Some Remarks on Extensions and Restrictions of Information Systems

Zbigniew Suraj *

Institute of Mathematics
Pedagogical University
Rejtana 16A, 35-310 Rzeszów, Poland
e-mail: zsuraj@univ.rzeszow.pl

Abstract. Information systems (data tables) are often used to represent experimental data [4],[5],[8]. In [16] it has been pointed out that notions of extension and restriction of information system are crucial for solving different class of problems [6],[7],[10]-[11],[12]-[15]. The intent of this paper is to present some properties of an information system extension (restriction) and methods of their verification.

Keywords: information systems, indiscernibility relation, discernibility function, minimal rules, information system extension.

1 Introduction

In the rough set theory, the information systems [5] and the rules extracted from information systems are the most common form of representing knowledge. In [16] it has been pointed out that notions of extension and restriction of information system are crucial for solving different class of problems, among others: (i) the synthesis problem of concurrent systems specified by information systems [10],[15], (ii) the problem of discovering concurrent data models from experimental tables [13], (iii) the re-engineering problem for cooperative information systems [12],[14], (iv) the real-time decision making problem [6],[11], (v) the control design problem for discrete event systems [7].

The main idea of an information system extension can be explained as follows: A given information system S defines an extension S' of S created by adding to S all new objects corresponding to known attribute values. If an extension S' of S is consistent with all rules true in S (i.e., any object u from a set of objects U matching the left hand side of the rule also matches its right hand side and there is an object u matching the left hand side of the rule) and S' is the largest (w.r.t. the number of objects in S) extension of S with that property then the system S' is called a maximal consistent extension of S.

Maximal consistent extensions are used in the design of concurrent system specified by information systems [10],[15]. If an information system S specifies a concurrent system then the maximal consistent extension of S represents the

* This work has been partially supported by the grant No. 8T11C 025 19 from the State Committee for Scientific Research in Poland.

W. Ziarko and Y. Yao (Eds.): RSCTC 2000, LNAI 2005, pp. 204–211, 2001.

largest set of global states of the concurrent system consistent with all rules true in S. The set of global states can include new states of the given concurrent system consistent with all rules true in S. Moreover, if we are interested in determining a minimal (w.r.t. the number of objects) description of a given concurrent system then we can compute a minimal consistent restriction of an information system which specifies that concurrent system.

The rest of the paper is structured as follows. A brief presentation of the basic concepts underlying the rough set theory is given in Section 2. The basic definitions and procedures for computing extensions are presented in Section 3.

2 Preliminaries of Rough Set Theory

2.1 Information Systems

An *information system* is a pair $S = (U, A)$, where U - is a non-empty, finite set called the *universe,* A - is a non-empty, finite set of *attributes*, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called the *value set* of a. Elements of U are called *objects*.

The set $V = \bigcup_{a \in A} V_a$ is said to be the *domain* of A.

Example 1. Consider an information system $S = (U, A)$ with $U = \{u_1, u_2, u_3\}$, $A = \{a, b\}$ and the values of the attributes are defined as in Table 1.

U/A	a	b
u_1	0	1
u_2	1	0
u_3	0	0

Table 1. An example of an information system

Let $S = (U, A)$ be an information system and let $B \subseteq A$. A binary relation $ind(B)$, called an *indiscernibility relation,* is defined by $ind(B) = \{(u, u') \in U \times U$ for every $a \in B, a(u) = a(u')\}$. Any information system $S = (U, A)$ determines an *information function* $Inf_A : U \to P(A \times V)$ defined by $Inf_A(u) = \{(a, a(u)) : a \in A\}$ where $V = \bigcup_{a \in A} V_a$ and $P(X)$ denotes the powerset of X. The set $\{Inf_A(u) : u \in U\}$ is denoted by INF(S). The values of an information function will be sometimes represented by vectors of the form $(v_1, ..., v_m), v_i \in V_a$ for $i = 1, ..., m$ where $m = \text{card}(A)$. Such vectors are called *information vectors* (over V and A).

If $S = (U, A)$ is an information system then the *descriptors* of S are expressions of the form (a, v) where $a \in A$ and $v \in V_a$. Instead of (a, v) we also write $a = v$ or a_v. If τ is a Boolean combination of descriptors then by $\parallel \tau \parallel$ we denote the meaning of τ in the information system S.

2.2 Rules in Information Systems

Rules express some of the relationships between values of the attributes described in the information systems.

Let $S = (U, A)$ be an information system and let V be the domain of A.

A *rule* over A and V is any expression of the following form: (1) $a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \Rightarrow a_p = v_p$ where $a_p, a_{i_j} \in A, v_p, v_{i_j} \in V_{a_{i_j}}$ for $j = 1, ..., r$.

A rule of the form (1) is called *trivial* if $a_p = v_p$ appears also on the left hand side of the rule. The rule (1) is *true in* S if $\emptyset \neq \| a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \| \subseteq \| a_p = v_p \|$.

The fact that the rule (1) is true in S is denoted in the following way: $a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a_p = v_p$. By $D(S)$ we denote the set of all rules true in S.

Let $R \subseteq D(S)$. An information vector $\mathbf{v} = (\mathbf{v}_1, ..., \mathbf{v}_m)$ is *consistent* with R iff for any rule $a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a_p = v_p$ in R if $\mathbf{v}_{i_j} = v_{i_j}$ for $j = 1, ..., r$ then $v_p = \mathbf{v_p}$.

2.3 Discernibility Matrix

The *discernibility matrix* and the *discernibility function* [9] help to compute minimal forms of rules w.r.t. the number of attributes on the left hand side of the rules.

Let $S = (U, A)$ be an information system and let $U = \{u_1, ..., u_n\}$, $A = \{a_1, ..., a_m\}$. By $M(S)$ we denote an $n \times n$ matrix (c_{ij}), called the *discernibility matrix* of S, such that $c_{ij} = \{a \in A : a(u_i) \neq a(u_j)\}$ for $i, j = 1, ..., n$.

A *discernibility function* $f_{M(S)}$ for an information system S is a Boolean function of m propositional variables $a_1^*, ..., a_m^*$ (where $a_i \in A$ for $i = 1, ..., m$) defined as the conjunction of all expressions $\bigvee c_{ij}^*$ where $\bigvee c_{ij}^*$ is the disjunction of all elements of $c_{ij}^* = \{a^* : a \in c_{ij}\}$ for $1 \leq j < i \leq n$ and $c_{ij} \neq \emptyset$. In the sequel we write a instead of a^*.

2.4 Minimal Rules in Information Systems

Now we recall a method for generating the minimal (i.e., with minimal left hand sides) form of rules in information systems [10],[15]. The method is based on the idea of Boolean reasoning [1] applied to discernibility matrices defined in [9].

Let $S = (U, A)$ be an information system and $B \subset A$. For every $a \notin B$ we define a function $d_a^B : U \to P(V_a)$ such that $d_a^B(u) = \{v \in V_a :$ there exists $u' \in U\ u'\ ind(B)\ u$ and $a(u') = v\}$ where $P(V_a)$ denotes the powerset of V_a.

Let $S = (U, A)$ be an information system. We are looking for all minimal rules in S of the form: $a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a = v$ where $a \in A, v \in V_a, a_{i_j} \in A$ and $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, ..., r$.

The above rules express functional dependencies between the values of the attributes of S. These rules are computed from systems of the form $S' = (U, B \cup \{a\})$ where $B \subset A$ and $a \in A - B$.

First, for every $v \in V_a, u_l \in U$ such that $d_a^B(u_l) = \{v\}$ a modification $\boldsymbol{M}(S'; a, v, u_l)$ of the discernibility matrix is computed from $M(S')$.
By $\boldsymbol{M}(S'; a, v, u_l) = (c_{ij}^*)$ (or $\boldsymbol{M}$, in short) we denote the matrix obtained from $M(S')$ in the following way:

if $i = l$ **then** $c_{ij}^* = \emptyset$;
if $c_{lj} \neq \emptyset$ and $d_a^B(u_j) \neq \{v\}$ **then** $c_{lj}^* = c_{lj} \cap B$
else $c_{lj}^* = \emptyset$.

Next, we compute the discernibility function f_M and the prime implicants ** [17] of f_M taking into account the non-empty entries of the matrix $\boldsymbol{M}$ (when all entries c^*_{ij} are empty we assume f_M to be always true).

Finally, every prime implicant $a_{i_1} \wedge ... \wedge a_{i_r}$ of f_M determines a rule $a_{i_1} = v_{i_1} \wedge ... \wedge a_{i_r} = v_{i_r} \underset{S}{\Rightarrow} a = v$ where $a_{i_j}(u_l) = v_{i_j}$ for $j = 1, ..., r$, $a(u_l) = v$.

The set of all rules constructed in the above way for any $a \in A$ is denoted by $\mathrm{OPT}(S, a)$. We put $\mathrm{OPT}(S) = \bigcup\{\ \mathrm{OPT}(S, a) : a \in A\}$.

We compute all minimal rules true in $S' = (U, B \cup \{a\})$ of the form $\tau \Rightarrow a = v$, where τ is a term in disjunctive form over B and $V_B = \bigcup_{a \in B} V_a$, with a minimal number of descriptors in any disjunct. To obtain all possible functional dependencies between the attribute values it is necessary to repeat this process for all possible values of a and for all remaining attributes from A.

3 Extensions and Restrictions of Information Systems

Let $S = (U, A)$ be an information system. For $S = (U, A)$, a system $S' = (U', A')$ such that $U \subseteq U'$, $A' = \{a' : a \in A\}$, $a'(u) = a(u)$ for $u \in U$ and $V_a = V_{a'}$ for $a \in A$ will be called an *extension* of S. S is then called a *restriction* of S'.

The number of the extensions of a given information system determines

Proposition 1. *Let S =(U, A) be an information system, k = card(U), n = card($V_{a_{i_1}} \times ... \times V_{a_{i_l}}$) where $a_{i_j} \in A$ for $j = 1, ..., l$ and l = card(A). Then the number of extensions of S is equal to* $2^{n-k} - 1$.

It follows from the following formula: $C^1_{n-k} + C^2_{n-k} + ... + C^{n-k}_{n-k}$ where C^i_j denotes the number of i-element combinations of a set with j-elements.

Let $S = (U, A)$ be an information system and let U'' denotes the set of objects corresponding to all admissible global states of S which do not appear into S, i.e., U'' equals the difference between the cartesian product of the value sets for all attributes $a \in A$ and those from $\mathrm{INF}(S)$. We say that an information system $S' = (U', A)$ is a *maximal extension* of S iff $U' = U \cup U''$.

Proposition 2. Let S =(U, A) be an information system. There exists only one maximal extension S' of S.

Let $S = (U, A)$ be an information system. An information system $S' = (U', A)$ is called a *minimal restriction* of S iff S' is a restriction of S and any restriction S'' of S is an extension of S'.

Proposition 3. A given information system S has at least one a minimal restriction S' of S (different from S).

** An implicant of a Boolean function f is any conjunction of literals (variables or their negations) such that if the values of these literals are true under an arbitrary valuation v of variables then the value of the function f under v is also true. A prime implicant is a minimal implicant. Here we are interested in implicants of monotone Boolean functions only, i.e. functions constructed without negation.

Example 2. Consider the information system S from Example 1. The system S_1 represented by Table 2 is its extension, but the system S_2 obtained from S_1 by deleting the object u_3 is not. S_1 is the maximal extension of S. However, the system S_3 obtained from S_1 by deleting objects u_3 and u_4 is the minimal restriction of S.

U_1/A_1	a	b
u_1	0	1
u_2	1	0
u_3	0	0
u_4	1	1

Table 2. The information system S_1

3.1 Maximal Consistent Extensions of Information Systems

The notion of a maximal consistent extension of a given information system has been introduced in [10]. In this subsection, we present some properties of this notion and procedures for computing the maximal consistent extension.

Let $S' = (U', A)$ be an extension of $S = (U, A)$. We say that S' is a *consistent extension* of S iff $D(S) \subseteq D(S')$. S' is called 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'.

From the above definition follows

Proposition 4. *Let S =(U, A) be an information system. There exists only one maximal consistent extension S' of S.*

PROCEDURE for computing maximal consistent extension S' of S:

`Input:` An information system $S = (U, A)$ and the set OPT(S) of all rules constructed as in subsection 2.4 for S.

`Output:` The maximal consistent extension S' of S.

Step 1. Compute all admissible global states of S which do not appear in S.

Step 2. Verify (using the set OPT(S) of rules) which global states of S obtained in *Step* 1 are consistent with rules true in S.

It is known that, in general, the set OPT(S) of all rules constructed as described in subsection 2.4 can be exponential complexity (w.r.t. the number of attributes). Nevertheless, there are several methodologies allowing to deal with this problem in practical applications (see e.g. [2],[3],[4] pages 3-97).

Proposition 5. *Let S =(U, A) be an information system and S' its maximal consistent extension. The set $OPT(S)$ of all minimal rules of S defined in subsection 2.4 is empty iff S' is equal to the space of all possible values of attributes from A.*

In order to decide if a given information system has the maximal consistent extension the following proposition can be useful.

Proposition 6. *Let $S = (U, A)$ be an information system. If S' is the maximal consistent extension of S different from S then for at least two attributes $a, b \in A$ $card(V_a) > 2$ and $card(V_b) > 2$.*

Proposition 6 constitutes a necessary condition for the existence of the maximal consistent extension of an information system different from that system.

Proposition 7. *Let $S = (U, A)$ be an information system and S' its maximal consistent extension. If there exists such combination of attribute values $(v_{i_1}, v_{i_2}, ..., v_{i_n})$ where $v_{i_j} \in V_{a_j}$ for $j = 1, 2, ..., n$ and $n = card(A)$ that there no exist a functional dependency between any two attribute values from this combination in S, then S' is different from S.*

Proposition 8. *Let $S = (U, A)$ be an information system and S' its maximal consistent extension. S' is the information system in which all new added objects to S have the property mentioned in Proposition 7.*

Example 3. Consider an information system S represented by Table 3. Applying to S the method for generating the minimal form of rules described in subsection 2.4 we obtain the following set OPT(S) of rules: $a_1 \vee a_2 \underset{S}{\Longrightarrow} b_0,\ b_1 \vee b_2 \underset{S}{\Longrightarrow} a_0$.

U/A	a	b
u_1	0	1
u_2	1	0
u_3	0	2
u_4	2	0

Table 3. An example of an information system S

After running the procedure for computing maximal consistent extension of S we obtain the system S' including all objects of the system S and new object u_5 such that $a(u_5) = b(u_5) = 0$.

PROCEDURE for finding a description of maximal consistent extension S' of S:

`Input:` An information system $S = (U, A)$, the set OPT(S) of all minimal rules of S defined in subsection 2.4, and V - the domain of A.

`Output:` A description of maximal consistent extension S' of S in the form of Boolean formula constructed from descriptors over A and V.

Step 1. Rewrite each rule from OPT(S) to the form of Boolean formula.

Step 2. Construct the conjunction of formulas obtained in *Step* 1.

Step 3. Compute prime implicants of the formula obtained in *Step* 2.

In order to find the description of all new elements of maximal consistent extension S' of S (i.e., those from outside of the set of objects U) it is sufficient to execute a procedure presented below.

PROCEDURE for finding a description of all new elements of maximal consistent extension S' of S:

`Input:` As for the procedure presented above.

`Output:` A description of new elements of maximal consistent extension S' of S in the form of Boolean formula constructed from descriptors over A and V.

Step 1. Execute the procedure for finding a description of maximal consistent extension of S.

Step 2. Rewrite each row from the system S to the form of Boolean formula.

Step 3. Construct the negation of the formula obtained in *Step* 2.

Step 4. Construct the conjunction of formulas obtained in *Steps* 2 and 3.

Step 5. Compute prime implicants of the formula obtained in *Step* 4.

Example 4. Consider again the information system S and OPT(S) described in Example 3. Now by applying to S the procedure for finding a description of its maximal consistent extension we obtain the following Boolean formula: $(\neg((a = 1) \vee (a = 2)) \vee (b = 0)) \wedge (\neg((b = 1) \vee (b = 2)) \vee (a = 0))$. After simplifications (using Boolean theory laws) we get the formula of the form: $(a = 0) \vee (b = 0)$. This formula is matching to all objects of S' from Example 3.

In order to find a description of new elements of maximal consistent extension of S represented by Table 3 we can perform the above procedure.

As result we obtain the following Boolean formula: $\neg(((a = 0) \wedge (b = 1)) \vee ((a = 1) \wedge (b = 0)) \vee ((a = 0) \wedge (b = 2)) \vee ((a = 2) \wedge (b = 0))) \wedge ((a = 0) \vee (b = 0))$. After simplifications we get the result formula of the form: $(a = 0) \wedge (b = 0)$. It is matching to the object u_5 of the system S' from Example 3.

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

Remark 1. The information system S from Example 3 is the minimal consistent restriction of the system S' considered in Example 3.

4 Concluding Remarks

The extensions (restrictions) of information systems appear in many investigations related to the rough set methods for solving different class of problems [16]. The presented approach can be treated as a constructive method of the information system extension to the largest data table including the same knowledge as the original information system. Maximal (minimal) consistent extensions (restrictions) provide a basis for modeling concurrent systems using rough set methods.

Acknowledgment

I am grateful to Professor A. Skowron for stimulating discussions and interesting suggestions about this work.

References

1. Brown, E.M.: Boolean Reasoning. Kluwer, Dordrecht (1990)
2. Michalski, R., Bratko, I., Kubat, M. (eds.): Machine Learning and Data Mining. Methods and Applications. Wiley, New York (1998)
3. Nadler, M., Smith, E.P.: Pattern Recognition Engineering. Wiley, New York (1993)
4. Pal, S.K., and Skowron, A. (eds.): Rough-Fuzzy Hybridization. A New Trend in Decision Making, Springer-Verlag, Singapore (1999)
5. Pawlak, Z.: Rough sets - Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
6. Peters, J.F., Skowron, A., Suraj, Z., Pedrycz, W., Ramanna, S.: Approximate Real-Time Decision Making: Concepts and Rough Fuzzy Petri Net Models. International Journal of Intelligent Systems **14**-4 (1998) 4-37
7. Peters, J.F., Skowron, A., Suraj, Z.: An Application of Rough Set Methods in Control Design. Fundamenta Informaticae (to appear)
8. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 2. Applications, Case Studies and Software Systems. Physica-Verlag, Heidelberg (1998)
9. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.): Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory, Kluwer, Dordrecht (1992) 331-362
10. Skowron, A., Suraj, Z.: Rough Sets and Concurrency. Bulletin of the Polish Academy of Sciences **41**-3 (1993) 237-254
11. Skowron, A., Suraj, Z.: A Parallel Algorithm for Real-Time Decision Making: A Rough Set Approach. Journal of Intelligent Information Systems **7**, Kluwer, Dordrecht (1996) 5-28
12. Suraj, Z.: An Application of Rough Set Methods to Cooperative Information Systems Re-engineering. In: Tsumoto, S., Kobayashi, S., Yokomori, T., Tanaka, H., Nakamura, A. (eds.): Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery (RSFD-96), Tokyo (1996) 364-371
13. Suraj, Z.: Discovery of Concurrent Data Models from Experimental Tables: A Rough Set Approach. Fundamenta Informaticae **28**-3,4 (1996) 353-376
14. Suraj, Z.: Reconstruction of Cooperative Information Systems under Cost Constraints: A Rough Set Approach. Journal of Information Sciences **111** (1998) 273-291
15. Suraj, Z.: The Synthesis Problem of Concurrent Systems Specified by Dynamic Information Systems. In: [8] 418-448
16. Suraj, Z.: Rough Set Methods for the Synthesis and Analysis of Concurrent Processes, ICS PAS Reports **893** (1999). Also in: Polkowski, L. (eds.): Studies in Fuzziness and Soft Computing, Physica-Verlag, Heidelberg (to appear)
17. Wegener, I.: The complexity of Boolean functions. Wiley and B.G. Teubner, Stuttgart (1987)