Discovery of Concurrent Data Models from Experimental Tables: A Rough Set Approach

Andrzej Skowron1* and Zbigniew Suraj2*

1Institute of Mathematics, Warsaw University
Bansacha 2, 02-097 Warsaw, Poland
2Institute of Mathematics, Pedagogical University
Rejtana 16A, 35-310 Rzeszow, Poland
email: skowron, suraj@mimuw.edu.pl

Abstract

The main objective of machine discovery is the determination of relations between data and of data models. In the paper we describe a method for discovery of data models represented by concurrent systems from experimental tables. The basic step consists in a determination of roles which yield a decomposition of experimental data tables; the components are then used to define fragments of the global system corresponding to a table. The method has been applied for automatic data models discovery from experimental tables with Petri nets as models for concurrency.

Key words: data mining, system decomposition, rough sets, concurrent models

Introduction

The aim of this paper is to present an approach to the decomposition of information systems. Our approach can be applied to the discovery of data models in the form of concurrent systems.

Decomposition of large experimental data tables can be treated as one of the fundamental tools in data mining. It is usually imposed by the high computational complexity of the search for relations between data on one hand and/or the structure of the process of data models discovery on the other (see e.g. (Zytkowski 1991), (Piatetsky-Shapiro & Frawley 1991), (Shrager & Langley 1990), (Kodratoff & Michalski 1990)).

Our approach is based on rough set theory (Pawlak 1991) and boolean reasoning (Brown 1990). It consists of three levels. First we show how experimental data tables are represented by information systems (Pawlak 1991). Next we discuss how any information system S can be decomposed (with respect to any of its reduct) into components linked by some connections which allow to preserve some constraints. Any component represents in a sense the strongest functional module of the system. The connections between components represent constraints which must be satisfied when these functional modules coexist in the system. The components together with the connections define a so called covering of S. Finally, we use the coverings of the information system S to construct its concurrent model in the form of a marked Petri net (N, M) (Skowron & Suraj 1994) with the following property: the reachability set R(N, M) is in one-to-one correspondence with the set of all global states consistent with all decision rules valid in S (and having examples in S).

In the paper we investigate the decomposition problems which can now be roughly defined as follows:

Component Extracting Problem:
Input: An information system S.
Output: All components of S.

Covering Problem:
Input: An information system S.
Output: The set of all coverings of S.

Our approach can be applied for automatic feature extraction and for control design of systems represented by experimental tables.

Information Systems

In the paper we represent experimental data by information systems (Pawlak 1991).

An information system is a pair S = (U, A), where U is a nonempty finite set of objects, called the universe, A is a nonempty finite set of attributes, i.e. a : U → V_a for a ∈ A, where V_a is called the value set of a.

Our approach is based on rough set theory (Pawlak 1991) and boolean reasoning (Brown 1990). It consists of three levels. First we show how experimental data tables are represented by information systems (Pawlak 1991). Next we discuss how any information system S can be decomposed (with respect to any of its reduct) into components linked by some connections which allow to preserve some constraints. Any component represents in a sense the strongest functional module of the system. The connections between components represent constraints which must be satisfied when these

Note: The text contains a table which is not transcribed here due to its layout and formatting.
Table 1. An example of an information system

<table>
<thead>
<tr>
<th>U \ A</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>u_2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>u_3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>u_4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>u_5</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Let $S = (U, A)$ be an information system. With any subset of attributes $B \subseteq A$ we associate a binary relation $\text{ind}(B)$, called indiscernibility relation, which is defined by: $\text{ind}(B) = \{(u, u') \in U \times U : \text{ for every } a \in B, a(u) = a(u')\}$.

Any information system $S = (U, A)$ determines an information function, $\text{Inf}_A : U \rightarrow P(A \times V)$ defined by $\text{Inf}_A(u) = \{(a, u(a)) : a \in A\}$, where $V = \bigcup_{a \in A} V_a$ and $P(X)$ denotes the power set of $X$.

The set $\{\text{Inf}_A(u) : u \in U\}$ will be denoted by $\text{INF}(S)$.

Hence, $u \text{ ind}(A) u'$ iff $\text{Inf}_A(u) = \text{Inf}_A(u')$.

Decision Rules

Decision rules express some of the relationships between attributes of information systems.

Let $S = (U, A)$ be an information system and let $B, C \subseteq A$. We say that the set $C$ depends on $B$ in $S$ in degree $I_C(0 \leq I_C \leq 1)$, symbolically $B \Rightarrow C$, if $k = \|\text{POS}_B(C)\|$ in $S$ is the $B$-positive region of $C$ in $S$ (Pawlak 1991).

If $k = 1$ we write $B \geq C$ instead of $B \Rightarrow C$. In this case $B \geq C$ means that $\text{ind}(B) \subseteq \text{ind}(C)$. If the right hand side of a dependency consists of one attribute only, we say the dependency is elementary.

A decision rule over $A$ and $V$ is any expression of the following form: (1) $a_1 = v_1 \wedge \ldots \wedge a_r = v_r$. A rule of type (1) is called trivial if $a_p = v_p$ also appears on the left hand side of the rule. It is true in $S$ if $\| a_1 = v_1 \wedge \ldots \wedge a_r = v_r \|, a_p = v_p \|$, which is denoted by (2) $a_1 = v_1 \wedge \ldots \wedge a_r = v_r \geq a_p = v_p$. A rule of type (2) has an example in $S$ if $\| a_i = v_i \wedge \ldots \wedge a_r = v_r \|, a_p = v_p \|$. By $D(S)$ we denote the set of all true decision rules which have examples in $S$.

Let $R \subseteq D(S)$. An information vector $v = (v_1, \ldots, v_m)$ is consistent with $R$ iff for any decision rule $a_1 = v_1 \wedge \ldots \wedge a_r = v_r \Rightarrow a_p = v_p$ in $R$ if $a_p = v_p$ for $j = 1, \ldots, r$ then $v_p = v_p$. The set of all information vectors consistent with $R$ is denoted by $\text{CON}(R)$.

Let $S' = (U', A')$ be a $U'$-extension of $S = (U, A)$. We say that $S'$ is a consistent extension of $S$ iff $D(S) \subseteq D(S')$. $S'$ is a maximal consistent extension of $S$ iff $S'$ is a consistent extension of $S$ and any consistent extension $S''$ of $S$ is a restriction of $S'$.

Reduction of Attributes

Let $S = (U, A)$ be an information system. A subset $B \subseteq A$ is a reduct of $S$ (Pawlak 1991) iff $\text{ind}(B) = \text{ind}(A)$ and $\text{ind}(B') \neq \text{ind}(A)$ for any $B' \subseteq B$. The set of all reducts in $S$ is denoted by $\text{RED}(S)$.

Let $S = (U, A)$ be an information system and let us assume that $U = \{u_1, \ldots, u_n\}$, and $A = \{a_1, \ldots, a_m\}$. By $M(S)$ we denote an $n \times n$ matrix $(c_{ij})$, called the discernibility matrix of $S$, such that $c_{ij} = \{a_i \in A : a_i = u_i\}$ for $i, j = 1, \ldots, n$ (Skowron & Rauszer 1992).

With any discernibility matrix $M(S)$ we can associate a discernibility function $f_M(S)$. The discernibility function $f_M(S)$ for an information system $S$ is a boolean function of $m$ propositional variables $a_1, \ldots, a_m$ (where $a_i \in A$ for $i = 1, \ldots, m$) defined as the conjunction of all expressions $\bigvee c_{ij}$, where $\bigvee c_{ij}$ is the disjunction of all the elements of $c_{ij} = \{a : a \in c_{ij}\}$, where $1 \leq j < i \leq n$ and $c_{ij} \neq \emptyset$. In the sequel we write $a$ instead of $a$ when no confusion can arise.

**PROPOSITION 1** (Skowron & Rauszer 1992). Let $S = (U, A)$ be an information system and let $f_M(S)$ be a discernibility function for $S$. Then all prime implicants (Wegner 1987) of the function $f_M(S)$ correspond to all reducts of $S$.

In order to construct a concurrent model for a given information system all reducts of the system have to be computed first (Skowron & Rauszer 1992).

**PROCEDURE for computing $\text{RED}(S)$:**

Step 1. Compute the discernibility matrix for the system $S$.

Step 2. Compute the discernibility function $f_M(S)$ associated with the discernibility matrix $M(S)$.

Step 3. Compute the minimal disjunctive normal form of the discernibility function $f_M(S)$. (The normal form of the function yields all reducts).

**EXAMPLE 2.** Applying the above procedure to the system in Example 1, we obtain two reducts: $R_1 = \{a, b, c\}$ and $R_2 = \{a, b, c\}$ of the system.
PROPOSITION 2 (Pawlak 1992). Let $S = (U, A)$ be an information system, $R \in \text{RED}(S)$, and $R \subset A$. Let $f_{M(S')}$ be a relative discernibility function for the system $S' = (U, R \cup \{a^*\})$, where $a^* \in A - R$. Then all prime implicants of the function $f_{M(S')}$ correspond to all $\{a^*\}$-reducts of $S'$.

The next example illustrates how to find all dependencies among attributes.

EXAMPLE 3. Let us consider again the system $S$ from Example 1. We have $\{a, b, c\} \not\subseteq \{d, e\}$ and $\{a, b, e\} \not\subseteq \{d\}$, so $\{a, b, c\} \not\subseteq \{d\}$. The set of rules corresponding to all nontrivial dependencies within the reduct $R_2$ has the form: $a_1 \lor a_2 \lor b_1 \lor e_0 \lor a_0 \lor a_2 \lor e_1 \lor e_2 \lor b_0 \lor e_1 \lor e_2 \lor a_0 \lor a_1 \lor b_1 \lor e_1 \lor e_2 \lor d_1$, while the set of decision rules corresponding to all nontrivial dependencies of $R_2$ with $c, d$ has the following form: $a_1 \lor a_2 \lor b_1 \lor e_0 \lor a_0 \lor a_2 \lor b_1 \lor e_0 \lor d_1$, $a_0 \lor e_1 \lor e_2 \lor d_2$. These rules can be generated by applying the method presented in (Skowron 1993).

**Decomposition of Information Systems**

This section introduces concepts and notation related to the decomposition of information systems as well as a method for constructing components and coverings of a given information system with respect to its reducts.

Let $S = (U, A)$ be an information system. An information system $S$ is said to be covered with constraints $C$ (or $C$-covered, in short) by information systems $S_1 = (U_1, A_1), \ldots, S_k = (U_k, A_k)$ if $\text{INF}(S') = \bigcup_{i=1}^{k} \text{INF}_{A_i}(u_i)$ and $\bigcup_{i=1}^{k} \text{INF}_{A_i}(u_i) \subseteq \text{INF}(S)$ and $\bigcup_{i=1}^{k} u_i \subseteq U$ for $i = 1, \ldots, k$, where $S'$ is a maximal consistent extension of $S$ and $C$ is a set of decision rules. The pairs $\{S_1, \ldots, S_k\}$, $C$ is called a $C$-covering of $S$ (or a covering of $S$, in short). The sets $S_1, \ldots, S_k$ are its components and $C$ is the set of constraints (connections).

EXAMPLE 4. Let us consider the information system $S$ from Example 1. It is easy to see that the information systems $S_1 = (U_1, A_1)$, $S_2 = (U_2, A_2)$ and $S_3 = (U_3, A_3)$ represented by Tables 2, 3 and 4, respectively, and the set of constraints $C$ containing decision rules $a_1 \lor a_2 \lor e_0$, $e_1 \lor e_2 \lor a_0$, $a_0 \lor e_2 \lor b_0$, $b_1 \lor a_1$, $b_1 \lor e_0$, and $e_1 \lor e_2 \lor b_0$ yield a $C$-covering of $S$.

<table>
<thead>
<tr>
<th>$U_1$</th>
<th>$A_1$</th>
<th>a</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. The information system $S_1$

<table>
<thead>
<tr>
<th>$U_2$</th>
<th>$A_2$</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$u_4$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. The information system $S_2$

PROPOSITION 3. Every information system has at least one covering.

If $S = (U, A)$, then the system $S' = (U', A')$ such that $U' \subseteq U$, $A' = \{a' : a \in B \subseteq A\}$, $a'(u) = a(u)$ for $u \in U'$ and $V_{a'} = V_a$ for $a \in A$ is said to be a subsystem of $S$.

EXAMPLE 5. Every information system in Example 4 is a subsystem of the system $S$ from Example 1.

Let $S = (U, A)$ be an information system and let $R \in \text{RED}(S)$. An information system $S' = (U', A')$ is a normal component of $S$ (with respect to $R$) iff the following conditions are satisfied: (i) $S'$ is a subsystem of $S$, (ii) $A' = B \cup C$, where $B$ is a minimal (with respect to $\subseteq$) subset of $R$ such that $B \not\subseteq \{a\}$ for some $a \in A - R$ and $C$ is the set of all attributes $a$ with the above property.

The set of all normal components of $S$ (with respect to $R$) is denoted by $\text{COMPR}_R(S)$.

EXAMPLE 6. The subsystems $S_1, S_2, S_3$ are normal components of $S$ from Example 1 (with respect to the reduct $R_2$). A more detailed explanation of this fact is included in Example 7.

PROPOSITION 4. Every information system has at least one normal component (with respect to any of its reduct).

Let $S' \in \text{COMPR}_R(S)$ and $S'' = (U', B'' \cup C'')$. By $X_R$ we denote the set of all attributes which simultaneously occur in normal components of $S$ (with respect to $R$) and in the reduct $R$, i.e. $X_R = \bigcup_{S' \in \text{COMPR}_R(S)} B''$. Let $X_R$ be a set defined for $S$ and $R$ as above. We say that a subsystem $S'' = (U'', A'')$ of $S$ is a degenerated component of $S$ (with respect to $R$) iff $a'' = \{a\}$ for some $a \in A - X_R$. We denote this fact by $\{a\} \not\subseteq \emptyset$ (the empty set).

In the sequel a component (with respect to a reduct) will be assumed to be either a normal component or a degenerated component (with respect to the reduct).

PROPOSITION 5. Let $S = (U, A)$ be an information system and let $R$ be its reduct. Then the information system $S$ consists of $|R - X_R|$ degenerated components (with respect to $R$).

Let $S = (U, A)$ be an information system, $R \in \text{RED}(S)$. We say that $S$ is $R$-decomposable into components or that $S$ is $C$-coverable by components (with respect to $R$) iff there exist components $S_1 = (U_1, B_1 \cup C_1), \ldots, S_k = (U_k, B_k \cup C_k)$ of $S$ (with respect to $R$) with a set of constraints $C$ such that $B_1 \cup \ldots \cup B_k = R$ and $C_1 \cup \ldots \cup C_k = A - R$, yielding a $C$-covering of $S$.  

<table>
<thead>
<tr>
<th>$U_3$</th>
<th>$A_3$</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$u_3$</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. The information system $S_3$
The set of constraints (connections) \( C \) includes: (i) decision rules corresponding to nontrivial dependencies between attributes in \( B_i \) \((i=1, ..., k)\) called internal linkings (the internal connections) within the component \( S_i \) of \( S \), (ii) decision rules corresponding to nontrivial dependencies between attributes in \( B_i \) \((i=1, ..., k)\) and those in the set \( A - A_i \), where \( A_i = B_i \cup C_i \) called external linkings (the external connections) with the outside of \( S_i \).

**THEOREM 1**. Every information system is \( C \)-coverable by components (with respect to any its reduct), where \( C \) is the set of all internal and external linkings of \( S \).

We obtained a constructive method of information systems (data tables) decomposition into functional modules interconnected by external linkings. One can observe a similarity of our data models to those used in general system theory and control design.

**PROPOSITION 6**. Let \( R \) be a reduct of an information system \( S \). Then \( S \) has at least one \( C \)-covering by components (with respect to \( R \)), where \( C \) is the set of all internal and external linkings of \( S \).

We denote by \( COVERR(S) \) the family of all \( C \)-coverings of \( S \) (with respect to \( R \)), where \( C \) is the set of all internal and external linkings of \( S \).

**Procedures for Computing Components and Coverings**

All normal components of a given information system \( S = (U, A) \) (with respect to a reduct \( R \in RED(S) \)) can be obtained by the following procedures:

**PROCEDURE for computing \( COMPR(S) \):**

Input. An information system \( S = (U, A), \) a reduct \( R \in RED(S) \).

Output: Components of \( S \) (with respect to \( R \)), i.e. the set \( COMPR(S) \).

1. **Step 1.** Compute all dependencies of the form: \( R \nRightarrow \{a\} \), for any \( a \in A - R \).

2. **Step 2.** Compute the discernibility function \( f_M(S) \) for each subsystem \( S' = (U, R \cup \{a\}) \) with \( a \in A - R \).

3. **Step 3.** For all dependencies of the form \( B \Rightarrow \{a_1\}, \ldots, B \Rightarrow \{a_k\} \), where \( B \) is any subset of \( R \) obtained in Step 2, construct a dependency \( B \Rightarrow C \), where \( C = \{a_1\} \cup \ldots \cup \{a_k\} \). The set \( C \) is the maximal subset in \( A - R \) such that the dependency \( B \Rightarrow C \) is true. Now the subsystem \( S'' = (U', B \cup C) \) of \( S \) defines a normal component of \( S \) (with respect to \( R \)).

The complexity of the computation of \( RED(S) \) is high. Nevertheless, the existing procedures and heuristics are sufficient for the computation of reducts in many applications (see e.g. (Bazan, Skowron & Synak 1994)).

**EXAMPLE 7.** Let us perform the procedure for the computation \( COMPR(S) \) for the information system \( S \) of Example 1 and its reduct \( R_2 \).

1. **Step 1.** The following elementary dependencies are valid in the system \( S \) for the reduct \( R : \{a, b, c\} \Rightarrow \{e\}, \{a, b, e\} \Rightarrow \{d\} \).

2. **Step 2.** We compute the minimal subsets of \( R_2 \) on which the sets \( \{e\} \) and \( \{d\} \) depend, i.e. we compute the relative reducts (cf. (Pawlak 1992)) of the left hand sides of the above dependencies. To do the first elementary dependency we consider the information system \( S_1 = (U, B \cup \{e\}) \) with \( B = \{a, b, c\} \). Hence, \( f_M(S_1)(a, b, e) = c \). Thus \( \{a, b, e\} \Rightarrow \{c\} \) can be simplified to \( \{e\} \Rightarrow \{c\} \).

We reduce the second dependency in a similar way.

As a consequence, \( \{a, b, e\} \Rightarrow \{d\} \) can be reduced either to \( \{a\} \Rightarrow \{d\} \) or \( \{e\} \Rightarrow \{d\} \). Eventually, we get the following minimal dependencies (i.e. dependencies with a minimal number of attributes on the left hand side) with respect to \( R_2 \) in the information system \( S : \{a\} \Rightarrow \{d\}, \{e\} \Rightarrow \{c\}, \{e\} \Rightarrow \{d\} \). This completes Step 2 of the above procedure.

3. **Step 3.** For dependencies \( \{e\} \Rightarrow \{c\} \) and \( \{e\} \Rightarrow \{d\} \) we construct a new dependency \( \{e\} \Rightarrow \{c, d\} \). Now we have \( \{a\} \Rightarrow \{d\} \) and \( \{e\} \Rightarrow \{c, d\} \).

The complexity of the computation of \( RED(S) \) is high. Nevertheless, the existing procedures and heuristics are sufficient for the computation of reducts in many applications (see e.g. (Bazan, Skowron & Synak 1994)).

**Example 8.** The information system \( S \) of Example 1 has one covering \((\{S_1, S_2, S_3\}, C)\) (with respect to \( R \)).
to the reduct $R_2$), where $S_1, S_2, S_3$ denote components of $S$ (with respect to $R_2$) computed in Example 7 and $C$ is the set of all internal and external linkings of $S$.

There are no internal linkings in the components of the system (with respect to $R_2$), since each component of $S$ contains only one attribute from the reduct $R_2$. However, the components are connected by external linkings of the form: for $S_1$ and $S_2$: $a_1 \lor a_2 \not\rightarrow e_1, e_1 \lor e_2 \not\rightarrow a_0$; for $S_1$ and $S_3$: $a_0 \lor a_2 \not\rightarrow b_0, b_1 \not\rightarrow a_1, b_1 \not\rightarrow d_1$; for $S_2$ and $S_3$: $e_1 \lor e_2 \not\rightarrow b_0, b_1 \not\rightarrow e_0, b_1 \not\rightarrow c_2, b_1 \not\rightarrow d_1$.

**How to Compute Concurrent Data Models from Information Systems?**

We present a method for constructing a marked Petri net $(N_S, M_S)$ for an arbitrary information system $S$ such that the reachability set $R(N_S, M_S)$ represents the set of all global states consistent with a given information system $S$.

That method consists of two steps. First, decision rules corresponding to two kinds of dependencies are generated. The first kind consists of the partial dependencies between attributes within reducts, the second - of the dependencies between attributes not in reducts and those within reducts. Next, the decision rules so obtained (and represented in an optimal form with respect to the number of descriptors on the left hand side) (Skowron 1993) are implemented by means of a Petri net (Murata 1989).

First initial transformations of decision rules are performed. There are two rules (see Figure 1).

![Figure 1](image)

**Figure 1. Initial transformation rules, where $p, q, r$ are descriptors in $S$**

Next a method for transforming decision rules representing a given information system into a Petri net is executed. This method consists of three levels:

1. A net representing all processes (attributes) in a reduct of an information system is constructed.
2. The net obtained in the first step is extended by adding the elements (arcs and transitions) of the net induced by the decision rules determined by: (i) all nontrivial dependencies between processes of the information system; (ii) partial dependencies between processes within a reduct of the information system.
3. We add to the net obtained so far the subnets corresponding to situations when between some internal states of processes (but not all states) there are no dependencies represented by the information system.

This method is repeated for all reducts of the given information system. Finally, the obtained nets are merged.

Such an approach makes the appropriate construction of a net much more readable. In the example which follows we only illustrate step 2(i) of the above transformation method. This example uses the reduct $R_2$ of the information system $S$ of Example 1.

**EXAMPLE 9.** Consider the decision rules from Example 3 for the system of Example 1: $a_0 \not\rightarrow d_2, a_1 \lor a_2 \not\rightarrow d_1, e_0 \not\rightarrow d_1, e_1 \lor e_2 \not\rightarrow d_2, e_0 \lor e_2 \not\rightarrow c_2, e_1 \not\rightarrow c_1$.

A net representation of the above decision rules obtained by an application of our construction (after some simplifications consisting in the deletion of superfluous arcs) is illustrated in Figure 2. The initial marking of the nets presented in the figure corresponds to the second row of Table 1.

![Figure 2](image)

**Remark.** In Figure 2 places $d_1$ and $d_2$ are drawn separately for readability reasons.

The construction method shortly described above has the following properties (Skowron & Suraj 1994):

**THEOREM 2.** Let $S$ be an information system and let $(N_S, M_S)$ be a marked Petri net representing a system $S$. Then $INF(N_S, M_S) = CON(D(S))$, where $INF(N_S, M_S)$ denotes the set $\{v(M) : M \in R(N_S, M_S)\}$ and $R(N_S, M_S)$ is the reachability set of $N_S$ from $M_S$.

**THEOREM 3.** Let $S$ be an information system, $S'$ its $U'$-extension constructed as above. Then $S'$ is the largest consistent extension of $S$.

**COROLLARY 1.** Let $S'$ be a maximal consistent extension of an information system $S$ constructed by the method presented in the paper. Then $COVERR(S) = COVERR(S')$, for any reduct $R$ of $S$. 
In Figure 3 we show nets representing the external linkings between components $S_1$, $S_3$. These nets also include arcs which guarantee the correctness of the construction.

![Diagram of nets](image)

Figure 3. External linkings between components $S_1$ and $S_3$

In the Petri net $(N_S, M_S)$ constructed for a given information system one can distinguish components and connections between them. The Petri net $(N_S, M_S)$ can be decomposed with respect to any $C$-covering in $\text{COVERR}(S)$, where $R$ is any reduct of $S$ and $C$ is the set of all internal and external linkings of $S$.

**Remark.** A net representation of components $S_1$, $S_3$, and $S_2$ of the system $S$ of Example 1 is shown in Figure 2.

**Conclusions**

The decomposition method has been implemented in C++ and preliminary tests are promising.

Our method can be applied for automatic feature extraction. The properties of the constructed concurrent systems (e.g. their invariants) can be interpreted as higher level laws of experimental data. New features can be also obtained by performing for a given decision table $S = (U, A \cup \{d\})$ the following steps:

**Step 1.** Extract from $S$ a subtable $S_i$ corresponding to the decision $i$, for any $i \in \mathcal{V}_d$, i.e. $S_i = (U_i, A_i)$, where $U_i = \{u \in U : d(u) = i\}$, $A_i = \{a^i : a \in A\}$, and $a^i(u) = a(u)$ for $u \in U_i$.

**Step 2.** Compute the components of $S_i$ for any $i \in \mathcal{V}_d$.

**Step 3.** For a new object $u$ compute the values of components defined on information included in $\text{Inf}_A(u)$ and check in what cases the computed values of components are matching $\text{Inf}_A(u)$ (i.e. they are included in $\text{Inf}_A(u)$). For any $i \in \mathcal{V}_d$ compute the ratio $n_i(u)$ of the number of components matching $\text{Inf}_A(u)$ to all components of $S_i$. The simplest strategy classifies $u$ to the decision class $i_0$, where $n_{i_0}(u) = \max_i n_i(u)$.

We also study some applications of our method in control design from experimental data tables.

**References**


