

Guarded Transitions in Rough Petri Nets

J.F. Peters¹, A. Skowron², Z. Suraj³, S. Ramanna¹

¹Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 2N2 Canada

²Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland

³Institute of Mathematics, Pedagogical University, Rejtana 16A, 35-310 Rzeszów, Poland

ABSTRACT: This paper considers the construction of Petri nets to simulate conditional computation in various forms of systems. Coloured Petri nets underly the definition of a family of Petri nets based on rough set theory. Two families of guards on transitions are introduced: Lukasiewicz guards and rough guards. Lukasiewicz guards provide a basis for transitions with a form of continuous enabling. The notion of level-of-enabling of transitions is introduced. Rough guards on transitions are derived from approximations of our knowledge of input values. There are a number of outstanding issues connected with this research, and not considered in this paper. In the case of a Lukasiewicz guard on a transition t , an unbounded number of values in the interval $(0, 1]$ can enable t . It is possible to introduce the notion of “bandwidth” of an input arc to t . The bandwidth of an input arc is proportional to the width of a subinterval $(a, 1]$ of $(0, 1]$ containing values which enable t . In the case where the bandwidth is greater than 1, transition t is decomposed into a family of transitions concurrently enabled by values of $(a, 1]$. A number of refinements of the family of Lukasiewicz guards are also possible. In addition, the bandwidth of an input arc on a transition t with a rough guard can be considered. A transition with a rough guard can be decomposed into two families L and U of concurrent transitions: transitions in L enabled by inputs relative a lower approximation, and transitions U enabled by inputs relative to an upper approximation

KEYWORD: decision systems, enabling, guard, multivalued logic, Petri net, rough sets.

1 INTRODUCTION

Considerable work has already been carried out in modeling various forms of systems with Petri nets in Skowron and Suraj (1993, 1994, 1996), Peters (1997), Peters et al. (1998a, 1998b, 1998c). The aim of the earlier as well as the current research has been to provide a complete framework for approximate reasoning, especially in the context of rough set theory from Pawlak (1982, 1991, 1998). Rough set theory provides an inductive approach to reasoning about data. This paper returns to the idea of rough Petri nets in Peters et al. (1998a, 1998c). Two families of guarded transitions based on multivalued logic and rough sets are introduced. This paper also introduces the notion of level-of-enabling of guarded transitions in Petri nets.

2 ROUGH PETRI NETS

In what follows, it is assumed that the reader is familiar with classical Petri nets in Petri (1962), coloured Petri nets in Jensen (1992), and rough set theory in Pawlak (1991). Rough Petri nets are derived from coloured and hierarchical Petri nets as well as from rough set theory. A rough Petri net provides a basis for modeling, simulating and analyzing decision systems.

2.1 BASIC DEFINITIONS

A rough Petri net (rPn) is a structure $(S, P, T, A, N, C, G, E, I, W, \text{Proc})$ where

- S is a finite set of non-empty data types called color sets.
- N is a 1-1 node function where $N: A \rightarrow (P \times T) \cup (T \times P)$.
- C is a color function where $C: P \rightarrow S$.
- G is a guard function where $G: T \rightarrow \text{Set_of_Boolean_Expressions}$.
- E is an arc expression function where $E: A \rightarrow \text{Set_of_Expressions}$ where $E(a)$ is an expression of type $C(p(a))$ and $p(a)$ is the place component of $N(a)$.
- I is an initialization function where $I: P \rightarrow \text{Set_of_Closed_Expressions}$ where $I(p)$ is an expression of type $C(p)$.
- W is a set of strengths-of-connections where $W: A \rightarrow W$.
- $\text{Proc} = \{ \text{Proc} \mid \text{Proc is a procedure which constructs } \{ \text{rough set structure} \} \text{ from a set of information granules} \}$

Let U, S, A, d be a set of inputs, information system S , attributes of S , decision d , respectively. Examples of rough set structures constructed by from information granules are the decision system $S = (U, A \cup \{d\})$ and the set $OPT(S)$ of all rules derived from reducts of a decisions system table for S . Borrowing from coloured Petri nets, a rough Petri net provides data typing (colour sets) and sets of values of a specified type for each place. The expression $E(p, t)$ specifies the input associated with the arc from input place p to transition t , and the expression $E(t, p')$ specifies a transformation (activity) performed by transition t on its inputs $\{E(p, t)\}$ to produce an output for place p' .

2.2 EXAMPLE

A guard $G(t)$ is an enabling condition associated with transition t . A sample CPN is given in Fig. 1, where place p_1 supplies a value of x of type item to transition t , which outputs $\text{complement}(x) = 1 - x$ to place p_2 whenever the token x in place p_1 satisfies the guard $[x \geq 0.35]$. The notation $1 \cdot 0.4 + 4 \cdot 0.3$ specifies that a multiset contains 1 element with the value 0.4 and four elements with value 0.3. A *multiset* is a set which can have multiple appearances of the same element. The prefix 1 indicates the number of tokens in the multiset (in this case, one token in place p_1), and the suffix indicates that x has been assigned an x -value. Whenever transition t fires, it augments the multiset associated with place p_2 .

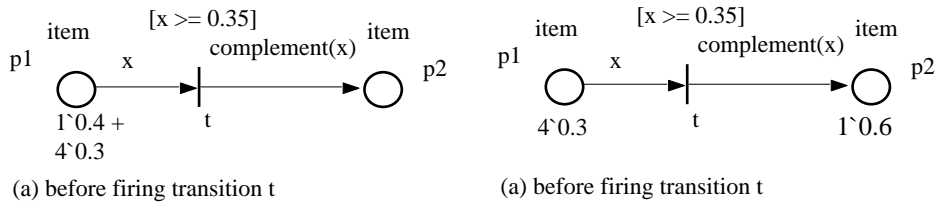


Figure 1: Sample Coloured Petri Net

3 CONDITIONAL ENABLING OF TRANSITIONS

In rough Petri nets, a number of families of guards can be defined by appealing to the basic notion of approximation in rough sets and multi-valued truth functions. What we call an ordinary guard is “inherited” from the coloured Petri net framework used to define rough Petri nets. The notation $P(x)$ denotes a propositional function. The truth value of $P(x)$ depends on the value of x . A propositional function $P(x)$ is Boolean, if the values of $P(x)$ are from $\{0, 1\}$. A guard in coloured Petri nets is what we term an ordinary guard.

Def. 1 *Ordinary Guard*. An *ordinary guard* is a Boolean-valued propositional function $P(x)$ on a transition t with input x such that $P(x)$ must evaluate to 1 before t can be enabled. Otherwise, t is not enabled if $P(x) = 0$.

Borrowing from the notation of probability theory, we also write $P(x = k)$ to denote the fact that the evaluation of the propositional function P depends on the assertion $x = k$.

Example 1. The guard in Fig. 1 can be expressed as a propositional function of the form $P(x \geq 0.35)$ which returns 1 for values of x which satisfy the inequality. Otherwise, $P(x \geq 0.35) = 0$.

Example 2. Consider a dosing tank like the one shown in Fig. 2. Assume that there is a drain sensor and a pump sensor.

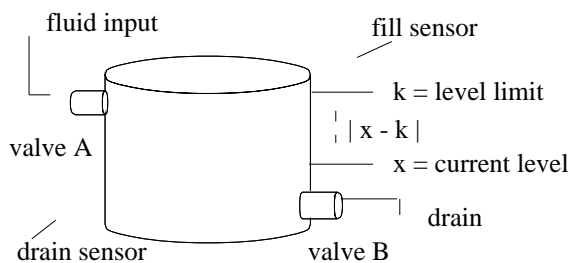


Figure 2: Dosing Tank

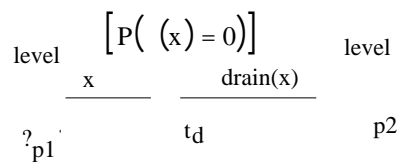


Figure 3: Guarded Rough Petri Net

In this example, the focus is on the operation of the drain sensor. The drain valve B is opened to lower the liquid level in the tank in cases where the drain sensor detects liquid levels rising above the height of a pre-set limit k . Further, let

x be the current liquid level and let k be the limit level on the height of the liquid in the tank. Assume that x and k are normalized over the interval $[0, 1]$. Let $s(x) = |k - x|$. Then introduce the guard $P(s(x) = 0)$ on transition t_d in the Petri net in Fig. 3. The function $\text{drain}(x)$ associated with transition t_d in Fig. 3 opens valve B in Fig. 2 whenever transition t_d fires. Transition t_d is enabled in the case where $s(x) = 0$. Intuitively, $\text{drain}(x) = \text{“drain off a fraction of the liquid in the tank until } s(x) > 0\text{.”}$ The place p_1 in Fig. 3 is called the start place relative to sensor $s(x)$ and transition t_d . We make the assumption that a start place is always ready to receive input. We further assume that a start place is part of a process which waits for input from the environment. In effect, a start place belongs to a receptive process which models the interaction between a system and its environment. The notion of a receptive process was introduced by Dill (1989). The rate of filling or draining is not considered in this example. The Petri net in Fig. 3 is the beginning of a model of a reactive system which is always input ready and which controls the fluid level in a dosing tank.

3.1 LEVEL-OF-ENABLING

In a rough Petri net, various families of guards can be defined which induce a level-of-enabling of transitions. Consideration of level-of-enabling stems from guards named after Jan Lukasiewicz (1920), who inaugurated the study of multivalued logic. Let U denote a universe of objects, and let $X \subseteq U$. Further, let f denote a mapping $f: U \rightarrow [0, 1]$.

Def. 2 Lukasiewicz Guard. A Lukasiewicz guard on transition t with input x is a higher order propositional function $P(f(x))$ labeling the transition t with input x and output $f(x)$. The guard $P(f(x))$ returns the value of the sensor $f(x)$, which must have a value greater than 0 to enable t . Otherwise, t is not enabled if $P(f(x)) = 0$. Also, observe that the value of $f(x)$ is used as an argument of the function labeling the outgoing edge from transition t .

A Lukasiewicz guard $P(f(x))$ on a transition t with input x is a higher order propositional function in the sense that its evaluation depends on the value returned by the function $f(x)$. This contrasts with an ordinary guard $P(x)$ with an evaluation which depends on x . It should be noted that we test the condition $f(x) > 0$ and this is not new. $f(x)$ is also used to specify the degree of the output of t relative to function labeling the outgoing arc of transition t (e.g., the degree of openness of a valve controlled by a pump function in Fig. 4).

Example 3.

Consider control of the inlet valve A in Fig. 2. Let the opening of the inlet valve be proportional to $|k - x|$ in the sense that as the value of $|k - x|$ decreases, the opening of the valve decreases until it is completely closed when $|k - x| = 0$. Let $s(x)$ be a sensor which returns a value of $|k - x|$. Further, let $P(s(x))$ be a Lukasiewicz guard on transition t_p in the Petri net in Fig. 4. The function $\text{pump}(s(x))$ associated with transition t_p computes the size of the opening of input valve A used to

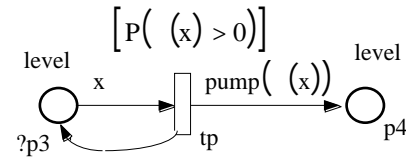


Figure 4: Lukasiewicz Guarded Transition

pump fluid into the tank in Fig. 2. In other words, transition t is enabled for all values $P(s(x) > 0)$. Guarded and unguarded enabling of a transition t define a two-state machine (t is either enabled (state 1) or not enabled (state 0)). Intuitively, the notion of enabling a transition is defined informally as follows.

Def. 3 [informal]. A transition t in a Petri net is enabled if the information required by t is available.

Def. 4 [informal]. A transition t with ordinary guard g is enabled if required inputs to t are available and g tests true.

Using a Lukasiewicz guard makes it possible to define (i) a form of continuous enabling and (ii) level of enabling. To see this, we first give a formal definition of enabling a transition in a coloured Petri net. Let $p, P, t, \langle b \rangle, Y, M(p), E(p, t)$ be a place $p \in P$ (set of places) for transition t , binding b of a value to a variable relative to t , step Y in a multiset (bag of typed variables), marking of place p in P , arc expression relative to p and t .

Def. 5 A transition t is enabled in a marking M of a Petri net if $p \in P$.
$$E(p, t) \setminus \langle b \rangle \subseteq M(p)$$

$$(t, b) \in Y$$

In other words, the marking M of the input places to a transition t results in the binding of each $E(p, t)$, which enables t . In the case of a transition with a Lukasiewicz guard, enabling occurs over a subinterval of $(0, 1]$.

Def. 6.1 A transition t with Lukasiewicz guard $P(s(x) > 0)$ is enabled in M if $p \in P$.
$$(E(p, t) \setminus \langle b \rangle) \subseteq M(p)$$

Def. 6.2 A transition t with Lukasiewicz guard $P(s(x) > 0)$ is enabled if in M if $p \in P$.
$$(E(p, t) \setminus \langle b \rangle) \subseteq M(p)$$
 and $s(x) > 0$.

The integration of a Lukasiewicz guard on transition t in Def. 6.1 (6.2) is defined relative to an infinite number of inputs which enable transition t . A Lukasiewicz guard on a transition t is defined as a continuous function, which causes t to be enabled to some degree or level. The level-of-enabling of μ -guarded transition equals $(E(p,t) < b >)$, a real number in $(0, 1]$. Each new stimulus $E(p,t) < b >$ will result in a new enabling of t at a possibly different level. Feedback to input place p from t guarantees continuous enabling of t , even if no stimulus arrives.

Axiom 0. The condition sensor $(E(p,t) < b >)$ induces a level-of-enabling of a transition t in a rough Petri net.

Examples of physical analogs of level-of-enabling of rough transitions are numerous. Consider, for instance, the presence of a mechanism (a form of faucet) which makes it possible to adjust the size of the opening of valve A in Fig. 2. Assuming that the fluid input pressure is fixed, the extent that valve A is open controls the rate of fluid flow into the tank. Similarly, the level-of-enabling (and firing) conceptual construct can be used in some sense to affect the outcome of firing a transition which is constrained by a Lukasiewicz guard.

Example 4.

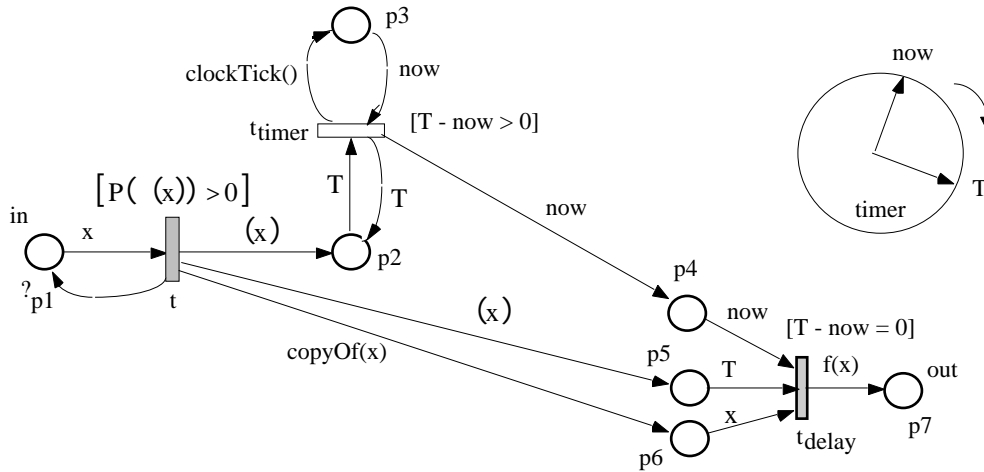


Figure 5: Timer-controlled Transition

As a second example of a μ -guarded transition t , consider the model of a delayed reaction to a stimulus x in Fig. 5. The guard (x) outputs deadline T to $p2$ and $p5$. The current time (now) is output by t to $p3$, and copied to $p4$. Notice that transition t fires continuously. The level-of-enabling of t depends (x) , which puts time limit T in $p2$ and $p5$. The value of T determines the length of delay before transition t_{delay} is enabled. In addition, we make the following assumption based on our interest in the value of a condition sensor (x) , which controls the enabling of a transition.

Axiom 1. If $(x) \geq k$, then $P((x) \geq k) = (x)$. Otherwise, $P((x) \geq k) = 0$.

In effect, the notation $P((x) \geq k)$ asserts that $P((x) \geq k) = (x)$ for all x such that $(x) \geq k$.

Prop. 1. If $k = 0$ and $(x) > 0$, then $P((x) > 0) > 0$.

From Axiom 1 and Def. 2, we have

Prop. 2 If a Lukasiewicz guard $P((x) \geq k)$ enables a transition t in a rough Petri net, then $P((x) \geq k) \in (0, 1]$.

3.2 ROUGH GUARDS

The notion of conditional execution is enriched by considering an approximate reasoning approach to guards on transitions in a rough Petri net.

Def. 7 A rough function μ is defined relative $x \in X \rightarrow U$, lower approximation $\underline{\mu}X$ and upper approximation $\overline{\mu}X$ such that

$$\begin{aligned}
 & 1, \text{ if } x \in \underline{BX} \quad [\text{boundedness}] \\
 P(x) = & v \in (0,1), \text{ if } x \in \overline{BX} - \underline{BX} \\
 & 0, \text{ if } x \in U - \overline{BX} \quad [\text{boundedness}]
 \end{aligned}$$

In other words, a rough function is bounded above by 1 and bounded below by 0. From the definition of a rough function, we have

Prop. 3 $P(x) \in [0, 1]$.

Rough functions are used to define what we call rough guards. The notation $P(x)$ is defined as follows.

$$P(x) = \left(x \in X \cdot P(x) = v(x) \right)$$

Def. 8 Rough Guard. A rough guard is a higher order propositional function $P(x)$ on transition t with input x . The guard $P(x)$ returns the value of the sensor $v(x)$, which must be greater than 0 to enable t . Otherwise, t is not enabled if $P(x) = 0$.

By definition, we obtain

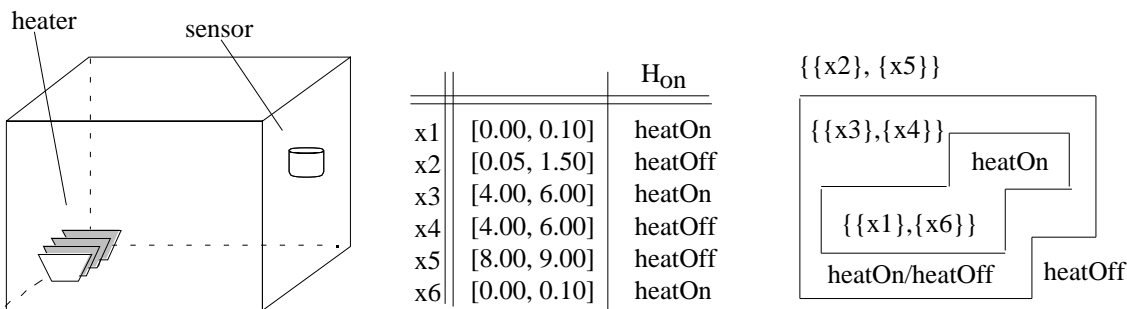
Prop. 4 Every rough guard is a Lukasiewicz guard.

We make a basic assumption about the restriction of propositional functions which evaluate rough functions. We introduce a threshold condition k on a rough function.

$$\text{Axiom 2. [Restriction]} \quad P(x) = \begin{cases} 1, & \text{if } v(x) > k \\ 0, & \text{otherwise} \end{cases}$$

In effect, we assume that a rough guard $P(x)$ returns 1 only in cases where the value of a rough function $v(x)$ is greater than some threshold k .

Example 5. Consider a large room where a uniform temperature must be achieved with a combination of sensors and heating elements. Ambient temperatures are a problem. It may be warm in one part of the room, and cold in another part of the room. For simplicity, a single sensor and single heating element are modeled in Fig. 6. Let $v(x)$ denote the current temperature recorded by a "smart" sensor relative to a stimulus x , and let $H_{on} = \{ x \mid H_{on}(x) = \text{heatOn} \}$. A sample decision table reflecting reactions of the system to sensed temperatures is shown in Fig. 7. The decision H_{on} has a value chosen by an expert for any global temperature state (represented by measurements of temperature in all points of a sufficiently dense set of points in a room). The attribute H_{on} is giving the values only in some points chosen by a smart sensor. For example, $v(x_1)$ returns a value in a vector of temperature measurements $[0.00, 0.10]$ for different points in a room. The corresponding approximating regions are shown in Fig. 8.



The lower approximation $\underline{H}_{on} = \{x1, x6\}$ and the upper approximation $\overline{H}_{on} = \{x1, x3, x4, x6\}$. From Def. 7, we know that $P(x) > 0$ in cases where stimulus x belongs to either the upper or lower approximation sets. Then $P(x)$ can be used as a rough guard on transition t_h in the Petri net in Fig. 9. The operation $heat(x)$ turns a heater on each time transition t_h is enabled and fires. In the event that x_2 or x_5 stimulate the sensor, transition t_h is not enabled. In that case, $P(x) = 0$ by definition, and this enables transition t_0 in Fig. 9. Each time transition t_0 fires, the operation off causes the room heater to turn off. The operation $copy(x)$ puts a copy of x in place p_3 (the input place for t_0) each time transition t_h fires. However, transition t_0 will only be enabled if its guard is satisfied.

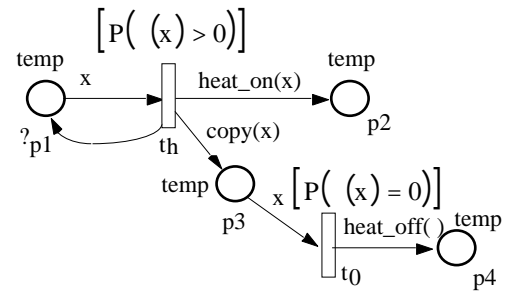


Figure 9: Petri Net with Rough Guards

4 CONCLUDING REMARKS

The contribution of this paper is the extension of rough Petri nets to include two families of guards besides “ordinary” guards inherited from coloured Petri nets. These families draw upon multivalued logic introduced by Lukasiewicz and rough set theory introduced by Pawlak. In addition, the notion of level-of-enabling is introduced, briefly. As a result, it is possible to model continuous reaction to stimuli in decision systems.

REFERENCES

- Dill, D.L., 1989, “Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits”, Cambridge, MA, MIT Press.
- Jensen, K., 1992, “Coloured Petri Nets--Basic Concepts, Analysis Methods and Practical Use 1”, Berlin, Springer-Verlag.
- Lukasiewicz, J., 1920, “O logice trojwartosciowej”, *Ruch Filozoficzny* 5, pp. 170-171. See “On three-valued logic”, English translation in L. Borkowski (Ed.), *Jan Lukasiewicz: Selected Works*, Amsterdam, North-Holland, 1970, pp. 87-88.
- Pawlak, Z., 1982, “Rough sets”, *Int. J. of Computer and Information Sciences*, Vol. 11, pp. 341-356.
- Pawlak, Z., 1991, “Rough Sets: Theoretical Aspects of Reasoning About Data”, Boston, MA, Kluwer Academic Publishers.
- Pawlak, Z., 1998, “Reasoning about data: A rough set perspective”, In: *Proc. Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98)*, Warsaw, Poland, Lecture Notes in Artificial Intelligence 1424, Berlin, Springer-Verlag, pp. 25-34.
- Petri, C.A., 1962, “Kommunikation mit Automaten”, *Schriften des IIM Nr. 3*, Institut für Instrumentelle Mathematik, Bonn, West Germany.
- Petri, C.A., 1996, “Nets, time and space”, *Theoretical Computer Science*, Vol. 153, pp. 3-48.
- Peters, J.F., 1997, “Time and clock information systems: Concepts and roughly fuzzy Petri net models”, In: L. Polkowski, A. Skowron (Eds.), *Rough Sets in Knowledge Discovery*. Heidelberg, Physica-Verlag, Vol. 2, pp. 385-418.
- Peters, J.F.; Skowron, A.; Suraj, Z.; Pedrycz, W.; Ramanna, S., 1998a, “Approximate Real-Time Decision Making: Concepts and Rough Fuzzy Petri Net Models”, *International Journal of Intelligent Systems*, Vol. 14, no. 4, pp.4-37.
- Peters, J.F.; Ziaei, K.; Ramanna, S., 1998b, “Approximate Time Rough Control: Concepts and Application to Satellite Attitude Control, In: *Proc. Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98)*, Warsaw, Poland, Lecture Notes in Artificial Intelligence 1424, Berlin, Springer-Verlag, pp. 491-498.
- Peters, J.F.; Skowron, A.; Suraj, Z.; Ramanna, S.; Paryzek, A., 1998c, “Modeling real-time decision-making systems with rough fuzzy Petri nets”, *Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT98)*, Vol. 1, Aachen, Germany, pp. 985-989.
- Skowron, A.; Suraj, Z., 1993, “A rough set approach to real-time state identification for decision making”, *Institute of Computer Science Research Report 18/93*, Warsaw University of Technology.
- Skowron, A.; Suraj, Z., 1994, “Synthesis of concurrent systems specified by information systems”, *ICS Research Report 39/94*, Institute of Computer Science, Warsaw Institute of Technology.
- Skowron, A.; Suraj, Z., 1996, “A parallel algorithm for real-time decision making: a rough set approach”, *Journal. of Intelligent Systems*, Vol. 7, pp. 5-28.