Andrzej KARATKIEWICZ
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

Analiza cyklicznych sieci Petriego przy pomocy dekompozycji blokowej

Dr inż. Andrej KARATKIEWICZ

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 i jej zastosowań w dziedzinie analizy i weryfikacji współbieżnych systemów sterowania.

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

Streszczenie

Artykuł przedstawia metodę analizy żywotności i bezpieczeństwa sieci Petriego, na których znakowanie początkowe i strukturę nałożone są pewne ograniczenia, typowe dla sieci cyklicznych, modelujących algorytmy sterowania. Metoda jest wzorowana na metodzie analizy operacyjnych sieci Petriego i jest jej zaadaptowaniem do innej klasy sieci. Przedstawiona metoda polega na dekompozycji sieci i konstruowaniu przestrzeni osiągalności bloków, w kolejności wyznaczonej strukturą sieci. W artykule przytoczono wyniki eksperymentów, opisujące stopień redukcji przestrzeni stanów.

Słowa kluczowe: sieci Petriego, dekompozycja, eksploracja stanów.

Analysis of Cyclic Petri Nets by Means of Block Decomposition

Abstract

The paper presents a methods of deciding of liveness and safeness of Petri nets with certain restrictions imposed on their structure and initial marking, which are typical for cyclic nets, modeling the control algorithms. The method is based on the method of analysis of operational Petri nets, and it is an adaptation of this method to another class of nets. The method decomposes the net and explores state spaces of its blocks in an order depending on the net structure. The experimental results are presented, demonstrating reduction of state space.

Keywords: Petri nets, decomposition, state exploration.

1. Wstęp

Sieci Petriego są szeroko stosowane w informatyce jako formalny model opisu współbieżnych procesów i systemów, takich jak współbieżne algorytmy, asynchroniczne układy i komunikacyjne protokoły. Analiza takich sieci jest problemem ważnym i skomplikowanym, gdyż nawet prosta sieć może mieć dużą ilość osiągalnych stanów.

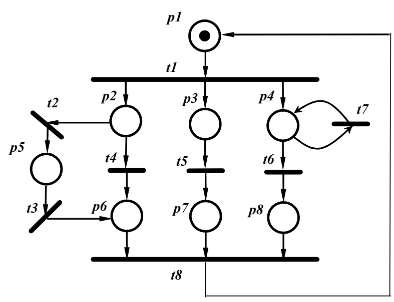
Obiecującym podejściem do analizy sieci Petriego i ogólnie rzecz biorąc, współbieżnych systemów dyskretnych, jest dekompozycja [1, 4, 7, 11, 14-16]. W artykule przedstawiono zastosowanie tej idei do podklasy cyklicznych sieci. Metoda ta sprowadza analizę sieci do analizy bloków dekompozycji, które mogą być wielokrotnie mniejsze od badanej sieci. Takie podejście może znacznie ułatwić analizę dużych sieci.

Zaproponowana metoda pozwala stwierdzić, czy dana sieć jest „poprawnie zbudowana" – z reguły właśnie takie sieci odpowiadają poprawnym systemom.

2. Sieci Petriego

Sieci Petriego [10, 12] można opisać jako dwudzielny graf skierowany z dwoma rodzajami wierzchołków, nazywanych *miejscami* i *tranzycjami*. Miejsca i tranzycje są połączone przez łuki. *Znakowanie* wyznacza położenie w miejscach sieci tak zwanych *znaczników*. Znakowanie może się zmieniać przez *odpalenie* tranzycji (reguły odpalenia można znaleźć w [9, 10, 12]). Istotnymi właściwościami sieci Petriego są *żywotność* i *bezpieczeństwo*. Sieć jest *żywa*, jeśli dla każdej tranzycji *t* z każdego znakowania *M* jest osiągalne znakowanie *M'*, w którym tranzycja *t* jest *aktywna* (może zostać odpalona). Sieć jest *bezpieczna*, jeśli w każdym z osiągalnych znakowań żadne miejsce nie zawiera więcej niż jeden znacznik. Sieć, która z każdego osiągalnego znakowania może powrócić do znakowania początkowego, nazywana jest siecią *cykliczną*. Sieć żywa i bezpieczna często nazywana jest siecią *poprawnie zbudowaną* (ang. *well-formed*); takie sieci, zwłaszcza cykliczne, mają szczególne znaczenie praktyczne. Znakowanie takie, przy którym żadna tranzycja w sieci nie jest aktywna, nazywane jest *zastojem* sieci. Na rys. 1 pokazany jest przykład sieci Petriego.

Szczegółowe definicje i notacje nie są tu przedstawione z powodu braku miejsca; patrz [10, 12].



Rys. 1. Sieć Petriego
Fig. 1. A Petri net

Operacyjna siec Petriego (OSP) jest siecią, mającą *wejściowe* i *wyjściowe* miejsca, przy czym tylko wejściowe miejsca mogą zawierać znaczniki w znakowaniu początkowym; jeśli sieć jest poprawna dla danego znakowania początkowego, to ze wszystkich osiągalnych znakowań jest osiągalne znakowanie *końcowe*, w którym znaczniki są tylko w miejscach wyjściowych (poza tym, poprawna OSP jest bezpieczna). Taka sieć nie jest silnie spójna. Szczegóły o OSP patrz w [8, 14-16].

3. Analiza operacyjnych sieci Petriego

Głównymi problemami analizy operacyjnych sieci Petriego są: sprawdzenie, czy dana sieć jest poprawna dla danego znakowania początkowego, oraz, jeśli tak, znajdowanie osiągalnych z niego końcowych znakowań. Każde z tych zadań może być rozwiązane przez konstruowanie przestrzeni osiągalnych znakowań sieci, ale rozmiar tej przestrzeni zależy w sposób wykładniczy od rozmiaru sieci, a w zastosowaniach spotykane są sieci, mające kilkaset miejsc i tranzycji. Oczywiście, takie sieci praktycznie nie można zbadać przez eksplorację wszystkich osiągalnych stanów. Analizę sieci jednak można uprościć, stosując dekompozycję. Metoda takiej analizy, opracowana przez A. Zakrewskiego [14-16], jest krótko omówiona poniżej.

Dwie tranzycje są w relacji *alternatywnego połączenia*, jeśli mają one wspólne wejściowe lub wyjściowe miejsca. Tranzytywne domknięcie tej relacji zadaje podział OSP na minimalne *bloki*. Jeśli dla bloków T_i i T_j istnieje takie miejsce, że jest ono wejściowym dla T_j i wyjściowym dla T_i, to nie można analizować T_j przed T_i. W ten sposób określona jest relacja *V* praporządku na zbiorze bloków, która po połączeniu cykli (krok 2 algorytmu 1) staje się relacją częściowego porządku. Poniżej został przedstawiony algorytm analizy [16]:

Algorytm 1

Wejście: OSP T i jej znakowanie początkowe M_0.

Wyjście: zbiór końcowych znakowań, jeśli sieć jest poprawna, w przeciwnym razie komunikat o niepoprawności.

1. Dokonujemy dekompozycji sieci T na minimalne bloki.
2. Łączymy wszystkie cykle, tworzone przez bloki.
3. $D := \{M_0\}$.
4. Dopóki D zawiera co najmniej jedno znakowanie, w którym znaczniki są w wewnętrznych miejscach T, wykonujemy:
    - 4.1. Znajdujemy blok T_i taki, że nie ma takiego bloku T_j, że $(T_j,T_i) \in V$ i T_j jeszcze nie był analizowany.
    - 4.2. Dla każdego znakowania $M \in D$ takiego, że są znaczniki w wejściowych miejscach T_i, znajdujemy znakowania końcowe T_i, osiągalne z M, i zastępujemy M w D przez te znakowania (znaczniki, które znajdują się poza miejscami T_i, nie zmieniają swojego położenia).
    - 4.3. Jeśli co najmniej dla jednego z początkowych znakowań T_i ten blok okazał się niepoprawnym, to przechodzimy do kroku 6.
5. D zawiera wszystkie znakowania końcowe, osiągalne w T z M_0. STOP.
6. Sieć T niepoprawna dla M_0. STOP.

Szczegółowe uzasadnienie algorytmu i wyniki przeprowadzonych eksperymentów można znaleźć w [14-16].

Metoda „upartych zbiorów" A. Valmariego [13] ma cele, podobne do celów omawianej metody. Niektóre wyniki analizy porównawczej tych metod patrz w [5].

4. Analiza cyklicznych sieci Petriego

Cykliczna sieć Petriego, spełniająca pewne warunki, może być „rozłożona" w operacyjną sieć Petriego. Taka transformacja jest najprostsza, jeśli w znakowaniu początkowym sieć zawiera tylko jeden znacznik. Warunek ten można uogólnić – jeśli wszystkie tranzycje, których miejsca wejściowe zawierają znaczniki w znakowaniu początkowym, mają ten sam zbiór miejsc wejściowych, to sieć można transformować w OSP; tak samo, jak w przypadku gdy podobny warunek jest spełniony dla miejsc *wyjściowych*. Łatwo jest zauważyć, że jeśli dana sieć jest żywa i bezpieczna, to odpowiednia OSP będzie quasi-żywa i poprawna dla dokładnie jednego znakowania początkowego, z którego będzie osiągalne dokładnie jedno znakowanie końcowe.

Jeśli więc sieć Petriego ma być cykliczna i spełnia jeden z powyższych warunków, to może ona zostać transformowana w OSP i analizowana przy pomocy opisanej metody. Analiza taka pozwoli sprawdzić żywotność i bezpieczeństwo sieci (tej przed transformacją). Poza tym, jeśli sieć nie jest żywa lub nie jest bezpieczna, analiza pozwoli znaleźć niebezpieczne znakowanie lub takie, z którego nieosiągalne jest znakowanie początkowe, oraz ciąg znakowań, prowadzący od M_0 do tego niepożądanego znakowania, lub zbiór martwych tranzycji.

Z powyższego wynika, że przy pomocy następnego algorytmu (algorytm 2, modyfikacja algorytmu, przedstawionego w [16]) można badać żywotność i bezpieczeństwo takiej sieci, w której wszystkie znaczniki mogą być usunięte ze wszystkich miejsc, znakowanych w znakowaniu początkowym (*lub* dodane do wszystkich tych miejsc) tylko jednocześnie (przez odpalenie jednej tranzycji). Oznacza to, między innymi, że algorytm ten pozwala badać α-sieci [15]. Algorytm może być łatwo rozszerzony w ten sposób, że dla sieci, która nie jest poprawnie zbudowana, pozwoli zlokalizować niepoprawność, jak opisano powyżej. Warto jednak zauważyć, że takie lokalizowanie dla całej sieci (a nie pojedynczego bloku) wymaga przechowywania w pamięci wszystkich eksplorowanych znakowań; inaczej wystarczy pamiętać tylko zbiór D, który może być o rzędy wielkości mniejszy.

Algorytm 2

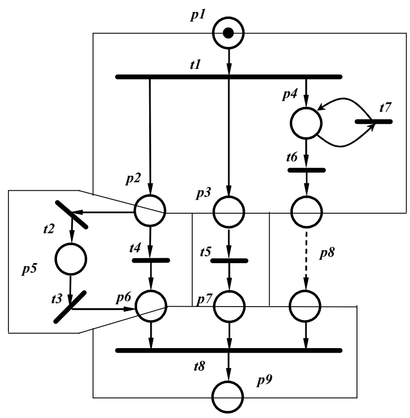
Wejście: sieć Petriego T i jej znakowanie początkowe M_0., spełniające warunki.

Wyjście: komunikat o właściwościach sieci.

1. Dodajemy do zbioru miejsc P nowe miejsce dla każdego z miejsc znakowanych w M_0. Zamieniamy wszystkie łuki sieci, prowadzące do miejsc znakowanych w M_0 na łuki, prowadzące do odpowiednich nowych miejsc (z tych samych tranzycji). Otrzymujemy w ten sposób OSP T'. Miejsca znakowane w M_0 są jej miejscami wejściowymi, a nowe miejsca – miejscami wyjściowymi.
2. Analizujemy T' dla znakowania początkowego M_0, stosując Algorytm 1.
3. Jeśli sieć T' jest niepoprawna:
    - 3.1. jeśli żadne znakowanie końcowe nie jest osiągalne z M_0:
        - 3.1.1. jeśli jest osiągalne takie znakowanie, w którym są oznakowane wszystkie miejsca wyjściowe i jednocześnie jakieś miejsca wewnętrzne, to T nie jest bezpieczna;
        - 3.1.2. w przeciwnym razie T nie jest żywa;
    - 3.2. jeśli znaleziono niebezpieczne miejsce, to T nie jest bezpieczna.
4. Jeśli sieć T' jest poprawna:
    - 4.1. jeśli jest osiągalne takie znakowanie końcowe, że nie wszystkie miejsca wyjściowe mają znaczniki, lub niektóre tranzycje nigdy nie zostały odpalone w trakcie symulacji, to T nie jest żywa;
    - 4.2. w przeciwnym razie T jest poprawnie zbudowana.
5. STOP.

5. Przykład i badania eksperymentalne

Sieć, przedstawiona na rys. 1, może być transformowana w operacyjną sieć Petriego, której dekompozycję blokową przedstawiono na rys. 2.



Rys. 2. Dekompozycja operacyjnej sieci Peteriego
Fig. 2. Decomposition of an operational Petri net

Sieć, przedstawiona na rys. 2 ma 32 osiągalne znakowania. Jej badanie przy pomocy opisanej metody wymaga eksploracji tylko 7 znakowań.

Tabela 1 opisuje wyniki eksperymentów, w których porównywano ilość znakowań, eksplorowanych przy wykorzystaniu opisanej metody, z ilością wszystkich osiągalnych znakowań (wyniki zostały otrzymane przez T. Jasiukiewicza podczas wykonania pracy dyplomowej [3], której promotorem był autor). Eksperymenty pokazują, że metoda wymaga eksploracji średnio około 40% całej przestrzeni osiągalnych stanów dla zbadanych przykładów.

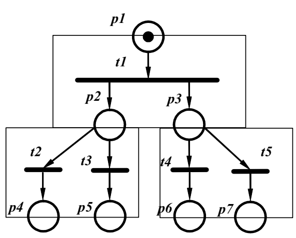
Z otrzymanych rezultatów wynika, że im większa sieć, tym większa redukcja przestrzeni stanów, czyli stosowanie metody bardziej się opłaca dla dużych sieci. Natomiast w przypadku sieci, u których ilość miejsc jest znacznie mniejsza od ilości tranzycji, metoda ta jest mniej skuteczna; struktura takich sieci daje mniej możliwości dla głębokiej dekompozycji.

Tab. 1. Redukcja przestrzeni stanów przy zastosowaniu dekompozycji blokowej
Tab. 1. Reduction of state space by the block decomposition method

Sieć №	Osiągalne stany	Zbadane stany	%
1	49	11	22,45
2	66	23	34,85
3	43	13	30,23
4	19	7	36,84
5	47	10	21,28
6	11	11	100,00
7	26	13	50,00
8	10	7	70,00
9	126	31	24,60
10	51	16	31,37
11	29	9	31,03
12	141	41	29,08
Średnio	**51,5**	**16**	**40,14**

6. Dekompozycja i „uporczywe zbiory"

Jedną z przyczyn skuteczności dekompozycyjnego podejścia do analizy jest to, że to podejście pozwala „wielokrotnie wykorzystywać" wyniki analizy bloków systemu. Analiza przez eksplorację przestrzeni stanów (w tym zredukowanej, wykorzystując metody „uporczywych zbiorów" – *persistent set methods*, PSM [2]) bez dekompozycji, z reguły na to nie pozwala i może prowadzić do wielokrotnej analizy tych samych części systemu. Patrz rys. 3: przy pełnej eksploracji przestrzeni stanów znakowania dwóch z bloków tej sieci będą eksplorowane trzykrotnie; przy zastosowaniu PSM jeden z tych bloków będzie zbadany dwukrotnie. Dekompozycja pozwala tego uniknąć.



Rys. 3. Sieć Petriego z dekompozycją blokową
Fig. 3. A Petri net and its block decomposition

Z drugiej strony, dla pełnej analizy bloku trzeba znać wszystkie jego możliwe stany początkowe. To nakłada pewne ograniczenia na możliwości zastosowania takiego podejścia, i dlatego metoda analizy przez dekompozycję blokową, tak samo jak metoda analizy przez dekompozycję hierarchiczną [4], przeznaczona jest dla sieci bezpiecznych.

Bliskie pokrewieństwo między metodą dekompozycji a PSM opisuje następujące twierdzenie, którego dowodu tu nie podajemy z powodu braku miejsca (dowód można znaleźć w [6]).

Twierdzenie

Niech T będzie OSP; niech T_1 będzie jej blokiem, i niech w znakowaniu M są aktywne tranzycje, należące do T_1, i nie ma w T takiego bloku T_2, że w T_2 są aktywne tranzycje i $(T_2, T_1) \in R$, gdzie R jest tranzytywnym domknięciem relacji V, zdefiniowanej w sekcji 3. Wtedy wszystkie aktywne tranzycje, należące do T_1, stanowią „uporczywy zbiór" (*persistent set*).

Metoda dekompozycji blokowej i PSM, takie jak metoda „upartych zbiorów" A. Valmariego [13], mają zbliżone do siebie cele. Powstają więc następne pytania: po pierwsze, która z tych metod jest lepsza w konkretnym przypadku, i po drugie, czy istnieje możliwość połączenia tych metod.

Odpowiedzieć na te pytania można następująco: PSM są bardziej uniwersalne, niż metoda dekompozycji blokowej, która, jak to zostało opisane powyżej, może być stosowana tylko do określonych klas sieci. Z drugiej strony, dla każdego bloku w tej metodzie są poszukiwane zastoje, dlatego wykorzystanie metod „uporczywych zbiorów" dla analizy pojedynczych bloków byłoby oczywistym ulepszeniem metody dekompozycji blokowej. Jednak tu zachodzi pewne ograniczenie: jeśli należy sprawdzić bezpieczeństwo sieci, to nie można zastąpić eksploracji całej przestrzeni stanów bloku przez częściową eksplorację, bo w niektórych przypadkach taka eksploracja może ominąć niebezpieczne znakowania.

Generalnie rzecz ujmując, metoda dekompozycji i metody „uporczywych zbiorów" mogą być skutecznie łączone ze sobą.

7. Podsumowanie

Prowadzone badania wskazują, że dekompozycja blokowa pozwala znacznie ułatwić analizę właściwości sieci Petriego, należących do klas, które mają szerokie zastosowanie praktyczne – na przykład w modelowaniu algorytmów sterowania. Metoda ta, w zaproponowanej postaci, pozwala na sprawdzanie, czy sieć jest „poprawnie zbudowana", oraz konkretnie wskazać niepożądany stan (i dojście do niego), co pozwala w istotnym zakresie weryfikować systemy, modelowane przez sieci Petriego, które należą do określonych klas. Więcej szczegółów na temat rozmaitych metod dynamicznej analizy sieci i możliwych kombinacji między nimi można znaleźć w [6].

8. Literatura

[1] Banaszak Z., Kuś J., Adamski M.: Sieci Petriego. Modelowanie, Sterowanie i Synteza Systemów Dyskretnych, WSInż, Zielona Góra, 1993.
[2] Godefroid P.: Partial-Order Methods for the Verification of Concurrent Systems: an Apprpach to the State-Explosion Problem, LNCS, Vol. 697, Springer, 1996.
[3] Jasiukiewicz T. System analizy sieci Petriego poprzez badanie przestrzeni stanów. Praca dyplomowa. Uniwersytet Zielonogórski, 2004.
[4] Karatkevich A.: Hierarchical Decomposition of Safe Petri Nets, Proceedings of the International Conference on CAD DD'99, 1999,. 34-39.
[5] Karatkevich A.: Optimal Simulation of α-Nets, Proceedings of the Polish-German Symposium SRE, 2000, 217-222.
[6] Karatkevich A.: Dynamic Analysis of Petri Net-Based Discrete Systems, to be published by Springer in 2007.
[7] Karatkevich A.G., Gratkowski T.: Analysis of the Operational Petri Nets by a Distributed System, Proceedings of TCSET'2004, February 2004, 319-322.
[8] Karatkevich A., Zakrevskij A. Analysis of Petri Nets by means of Concurrent Simulation, Proceedings of the International Conference PARELEC, September 2002, IEEE, 87-91.
[9] Kulikowski J.L.: Zarys teorii grafów. PWN, 1986.
[10] Murata T.: Petri Nets: Properties, Analysis and Applications, Proceedings of IEEE, Vol. 77, № 4, April 1989, 548-580.
[11] Notomi M., Murata T.: Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent-Software Analysis. IEEE Transactions on software engineering, Vol. 20, No 5, May 1994, 325-336.
[12] Peterson J.L.: Petri Net Theory and the Modeling of Systems, Prentice-Hall, Inc., 1981.
[13] Valmari A.: State of the Art Report: Stubborn Sets, Petri Nets Newsletter, № 46, 1994, 6-14.
[14] Закревский А.Д.: Анализ операционных сетей Петри, Препринт № 4 / ИТК АН БССР, 1987.
[15] Закревский А.Д.: Параллельные алгоритмы логического управления, ИТК НАНБ, 1999.
[16] Zakrevskij A., Karatkevich A., Adamski M.: A Method of Analysis of Operational Petri Nets, in Proceedings of the 8th International Conference ACS'2001, 449-460, Kluwer Academic Publishers, 2002.

Artykuł recenzowany