

Zagadnienia redukcji podsieci automatowych w sieciach Petriego

Dr inż. Andrei Karatkevich



Dr inż. Andrei Karatkevich ukończył w roku 1993 studia na Wydziale Techniki Obliczeniowej Mińskiego Radiotechnicznego Instytutu. Obronił pracę doktorską w Białoruskim Państwowym Uniwersytecie Informatyki i Radiotechniki w roku 1998. Od roku 2000 jest adiunktem w Instytucie Informatyki i Elektroniki na Uniwersytecie Zielonogórskim. Jego zainteresowania naukowe dotyczą głównie teorii sieci Petriego oraz zagadnień analizy i weryfikacji współbieżnych systemów sterowania.

e-mail: A.Karatkevich@iie.uz.zgora.pl

Streszczenie

Tematem przedstawionego referatu jest pewien aspekt redukcyjnego podejścia do analizy sieci Petriego, mianowicie redukcja podsieci automatowych. Zaproponowana metoda takiej redukcji zastępuje podsieć automatową przez jej minimalną reprezentację, otrzymaną przez usunięcie miejsc wewnętrznych oraz minimalizację ilości miejsc wejściowych i wyjściowych.

Słowa kluczowe: sieci Petriego, podsieci automatowe, redukcja.

Abstract

This paper addresses such aspect of reduction approach to analysis of Petri nets, as reduction of the subnets being the SM-components. Proposed method of reduction replaces such subnet by its minimal representation obtained by removing all interior places and minimization of number of input and output places.

Keywords: Petri nets, SM-components, reduction.

1. Wstęp

Sieci Petriego [10] są popularnym modelem przedstawiającym współbieżne procesy dyskretne w jasny i czytelny sposób. Głównym problemem, ograniczającym ich zastosowanie, jest złożoność analizy takich sieci, której przyczyną jest wykładnicza zależność ilości osiągalnych znakowań (stanów) od rozmiaru sieci. Dlatego w analizie sieci Petriego często przydatne są metody redukcyjne [9][10], pozwalające uprościć sieć, zachowując przy tym takie jej właściwości, które chcemy przeanalizować (na przykład żywotność i bezpieczeństwo). Z reguły metody redukcyjne są stosowane jako pomocnicze (zredukowana sieć jest analizowana przez inne metody, takie jak eksploracja przestrzeni stanów lub badanie niezmienników), choć w niektórych przypadkach do sprawdzenia pewnych właściwości sieci sama redukcja okazuje się wystarczająca [12].

Zaskakujące jednak jest to, że niektóre z najważniejszych metod redukcyjnych nie mogą sobie poradzić z takim pozornie łatwym przypadkiem, jak podsieci czysto sekwencyjne, czyli należące do klasy automatów stanów (SM-nets). Na przykład klasyczne 6 operacji redukcji, przedstawione w [10], nie mogą zredukować takiej podsieci w przypadku ogólnym (czyli pisząc bardziej precyzyjnie, nie usuną takich jej miejsc, które są wejściowymi dla więcej niż jednej tranzycji).

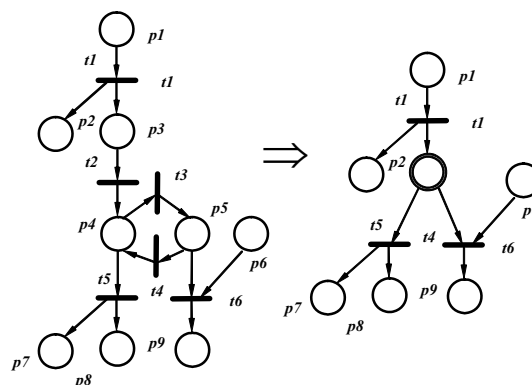
Lepsze rezultaty pod tym względem można uzyskać przy pomocy metody zaproponowanej przez Andre [7] i wykorzystanej m. in. w licznych publikacjach Adamskiego [1][2][3][4][5][6][8]. Istota tej metody została opisana w [8] następująco: „Metoda Andre polega na wykorzystaniu makrosieci, zawierającej wyłącznie tranzycje wielowejsiowe i wielowyjsiowe oraz makromiejsca typu automatowego. Makromiejsca te reprezentują sekwencyjne części sieci, przedstawione przez składowe spójne grafy, powstałe po usunięciu tych tranzycji”.

Metoda Andre, opracowana głównie w celach syntezy, równie dobrze może służyć do analizy, gdyż, jak łatwo jest zauważyć, bezpieczeństwo i ograniczoność sieci są niezmiennikami operacji,

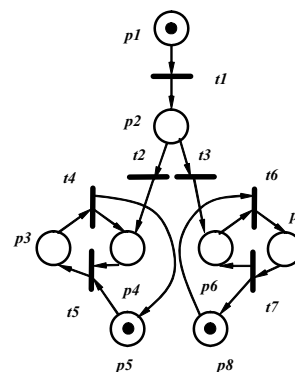
zastępującej podsieć automatową przez pojedyncze (makro)miejsce.

Ale czy *każda* spójna składowa grafu sieci, zawierająca tylko tranzycje o jednym wejściu i jednym wyjściu, może zostać zastąpiona przez pojedyncze miejsce? Oczywiście jest, że nie każda.

Wszystkie podsieci automatowe, występujące w przykładach zastosowania metody Andre w wymienionych publikacjach, są „blokami dobrze zbudowanymi” (*well-formed blocks*). To znaczy między innymi, że mają one nie więcej niż dwa miejsca, incydentne z tranzycjami, nie należącymi do podsieci (jedno wejście i jedno wyjście). Nasuwa się pytanie, a co w przypadku, jeśli takich miejsc jest więcej? Kilka miejsc wejściowych przy jednym wyjściu nie stanowi jeszcze problemu, lecz kilka miejsc wyjściowych – już tak. Jeśli jednak wszystkie miejsca wyjściowe są wzajemnie osiągalne w podsieci, wówczas może ona być zastąpiona przez jedno miejsce (rys. 1). W przeciwnym przypadku takie zastąpienie nie jest możliwe, przykład czego został pokazany na rys. 2.



Rys. 1. Przykład redukcji podsieci z kilkoma miejscami wyjściowymi
Fig. 1. An example of reduction of a subnet with several output places



Rys. 2. Przykład podsieci, której nie można zastąpić przez jedno miejsce (tranzycje $t1, t2, t3$ składają się na sieć automatową, ale jej zastąpienie przez jedno miejsce zmienia nieżywą sieć w żywą)

Fig. 2. An example of a subnet which cannot be replaced by single place (transitions $t1, t2, t3$ constitute an SM-net, but replacing it by single place would transform a non-live net into a live one)

Jest intuicyjnie oczywiste, że podsieć automatową, której nie da się zastąpić przez pojedyncze miejsce, często można jednak uprościć. Między innymi, zawsze można wyeliminować miejsca wewnętrzne, łącząc w odpowiedni sposób miejsca wejściowe z wyjściowymi (na przykład w sieci pokazanej na rys. 2 można usunąć miejsce $p2$). Celem niniejszego referatu jest zaproponowanie ogólnej metody postępowania z podsieciami automatowymi w trakcie redukcji sieci Petriego.

2. Metoda redukcji podsieci typu automatowego

W celu przedstawienia metody konieczne jest wprowadzenie definicji kilku pojęć.

Def. 1. *Podsiecią automatową* sieci Petriego $\Sigma = (P, T, F)$ jest taka jej spójna podsieć $\Sigma' = (P', T', F')$, że $\forall t \in T': |t| = |t'| = 1$; $P' = {}^*T' \cup T'^*$; $F' = ((P' \times T') \cup (T' \times P')) \cap F$.

Def. 2. *Wejściem* podsieci automatowej $\Sigma' = (P', T', F')$ jest miejsce $p \in P'$ takie, że $\exists t \in (T \setminus T') : p \in {}^*t$.

Def. 3. *Wyjściem* podsieci automatowej $\Sigma' = (P', T', F')$ jest miejsce $p \in P'$ takie, że $\exists t \in (T \setminus T') : p \in t^*$.

Def. 4. *Miejsce wewnętrzne* podsieci automatowej $\Sigma' = (P', T', F')$ jest miejsce $p \in P'$, nie należące ani do wejść, ani do wyjść Σ' . Zbiór miejsc wewnętrznych Σ' oznacza się jako P^{int} .

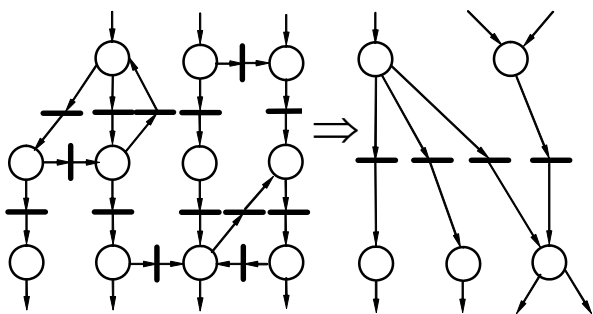
(Pozostałe pojęcia i oznaczenia dot. sieci Petriego patrz w [10].)

Zakładamy, że zbiór miejsc podsieci automatowej, która zostanie zredukowana, nie zawiera blokady ani pułapki (względem Σ). Zakładamy ponadto, że miejsca wewnętrzne podsieci nie są oznakowane w znakowaniu początkowym.

Proponowana metoda redukcji podsieci automatowej składa się z następujących kroków:

1. Podzielić wyjścia podsieci na maksymalne zbiory wzajemnie osiągalnych w podsieci miejsc. Wprowadzić dla każdego takiego zbioru O_i dodatkowe miejsce o_i , przy czym $M_0(o_i) = M_0(O_i)$. Jeśli wszystkie wyjścia są wzajemnie osiągalne, wprowadzić tylko jedno miejsce $o = i$ ($M_0(o) = M_0(P')$) i przejść do 4.
2. Dla każdego wejścia podsieci znaleźć zbiór osiągalnych wyjść. Podzielić wejścia podsieci na maksymalne zbiory miejsc, z których w podsieci są osiągalne takie same zbiory wyjść. Wprowadzić dla każdego takiego zbioru I_j dodatkowe miejsce i_j , przy czym $M_0(i_j) = M_0(I_j)$.
3. Dla każdej pary o_i, i_j wprowadzić tranzycję t taką, że $\{i_j\} = {}^*t$, $\{o_i\} = t^*$, jeśli z wejść I_j osiągalne są wyjścia O_i .
4. Każdy łuk, prowadzący od tranzycji t , nie należącej do podsieci, do wejścia $p \in I_j$, zastąpić łukiem, prowadzącym od t do i_j . Każdy łuk, prowadzący od wyjścia $p \in O_i$ do tranzycji t , nie należącej do podsieci, zastąpić łukiem, prowadzącym od t do o_i (uwaga: w wyniku tej operacji mogą powstać łuki wielokrotne).
5. Usunąć wszystkie miejsca i tranzycje podsieci,

Przykład zastosowania opisanej metody przedstawiono na rys. 3.



Rys. 3. Przykład redukcji

Fig. 3. An example of reduction

Głównym rezultatem teoretycznym danego artykułu jest następujące twierdzenie.

Twierdzenie 1. Żywotność, bezpieczeństwo i ograniczoność sieci Petriego są niezmiennikami przekształcenia, opisanego powyżej.

Zanim będzie można udowodnić to twierdzenie, trzeba udowodnić twierdzenie pomocnicze. Niech $\Sigma = (P, T, F)$ będzie siecią Petriego, $\Sigma' = (P', T', F')$ – jej podsiecią automatową, $\Sigma^R = (P^R, T^R, F^R)$ – siecią otrzymaną z Σ po zastosowaniu opisanego przekształ-

cenia do podsieci Σ' . Niech $F(M)$ będzie funkcją zdefiniowaną dla wszystkich znakowań M sieci Σ takich, że $M(P^{int}) = 0$, i obliczającą dla M odpowiednie znakowanie M^R sieci Σ^R w ten sam sposób, w który w punkcie 1 zaproponowanej metody jest obliczane znakowanie początkowe sieci Σ^R , odpowiadające M_0 .

Twierdzenie 2. Niech dla Σ istnieje sekwencja tranzycji σ taka, że $M_0 \sigma M$, $M(P^{int}) = 0$. Wtedy i tylko wtedy dla Σ^R istnieje sekwencja tranzycji σ' taka, że $M_0^R \sigma' M^R$, gdzie $M^R = F(M)$.

Dowód. Jeśli podsieć automatowa została zastąpiona przez pojedyncze miejsce, twierdzenie jest oczywiste. Przypadkiem, wymagającym dowodu, jest sytuacja, w której podsieć zostaje zastąpiona przez prostszą sieć automatową

($\Rightarrow$) Jeśli wszystkie tranzycje w σ należą do $(T \setminus T')$, twierdzenie jest oczywiste. Niech w σ będą tranzycje należące do T' . Wtedy takie tranzycje muszą składać się na jedną lub więcej dróg z wejścia do wyjścia Σ' , co wynika z tego, że miejsca należące do P^{int} , nie są oznakowane ani w M_0 ani w M . Każdej takiej drodze odpowiada jedna tranzycja w M^R . Zastąpienie w σ każdej podsekwencji odpowiadającej drodze w Σ' (która nie może być przedłużona przez tranzycje, należące do σ), przez odpowiednią tranzycję pozwala uzyskać sekwencję σ' taką, że $M_0^R \sigma' M^R$.

($\Leftarrow$) Niech $M_0^R \sigma' M^R$. Jeśli wszystkie tranzycje w σ' należą do $(T \setminus T')$, twierdzenie jest oczywiste. Niech w σ' są tranzycje należące do $(T^R \setminus T)$. Każda taka tranzycja według konstrukcji odpowiada drodze lub kilka drogami z wejścia do wyjścia Σ' . Zastąpienie w σ' każdej takiej tranzycji przez sekwencję tranzycji Σ' , składającej się na odpowiednią drogę, pozwala uzyskać sekwencję σ taką, że $M_0 \sigma M$ w Σ .

Dowód Twierdzenia 1. Żywotność:

Niech Σ jest siecią żywą. Przypuśćmy, że Σ^R nie jest żywa. Wtedy dla Σ^R istnieje sekwencja tranzycji $M_0^R \sigma' M^R$ taka, że tranzycja $t \in T^R$ nie jest przygotowana w żadnym znakowaniu, osiągalnym z M^R w Σ^R . Bez utraty ogólności możemy założyć, że $t \in T$. Z Twierdzenia 2 wynika, że w Σ osiągalne jest znakowanie M takie, że $F(M) = M^R$. Z żywotności sieci Σ wynika, że istnieje sekwencja tranzycji σ taka, że $M \sigma M'$, gdzie tranzycja t jest przygotowana w M' . Bez utraty ogólności możemy stwierdzić, że $M'(P^{int}) = 0$ (inaczej wystarczy dodać do σ tranzycje, wyprowadzające znaczniki z miejsc P^{int} , co zawsze jest możliwe, ponieważ P^{int} nie zawiera pułapek). Zastosujmy ponownie Twierdzenie 2, traktując M jako znakowanie początkowe, a M' jako znakowanie docelowe. Wynika z niego, że w Σ^R osiągalne jest znakowanie $M'^R = F(M')$. W tym znakowaniu tranzycja t jest przygotowana, gdyż jak wynika ze sposobu przekształcania znakowania przez funkcję F , każde jej miejsce wejściowe ma w M'^R potrzebną ilość znaczników. Wobec założenia, że Σ^R nie jest żywa, prowadzi to do sprzeczności, z czego wynika, że Σ^R jest żywa.

Niech Σ^R jest siecią żywą. Przypuśćmy, że Σ nie jest żywa. Wtedy dla Σ istnieje sekwencja tranzycji $M_0 \sigma M$ taka, że tranzycja $t \in T$ nie jest przygotowana w żadnym znakowaniu osiągalnym z M w Σ . Bez utraty ogólności możemy założyć, że $t \in (T \setminus T')$ oraz że $M(P^{int}) = 0$. W takim razie z Twierdzenia 2 wynika, że w Σ^R osiągalne jest znakowanie M^R takie, że $F(M) = M^R$. Z żywotności sieci Σ^R wynika, że istnieje sekwencja tranzycji σ' taka, że $M^R \sigma' M'^R$, gdzie tranzycja t jest przygotowana w M'^R . Niech M odpowiada w Twierdzeniu 2 znakowaniu początkowemu, a M' docelowemu. W takim razie z tego twierdzenia wynika, że w Σ osiągalne jest znakowanie M' takie, że $M'^R = F(M')$. Tranzycja t może nie być przygotowana w M' tylko w takim przypadku, jeśli jedno (lub więcej) z jej miejsc wejściowych p należy do podzbioru O_i wyjść Σ' , zastąpionych przy redukcji przez jedno miejsce o_i , przy czym p nie ma w M' wystarczającej ilości znaczników. Według opisanej metody cały zbiór O_i zawiera ilość znaczników wystarczającą dla wszystkich miejsc wejściowych t należących do O_i , co wynika z przygotowania t w M^R . Ponadto, wszystkie miejsca, należące do O_i , są wzajemnie osiągalne w sieci automatowej, a to znaczy, że znaczniki wewnątrz O_i można przemieszczać w dowolny sposób przy pomocy odpowiednich sekwencji tranzycji T' . Wobec założenia, że t nie jest przygotowana w M' , to istnieje taka sekwencja

tranzycji σ'' (składająca się wyłącznie z tranzycji, należących do T'), że $M'\sigma''M''$ i t jest przygotowana w M'' . Prowadzi to do sprzeczności, więc Σ jest żywa.

Bezpieczeństwo:

Funkcja $F(M)$ każde niebezpieczne znakowanie przekształca na znakowanie, które też jest niebezpieczne, więc z Twierdzenia 2 wynika, że jeśli sieć Σ jest niebezpieczna, sieć Σ^R też jest niebezpieczna. Niech Σ będzie siecią bezpieczną. Załóżmy, że istnieje osiągalne w Σ znakowanie M takie, że $M(P^{int}) = 0$, $M^R = F(M)$, $\forall p \in P^R: M^R(p) > 1$. Wtedy $p \in (P^R \setminus P)$, ponadto p jest miejscem, które zastąpiło w wyniku redukcji zbiór albo wejść I sieci Σ' , albo jej wyjść O , albo wszystkich jej miejsc P' , przy czym $M(I) > 1$, $M(O) > 1$, albo $M(P') > 1$, odpowiednio. W pierwszym przypadku ze wszystkich miejsc, należących do I , dostępne jest to samo wyjście o Σ' , więc istnieje sekwencja tranzycji σ taka, że $M\sigma M'$ i $M'(o) > 1$, a to oznacza, że sieć Σ nie jest bezpieczna. W drugim przypadku z faktu, że wszystkie miejsca O są wzajemnie osiągalne w sieci automatowej, wynika, że istnieje sekwencja tranzycji σ' taka, że $M\sigma M''$ i $\exists o \in O: M''(o) > 1$, czyli sieć Σ znowu okazuje się niebezpieczna. W trzecim przypadku również istnieje co najmniej jedno miejsce wyjściowe sieci Σ' , w którym może się znaleźć więcej niż 1 znacznik. Ta sprzeczność dowodzi, że jeśli Σ jest bezpieczna, Σ' też jest bezpieczna.

Ograniczoność:

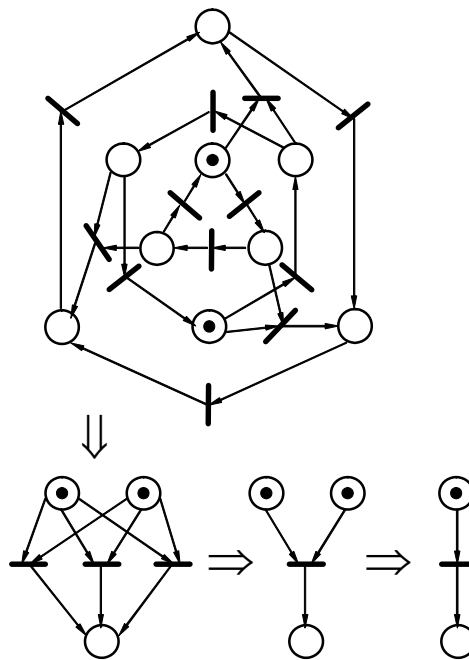
Łatwo jest zauważyć, że dla każdego możliwego znakowania M^R sieci Σ^R (niekoniecznie osiągalnego) istnieje co najmniej jedno znakowanie M sieci Σ takie, że $F(M) = M^R$. Więc z Twierdzenia 2 wynika, że ilość osiągalnych znakowań Σ^R nie może być większa, niż ilość osiągalnych znakowań Σ , a to znaczy, że jeśli Σ jest ograniczona, Σ^R też musi być ograniczona. Niech Σ będzie siecią nieograniczoną. Przypuśćmy teraz, że Σ^R jest ograniczona. W takim razie nieograniczone miejsca muszą należeć do P^{int} , inaczej odpowiednie miejsca byłyby nieograniczone w Σ^R . Ale jeśli istnieje nieograniczone miejsce $p \in P^{int}$, to nieograniczonym też jest wyjście o osiągalne z p . Więc odpowiednie miejsce sieci Σ^R też okazuje się nieograniczone i sieć Σ^R jest nieograniczona.

3. Podsumowanie i wnioski

1. Intuicyjnie, zaproponowana metoda zastępuje podsieć automatową przez jej "minimalną reprezentację", tzn. najprostszą sieć, która zachowuje się wobec reszty sieci tak samo, jak podsieć, którą ona reprezentuje. Wejściowe miejsca, z których są osiągalne takie same wyjścia oraz wzajemnie osiągalne wyjściowe miejsca w zaproponowanej metodzie są łączone.
2. Czy można osiągnąć taką samą redukcję, używając znanych metod? Tak, ale nie zawsze, ponadto używając kombinacji różnych metod i/lub wielokrotnie stosując inne operacje redukcji. Operacja *podstawienia*, zaproponowana przez Zakrewskiego [12], pozwala pozbyć się wszystkich miejsc wewnętrznych podsieci automatowej, ale musi być w tym celu stosowana wielokrotnie, usuwając za każdym razem jedno miejsce wewnętrzne. Stosując metodę Andre, rozumianą w sposób „rozszerzony” (to znaczy do podsieci, które mogą mieć więcej niż jedno wejście i jedno wyjście) do silnie spójnych składowych sieci automatowej, obejmujących jej wyjścia, można dokonać agregacji wyjść. Autorowi nie jest znana żadna metoda, pozwalająca agregować miejsca wejściowe.
3. Oczywiście, zaproponowana operacja redukcji może być łączona z innymi operacjami zachowującymi badane właściwości sieci. Można podać przykłady sieci, w których po zastosowaniu zaproponowanej operacji pojawia się możliwość zastosowania innych znanych operacji. Na przykład, na rys. 3 przedstawiono sieć, mającą 3 cykliczne podsieci automatowe, każda z których może zostać zastąpiona przez jedno miejsce. Następnie do sieci można zastosować operację łączenia równoległych tranzycji i łączenia równoległych miejsc [10], wynikiem czego jest sieć o jednej tranzycji i

dwóch miejscach, która w sposób oczywisty jest nieżywa i bezpieczna. W przypadkach, w których podsieć automatowa nie jest zastępowana przez pojedyncze miejsce, zawsze można zastosować do wejść otrzymanej „minimalnej reprezentacji” operację podstawienia [12], co pozwoli jeszcze zmniejszyć ilość miejsc sieci.

4. Warto zauważyć, że w [11] została zaproponowana metoda, wobec której metoda przedstawiona powyżej jest w pewnym sensie dualną. Mianowicie, w [11] omawia się zastępowanie przez minimalną reprezentację podsieci, należących do klasy grafów znakowań (*event graphs*).



Rys. 4. Jeszcze jeden przykład redukcji
Fig. 4. One more example of reduction

References

- [1] Adamski M.: Projektowanie układów cyfrowych systematyczną metodą strukturalną, Wydawnictwo Wyższej Szkoły Inżynierskiej, Zielona Góra, 1990.
- [2] Adamski M.: Parallel Controller Implementation using Standard PLD Software, FPGAs: International Workshop on Field Programmable Logic and Applications, Abingdon EE&CS Books, 1991, 296-304.
- [3] Adamski M., Węgrzyn M.: Field Programmable Implementation of Concurrent State Machine, Proceedings of the International Conference CAD DD, Vol. 1, 1999, 4-12.
- [4] Adamski M.: A Rigorous Design Methodology for Reprogrammable Logic Controllers, Proceedings of the International Workshop DESDes, 2001, 53-60.
- [5] Adamski M.: Formal Logic Design of Reprogrammable Controllers, Design of Embedded Control Systems, Springer 2005, 15-26.
- [6] Adamski M.: Reconfigurable Logic Controller for Embedded Applications, Proceedings of the International Workshop DESDes, 2006, 147-152.
- [7] Augin M., Boeri F., Andre C.: Systematic Method of Realization of Interpreted Petri Nets, Digital Processes, № 6, 1980, 55-68.
- [8] Banaszak Z., Kuś J., Adamski M.: Sieci Petriego. Modelowanie, sterowanie i synteza systemów dyskretnych, Wydawnictwo Wyższej Szkoły Inżynierskiej, Zielona Góra, 1993.
- [9] Berthelot G.: Checking Properties of Nets using Transformation, Advances in Petri Nets 1985, LNCS Vol. 222, Springer 1986, 19-40.
- [10] Murata T.: Petri Nets: Properties, Analysis and Applications, Proceedings of IEEE, Vol. 77, № 4, April 1989, 548-580.
- [11] Savi V.M., Xie X.: Liveness and Boundedness Analysis for Petri Nets with Event Graph Modules, Application and Theory of Petri Nets 1992, LNCS Vol. 616, Springer 1992, 328-347.
- [12] Zakrewski A.D.: O poprawności współbieżnych algorytmów sterowania logicznego, Izwiestija AN SSSR. Techniczna Cybernetyka, № 4, 1987, 106-112 (po rosyjsku).

Title : On Reduction of SM-Subnets in Petri Nets.

Artykuł recenzowany