning of the obtained results is that from the observation we can infer that agent d is dependent no matter which reduct is chosen , whereas agents c and e are dependent accordingly to the chosen reduct. ∎

Now we are ready to define the notion of totally concurrent system. Before, we need the definition of partial dependency of attributes. Let $B, C \subseteq A$ and k ($0 \leq k \leq 1$) be given. We say that C *depends on* B in the *degree* k ($B \Rightarrow_k C$) if

$$k = \frac{|POS_B(C)|}{|\ U\ |}$$

Note that in the case of $B = C$, i.e. $k = 1$, we get the previous definition, and we will say in this case that C depends *totally* on B; if $0 < k < 1$ – we will say that C *partially* depends on B and if $k = 0$ we will say that C is *totally* independet on B.

The definition which follows is a slight modification of a definition proposed by A. Skowron.

System $S = (U,A)$ is *totally concurrent* iff

$$|A| = 1 \text{ or } A - \{a\} \Rightarrow_0 \{a\} \text{ for every } a \in A.$$

For example in the system shown in Table 1 there are two partially concurrent subsystems $S' = (U, \{a,b,c\})$ and $S'' = (U,\{a,b,e\})$ but the system does not contain any totally concurrent subsystem. This means that on the basis of the observed behavior of the system we can only say that there are two possible candidates for partial concurrent subsystems S' and S'', however we are unable to conclude positively, using the available information, what is the real one. In the case of synthesis of the specified by Table 1 system, the obtained result means that we have two options in the design of the system, i.e. we can choose either S' or S'' as a concurrent (partially) subsystem, and the remaining agents must work sequentially.

5. Reduction of Dependencies

Suppose we are given a dependency $B \Rightarrow C$. It may happen that the set C depends not on the whole set B but on its subset B' and therefore we might be interested to find out this subset. In order to solve this problem we need the notion of a relative reduct, which will be defined and discussed next.

Let $B,C \subseteq A$, and let

$$POS_B(C) = \bigcup_{X \in U/IND(C)} \underline{B}X$$

where $\underline{B}X = \bigcup\{Y \in U/IND(B)\colon Y \subseteq X\}$ is so called the *lower approximation* of X by B (the *B-lower approximation* of X).

We will say that attribute $a \in B$ is *C-superfluous* in B, if $POS_B(C) = POS_{(B-\{a\})}(C)$; otherwise the attribute a is *C-indispensable* in B.

If all attributes $a \in B$ are C-indispensable in B, then B will be called *C-independent*.

Subset $B' \subseteq B$ is a *C-reduct* of B, iff B' is C-independent and $POS_B(C) = POS_{B'}(C)$.

The set of all C-indispensable attributes in B will be called the *C-core* of B, and will be denoted by $CORE_C(B)$. The counterpart of Proposition 1 has now the form.

Proposition 1'

$$CORE_C(B) = \bigcap_{R \in RED_C(B)} R$$

where $RED_C(B)$ is the family of all C-reducts of B. ∎

If $R = C$ we will get the previous definitions.

Relative reducts can be computed similarly as before, we have only to modify slightly the discernibilitry matrix in this case.

Let $S = (U,A)$ given with $U = \{x_1, x_2, \ldots, x_n\}$, and let $B, C \subseteq A$. By an *C-discernibility matrix of B* in S, denoted $M_C(B)$, we mean n x n matrix defined thus:

$$c_{ij} = \{a \in B: a(x_i) \neq a(x_j) \text{ and } w(x_i, x_j)\}$$

where $w(x_i,x_j) \equiv x_i \in POS_B(C)$ *and* $x_j \notin POS_B(C)$ *or*
$x_i \notin POS_B(C)$ *and* $x_j \in POS_B(C)$ *or*
$x_i, x_j \in POS_B(C)$ *and* $(x_i, x_j) \notin IND(C)$

for $i,j = 1,2, \ldots, n$ (cf. Skowron et all. (1991)).
If the partition defined by C is definable by B then the condition $w(x_i,x_j)$ in the above definition can be reduced to $(x_i,x_j) \notin IND(C)$.

Thus entry c_{ij} is the set of all attributes which discern objects x_i and x_j that do not belong to the same equivalence class of the relation $IND(C)$.

The remaining definitions need also slight modifications.

The C-core is the set of all single element entries of the discernibility matrix $M_C(B)$, i.e.

$$CORE_C(B) = \{a \in B: (a) \text{ is an element of } M_C(B)\}$$

Set $B' \subseteq B$ is the C-reduct of B, if B' is the minimal (with respect to inclusion) subset of B such that

$$B' \cap c \neq \emptyset \text{ for any nonempty entry } c\ (c \neq \emptyset) \text{ in } M_C(B).$$

Thus C-reduct is the minimal subset of attributes that discerns all equivalence classes of the relation $IND(C)$ discernible by the whole set of attributes.

Every discernibility matrix $M_C(B)$ defines uniquely a *discernibility* *(boolean) function* $f_C(B)$, defined as before

$$f_C(B) = \Pi \{\sum \delta(x,y) : (x,y) \in U^2 \text{ and } \delta(x,y) \neq \emptyset\}$$

Proposition 2 has now the form

Proposition 2'

All constituents in the minimal disjunctive normal form of the function $f_C(B)$ are all C-reducts of B. ∎

In the example which follows we will illustrate the idea more closely.

Example 3.

Let us compute the relative reducts for all elementary dependencies valid in the system

$\{a,b,c\} \Rightarrow \{d\}$ $\{a,b,c\} \Rightarrow \{e\}$

and

$\{a,b,e\} \Rightarrow \{c\}$

$\{a,b,e\} \Rightarrow \{d\}$.

We are going to compute relative reducts of the left hand sides of the above dependencies.

In order to compute d-reduct of $\{a,b,c\}$ first we have to define the corresponding discernibility matrix, which is given below

	1	2	3	4	5
1					
2	a,c				
3	–	a,c			
4	a	–	a		
5	–	a,b,c	–	a,b	

Table 3

The disernibility function for this table is a, hence the dependency $\{a,b,c\} \Rightarrow \{d\}$ can be simplified as $\{a\} \Rightarrow \{d\}$.

For the dependency $\{a,b,c\} \Rightarrow \{e\}$ e-reduct of $\{a,b,c\}$ can be computed from the following discernibility matrix

	1	2	3	4	5
1					
2	a,c				
3	–	a,c			
4	a	c	a		
5	–	a,b,c	–	a,b	

Table 4

This table yields the discernibility function ac and consequently the dependency $\{a,b,c\} \Rightarrow \{e\}$ can be simplified as $\{a,c\} \Rightarrow \{e\}$.

Proceeding in a similar way for the second set of dependencies $\{a,b,e\} \Rightarrow \{c\}$ and $\{a,b,e\} \Rightarrow \{d\}$, we get the following results. For the dependency $\{a,b,e\} \Rightarrow \{c\}$ the discernibility matrix is

	1	2	3	4	5
1					
2	a,e				
3	–	a,e			
4	–	e	–		
5	–	a,b,e	–	a,b,e	

Table 5

which reduces the dependency $\{a,b,e\} \Rightarrow \{c\}$ to $\{e\} \Rightarrow \{c\}$.

For the last dependency $\{a,b,e\} \Rightarrow \{d\}$ we have the discernibility matrix

	1	2	3	4	5
1					
2	a,e				
3	–	a,e			
4	a,e	–	a,e		
5	–	a,b,e	–	a,b,e	

Table 6

which yields the that the dependency $\{a,b,e\} \Rightarrow \{d\}$ can be reduced either to $\{a\} \Rightarrow \{d\}$ or $\{e\} \Rightarrow \{d\}$.

Intuitive interpretation of the obtained results is left for the interested reader. ■

6. Reduction of States

Suppose we are given a dependency $B \Rightarrow C$ where B is a relative C-reduct of B. To further investigation of the dependency we might be interested to know exactly how values of attributes from C depends on values of attributes from B. To this end we need a procedure eliminating values of attributes from B which does not influence on values of attributes from C. It turns out that this can be achieved by very similar thinking as in the case of elimination of superfluous attributes. In this section we will discuss this problem more formally.

Suppose we are given $B,C \subseteq A$, and $x \in U$. We say that value of attribute $a \in B$, is *C-superfluous* for x, if

$$[x]_{IND(B)} \subseteq [x]_{IND(C)} \text{ implies } [x]_{IND(B-\{a\})} \subseteq [x]_{IND(C)} ;$$

otherwise the value of attribute a is *C-indispensable* for x.

If for every attribute $a \in B$ value of a is C-indispensable for x, then B will be called *C-independent* for x.

Subset $B' \subseteq B$ is a *C-reduct* of B for x, iff B' is C-independent for x and

$$[x]_{IND(B)} \subseteq [x]_{IND(C)} \text{ implies } [x]_{IND(B')} \subseteq [x]_{IND(C)}.$$

The set of all C-indispensable for x values of attributes in B will be called the *C-core* of B for x, and will be denoted $CORE^x_C(B)$.

The counterpart of Proposition 1 now has the form

Proposition 1''

$$CORE^x_C(B) = \bigcap_{R \in RED^x_C(B)} R$$

where $RED^x_C(B)$ is the family of all C-reducts of B for x. ■

For computing reducts and the core in this case we use as a starting point the discernibility matrix $M_C(B)$ and the discernibility function, defined as below:

$$f^x_C(B) = \Pi \{\textstyle\sum \delta(x,y):\ y \in U \text{ and } \delta(x,y) \neq \emptyset\}$$

Example 4

In the considered example there is only one interesting elementary dependency $\{a,c\} \Rightarrow \{e\}$, which will be used to illustrate the ideas considered in this section. To this end we will need the discernibility matrix given in Table 4 from which we get the following discernibility functions

$$f_e^1(\{a,c\}) = a$$

$$f_e^2(\{a,c\}) = c$$

$$f_e^3(\{a,c\}) = a$$

$$f_e^4(\{a,c\}) = ac$$

$$f_e^5(\{a,c\}) = a.$$

The obtained result means that the dependency $\{a,c\} \Rightarrow \{e\}$ can be presented in a form shown in the table below

U	a	c	e
1	1	x	0
2	x	1	1
3	2	x	0
4	0	2	2
5	1	x	0

Table 7

where crosses "x" denote "don't care" values of attributes, i.e. states which do not contribute to the dependency and as such can be eliminated.

Dependencies can be also presented in a form of decision rules. For example the considered dependency $\{a,c\} \Rightarrow \{e\}$ can be presented as the set of the following decision rules:

$$a_1 \rightarrow e_0$$

$$a_2 \rightarrow e_0$$

$$b_1 \rightarrow e_1$$

$$a_0 b_2 \rightarrow e_2$$

or in shorter version

$$a_1 + a_2 \rightarrow e_0$$

$b_1 \rightarrow e_1$

$a_0 b_2 \rightarrow e_2$.

The decision rules can be viewed as a formal description of function defined by the dependency $\{a,c\} \Rightarrow \{e\}$. Similarly decision rules can be obtained for the remaining elementary dependencies. ■

4. Conclusions

The application of the rough set philosophy enable us to detach from the observation of a finite state system its concurrent and sequential subsystems, when the analysis from observation is of primary concern. In the case when a system should be designed according to a preassumed specification the obtained results enable us to find out parts of the system which can be performed concurrently and those which must act sequentially.

5. Acknowledgements

Thanks are due to Professor Andrzej Skowron for stimulating discussions and remarks.

References

Pawlak, Z. (1991). Rough Sets - Theoretical Aspects of Reasoning about Data. *KLUWER ACADEMIC PUBLISHERS.*

Petri, C.A. (1962). Fundamentals of a Theory of Asynchronons Information Flow. *Information Processing 1962, Proc. of the IFIP Congress 1962, Munchen, Amsterdam, North-Holland, 386-390.*

Skowron, A. and Rauszer, C. (1991). The Discernibility Matrices and Functions in Information Systems. *Institute of Computer Science Reports,* 1/91, *Warsaw University of Technology, and Fundamenta Informaticae (to appear)*

Zytkow, J. (1991). Interactive Mining of Regularities in Databases, In: *Piatetsky-Shapiro G., Frawley W. (Eds.), Knowledge Discovery in Databases, The AAAI Press, Menlo Park, CA.*