

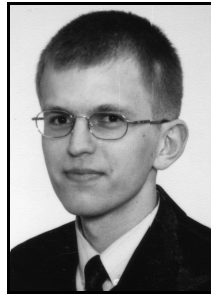
Piotr BUBACZ, Marian ADAMSKI

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Reprezentacja przestrzeni stanów sterownika logicznego z wykorzystaniem kodowanych diagramów decyzyjnych

Mgr inż. Piotr BUBACZ

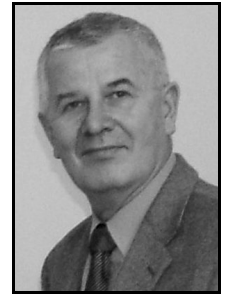
Asystent w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego, absolwent Zintegrowanych Studiów Zagranicznych Uniwersytetu Zielonogórskiego i Fachhochschule Giessen-Friedberg (Niemcy). Zainteresowania badawcze obejmują sieci komputerowe, projektowanie systemów cyfrowych oraz formalnych metod oprogramowania sterowników logicznych.



e-mail: P.Bubacz@iie.uz.zgora.pl

Prof. dr hab. inż. Marian ADAMSKI

Dyrektor Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego. Zainteresowania badawcze obejmują projektowanie mikrosystemów cyfrowych oraz formalne metody programowania sterowników logicznych. Członek IEEE, IEE, ACM, Polskiego Towarzystwa Elektrotechniki Teoretycznej i Stosowanej oraz Polskiego Towarzystwa Informatycznego.



e-mail: M.Adamski@iie.uz.zgora.pl

Streszczenie

W pracy porównano znane z literatury metody zwartej reprezentacji przestrzeni stanów dla rekonfigurowanego sterownika logicznego. Przedstawiono zalety heurystycznego sposobu kodowania miejsc sieci Petriego, dzięki któremu uzyskuje się diagramy OBDD o znacznie mniejszej złożoności przydatne zarówno podczas analizy, jak i syntezy układowej algorytmu sterowania binarnego.

Słowa kluczowe: Binarne diagramy decyzyjne OBDD, rekonfigurowane sterowniki logiczne, kodowanie stanów, synteza logiczna, weryfikacja formalna.

Reconfigurable Logic Controller state space representation using encoded Binary Decision Diagrams

Abstract

In the paper some known methods for an effective representation of the state space in reconfigurable logic controller are compared. The advantages of heuristic method of Petri net place encoding, which is adapted for a compact encoding technique of Binary Decision Diagrams, are given.

Keywords: Ordered Binary Decision Diagram, Reconfigurable Logic Controller, state encoding, Logic synthesis, formal verification.

1. Wstęp

Program funkcjonowania współbieżnego sterownika logicznego może być opisany za pomocą sieci Petriego lub sekwencyjnych diagramów funkcyjnych – sieci SFC (Sequential Function Chart) [2]. Aby określić własności: bezpieczeństwo, żywotność i ograniczoność tych sieci celowe jest wyznaczenie grafu znakowań osiągalnych [6]. Niestety wraz ze wzrostem złożoności modelowanego systemu współbieżnego silnie wzrasta liczba możliwych stanów globalnych – następuje eksplozja przestrzeni stanów.

Aby zapewnić efektywne przechowywanie w pamięci struktury grafu znakowań osiągalnych, stosuje się symboliczne metody kodowania i prezentacji przestrzeni stanów [10]. Złożone wyrażenia symboliczne w logice zdań reprezentowane są binarnymi diagramami decyzyjnymi BDD. Liczba zmiennych silnie wpływa na rozmiary tego grafu, zwłaszcza w przypadku, gdy każdemu miejscu sieci przypisuje się pojedynczy symbol kodujący. Równocześnie liczba zmiennych kodujących, przypisanych stanom automatu jest równa liczbie przerzutników w rejestrze stanów wewnętrznych, wpływając na koszt implementacji [3].

W artykule, podobnie jak w pracach [10, 11] wykorzystano możliwość przedstawienia przestrzeni stanów sieci Petriego lub diagramów SFC za pomocą kodowanych diagramów decyzyjnych. Prowadzone badania mają na celu zmniejszenie rozmiarów grafu decyzyjnego oraz czasu jego generowania i odtwarzania w procesie

formalnej analizy lub syntezy. W pracy przedstawiono zalety wykorzystania heurystycznego algorytmu kodowania miejsc sieci [1] do zminimalizowania rozmiarów drzewa BDD.

2. Binarne diagramy decyzyjne

Binarny diagram decyzyjny BDD to skierowany i acykliczny graf z wyróżnionym węzłem, będącym korzeniem diagramu. Diagram posiada dwa typy węzłów: węzły nieterminalowe, reprezentujące zmienne funkcji boolowskiej oraz węzły terminalowe, o etykietach 0 i 1, reprezentujące wartości tej funkcji [9].

Umownie przyjęto, że łuki przerywane łączące węzeł z jego następnikiem odpowiadają wartości zerowej zmiennej decyzyjnej węzła, zaś łuki łączące węzeł z prawym jego następnikiem – wartości jeden zmiennej decyzyjnej węzła. W stosowanym w pracy diagramie BDD, przedstawiającą funkcje boolowską $f(x_1..x_i..x_n)$, stosuje się rozkład Shannona:

$$f(x_1, x_2, \dots, x_n) = x_i * f_{x_i} + \bar{x}_i * f_{\bar{x}_i} \quad (1)$$

Człony składowe funkcji $f(x_1..x_i..x_n)$:

$$f_{\bar{x}_i} = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \quad (2)$$

$$f_{x_i} = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

są zwane odpowiednio pozytywnym oraz negatywnym dopełnieniem algebraicznym, ze względu na zmienna x_i .

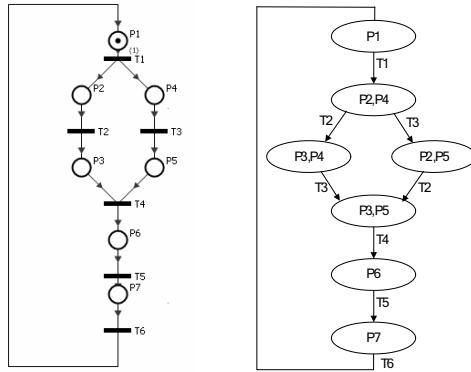
3. Funkcja charakterystyczna przestrzeni stanów sieci

Przestrzeń stanów sieci Petriego (rys.1) może być reprezentowana z wykorzystaniem funkcji charakterystycznej [4, 8, 10] lub sekwentu charakterystycznego [3]. Funkcję taką definiuje się w następujący sposób:

$$X_A(x) = \begin{cases} 1 \Leftrightarrow x \in A, \\ 0 \Leftrightarrow x \notin A. \end{cases} \quad (3)$$

Funkcja charakterystyczna dla sieci z rys.1 jest następująca:

$$\chi_M = p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1' p_2 p_3 p_4 p_5 p_6 p_7' + p_1' p_2 p_3 p_4 p_5 p_6 p_7 + p_1' p_2 p_3 p_4 p_5 p_6 p_7' + p_1' p_2 p_3 p_4 p_5 p_6 p_7 + p_1' p_2 p_3 p_4 p_5 p_6 p_7'$$



Rys. 1. Przykład sieci Petriego oraz graf znakowań osiągalnych
Fig. 1. An example Petri net and the reachability graph

4. Kodowanie heurystyczne miejsc

Heurystyczny algorytm kodowania strukturalnego stanów lokalnych w automatach współbieżnych został rozwinięty dla celów syntezy m.in. w pracach [4, 5, 7]. Celowość kodowania heurystycznego również w reprezentacji przestrzeni stanów za pomocą diagramów BDD uzasadniona jest następującymi właściwościami kodów:

1. Kody miejsc współbieżnych są nieortogonalne.
2. Kody miejsc sekwencyjnych są ortogonalne.
3. Całkowita długość kodu jest niemal minimalna, w porównaniu z ewentualnym, bezpośrednim kodowaniem wierzchołków grafu znakowań.
4. Kodom miejsc odpowiadają zawsze pojedyncze koniunkcje.

Kody uważa się za ortogonalne, gdy iloczyn logiczny (koniunkcja) wyrażeń kodujących jest równy logicznemu 0 (false).

Warto zwrócić uwagę, że w większości znanych z literatury metod kodowania grafów BDD warunki 1, 2 i 4 nie są spełnione. Przykładowo w pracy [10] zaproponowano metodę kodowania, w której część miejsc jest kodowana dysjunkcją dwóch koniunkcji. W realizacji sprzętowej powoduje to znaczne problemy (brak jednoznaczności w odwzorowywaniu każdej z tranzycji sieci w matrycowej strukturze rekonfigurowalnej).

5. Przykłady kodowania metodą heurystyczną

Sieć 1

Przykład kodowania dotyczy sieci z rys. 1, opisującego algorytm sterowania dwoma wózkami zaczerpnięty z pracy [2]. Dla potrzeb kodowania pominięto interpretację sieci. Graf znakowań osiągalnych dla przedstawianej sieci został wyznaczony za pomocą dowolnej z metod znanej z literatury. Funkcja charakterystyczna przedstawia się następująco:

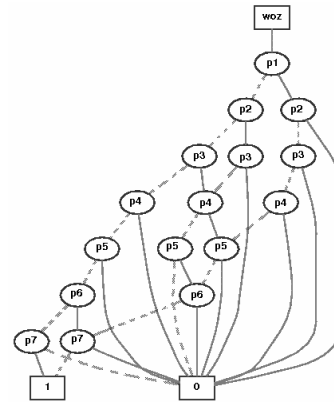
$$\chi_{(M)} = p_1 p_2 p_3 p_4 p_5 p_6 p_7 + p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1 p_2 p_3 p_4 p_5 p_6 p_7' + p_1 p_2 p_3 p_4 p_5 p_6 p_7'$$

Funkcję charakterystyczną przestrzeni stanów sieci z rys. 1 przedstawiono za pomocą diagramu BDD. Diagram (rys.2) został wygenerowany za wykorzystaniem oprogramowania *DDCalc* [12].

Można zauważyć, że głębokość drzewa jest równa liczbie miejsc sieci (7 miejsc). Zaletą diagramu jest możliwość bezpośredniego odczytania poszczególnych stanów globalnych na podstawie symboli stanów lokalnych umieszczonych na jego ścieżkach. Wadą jest silny wzrost rozmiarów diagramu wraz ze wzrostem liczby miejsc. Na podstawie diagramu nie można bezpośrednio odczytać relacji współbieżności lub sekwencyjności (nie-współbieżności) pomiędzy poszczególnymi miejscami.

Kodując miejsca można ograniczyć liczbę zmiennych wykorzystywanych do przedstawienia każdego miejsca sieci. W znanym kodowaniu miejsc na podstawie wyznaczonych uprzednio podsięci Petriego typu automatowego (PN-StateMachine) [4, 7] kody byłby następujące:

$$\begin{aligned} p_1 &= Q_1' Q_2' Q_3' & p_0 &= Q_4' Q_5' \\ p_2 &= Q_1' Q_2' Q_3 & p_4 &= Q_4' Q_5 \\ p_3 &= Q_1' Q_2 Q_3' & p_5 &= Q_4 Q_5' \\ p_6 &= Q_1 Q_1' Q_3' & & \\ p_7 &= Q_1 Q_2' Q_3 & & \end{aligned}$$



Rys. 2. Diagram OBDD funkcji charakterystycznej sieci z rys. 1
Fig. 2. OBDD Diagram of characteristic function for the net from Fig. 1

Dodatkowe miejsce P_0 wprowadzono w celu przechowywania znacznika (żetonu), gdy nie są oznakowane miejsca P_4 i P_5 . Rezultaty kodowania wpływają na złożoność grafu BDD z rys. 3a.

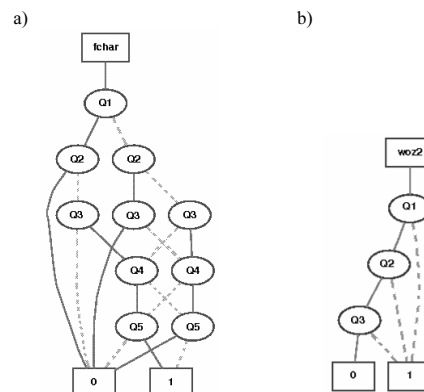
Kodując metodą heurystyczną i wykorzystując program [5] uzyskano następujące kody miejsc:

$$\begin{aligned} p_1 &= Q_1 Q_2 Q_3' & p_5 &= Q_1' Q_3 \\ p_2 &= Q_1' Q_2' & p_6 &= Q_1 Q_2' Q_3' \\ p_3 &= Q_1' Q_2 & p_7 &= Q_1 Q_2' Q_3 \\ p_4 &= Q_1' Q_3' & & \end{aligned}$$

Po podstawieniu kodów miejsc do funkcji charakterystycznej otrzymano następujący rezultat:

$$\chi_{(M)} = Q_1' + Q_2' + Q_3'$$

Diagram OBDD dla powyższej funkcji charakterystycznej po kodowaniu został przedstawiony na Rys. 3b.



Rys. 3. Diagram OBDD sieci z rys. 1 po kodowaniu automatowym (a) po kodowaniu heurystycznym (b)

Fig. 3. OBDD Diagram for the net from Fig. 1 after automata coding (a) after heuristic coding (b)

Możliwe jest podanie kodów dla każdej konfiguracji miejsc równocześnie oznakowanych (stanu globalnego):

$$\begin{aligned} p_1 &= Q_1 Q_2 Q_3' & p_2 p_5 &= Q_1' Q_2' Q_3 \\ p_2 p_4 &= Q_1' Q_2' Q_3' & p_6 &= Q_1 Q_2' Q_3' \\ p_3 p_4 &= Q_1' Q_2 Q_3' & p_7 &= Q_1 Q_2' Q_3 \\ p_3 p_5 &= Q_1' Q_2 Q_3 & & \end{aligned}$$

Rozłożenie poszczególnych kodów stanów globalnych w przestrzeni binarnej zilustrowano poglądowno za pomocą tablicy

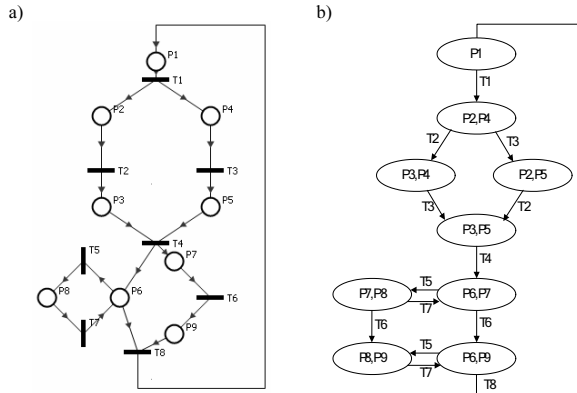
Karnaugh'a (rys. 4). Tylko wektory $[Q_1, Q_2, Q_3]$ zgodne (nieortogonalne) w stosunku do $\chi_{(M)}$ dopuszczalne są podczas pracy układu.

$Q_1 \backslash Q_2 Q_3$	00	01	11	10
0	$p_2 p_4$	$p_2 p_5$	$p_3 p_5$	$p_3 p_4$
1	p_6	p_7		p_1

Rys. 4. Tablica Karnaugh'a sieci z rys. 1
Fig. 4. Karnaugh table for the net from Fig. 1

Sieć 2

Kolejny przykład został zaczerpnięty z pracy [2, 3]. Na rys. 5 przedstawiona została sieć oraz graf znakowań osiągalnych.

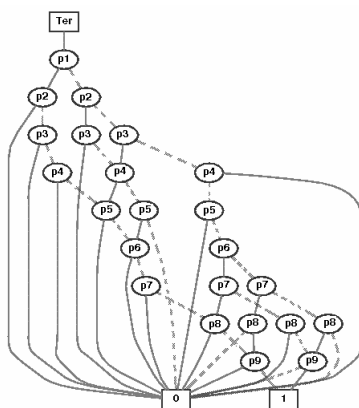


Rys. 5. Przykład sieci Petriego (a) oraz graf znakowań osiągalnych (b)
Fig. 5. An example Petri net (a) and the reachability graph (b)

Funkcja charakterystyczna

$$\chi_{(M)} = p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 + p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9$$

Funkcję charakterystyczną prezentowanej sieci przedstawiono za pomocą diagramu OBDD (rys. 6).



Rys. 6. Diagram OBDD funkcji charakterystycznej sieci z rys.5
Fig. 6. OBDD diagram of characteristic function from Fig. 5

Diagram ten jest dość złożony. Przy pomocy kodowania miejsc można ograniczyć liczbę zmiennych wykorzystywanych do przedstawienia każdego miejsca sieci. Wynik kodowania heurystycznego przedstawiono poniżej:

$$\begin{aligned} p_1 &= Q_1 Q_2' & p_6 &= Q_1 Q_2 Q_3' \\ p_2 &= Q_1' Q_2' & p_8 &= Q_1 Q_2 Q_3 \\ p_3 &= Q_1' Q_2 & p_7 &= Q_1 Q_2 Q_4' \\ p_4 &= Q_1' Q_3' & p_9 &= Q_1 Q_2 Q_4 \\ p_5 &= Q_1' Q_3 & & \end{aligned}$$

Poszczególnym stanom globalnym (wierzchołkom grafu znakowań) odpowiadają następujące koniunkcje kodowań:

$$\begin{aligned} p_1 &= Q_1 Q_2' & p_7 p_8 &= Q_1 Q_2 Q_3 Q_4' \\ p_2 p_4 &= Q_1' Q_2' Q_3' & p_8 p_9 &= Q_1 Q_2 Q_4 \\ p_3 p_4 &= Q_1' Q_2 Q_3' & p_6 p_9 &= Q_1 Q_2 Q_3' Q_4 \\ p_3 p_5 &= Q_1' Q_2 Q_3 & p_2 p_5 &= Q_1' Q_2' Q_3 \\ p_6 p_7 &= Q_1 Q_2 Q_3' Q_4' & & \end{aligned}$$

Po podstawieniu kodów miejsc do funkcji charakterystycznej otrzymano:

$$\chi_{(M)} = 1$$

Wynika stąd, że każdy wektor $[Q_1, Q_2, Q_3, Q_4]$ zapisany w rejestrze stanów określa prawidłowo oznakowanie sieci. Przykładowo wektor $Q_1 Q_2 Q_3' Q_4'$ będzie rozpoznany jako konfiguracja $p_6 p_7$. Rozłożenie kodów stanów globalnych w przestrzeni binarnej zilustrowano poglądowo za pomocą tablicy Karnaugh'a (rys. 7).

$Q_1 Q_2 \backslash Q_3 Q_4$	00	01	11	10
00	$p_2 p_4$	$p_2 p_4$	$p_2 p_5$	$p_2 p_5$
01	$p_3 p_4$	$p_3 p_4$	$p_3 p_5$	$p_3 p_5$
11	$p_6 p_7$	$p_6 p_9$	$p_8 p_9$	$p_7 p_8$
10	p_1	p_1	p_1	p_1

Rys. 7. Tablica Karnaugh'a sieci z rys. 5
Fig. 7. Karnaugh table for the net from Fig. 5

6. Wnioski

Kodowanie diagramu BDD prowadzi do znacznego zmniejszenia jego rozmiaru. Proponowane kodowanie jest przydatne zarówno do celów analizy jak i syntezy [8].

Grafy BDD kodowane metodą heurystyczną mają na ogół niższe rozmiary i gęstość w porównaniu z grafami kodowanymi metodą one-hot i kodowaniem automatowym.

7. Literatura

- [1] Adamski M.: Heurystyczna metoda strukturalnego kodowania miejsc sieci Petriego, Zeszyty Naukowe WSI, Nr 78, Zielona Góra, 1986, s. 113-125
- [2] Adamski M., Chodań M.: Modelowanie układów sterowania dyskretnego z wykorzystaniem sieci SFC, Wydaw. PZ, Zielona Góra, 2000
- [3] Adamski M., Karatkevich A., Węgrzyn M. (Red.): Design of embedded control systems, Springer, New York, 2005
- [4] Bilinski K., Adamski M., Saul J. M., Dagless E. L.: Petri-net-based algorithms for parallel-controller synthesis, IEE Proceedings - Computers and Digital Techniques - 1994, Vol. 141, no 6, s. 405-412
- [5] Bubacz P., Adamski M.: Heuristic algorithm for an effective state encoding for reconfigurable matrix-based logic controller design, PDES 2006 : proceedings of IFAC workshop. Brno, Czechy, 2006, s. 236-241
- [6] David R., Alla H.: Petri Nets and Grafset, Prentice Hall Int., USA, 1992
- [7] Kozłowski T., Dagless E. L., Saul J. M., Adamski M., Szajna J.: Parallel controller synthesis using Petri nets, IEE Proceedings - Computers and Digital Techniques - 1995, Vol. 142, no 4, s. 263-271
- [8] Miczulski P.: Reprezentacja hierarchicznego grafu znakowań z wykorzystaniem funkcji monotonicznych, Informatyka - sztuka czy rzemiosło - KNWS'05, Oficyna wydawnicza UZ, 2005, s. 73-78
- [9] Minato S.: Binary decision Diagrams and application for VLSI CAD, Kluwer Academic Publishers, Boston 1996
- [10] Pastor E., Cortadella J., Roig O. Symbolic Analysis of Bounded Petri Nets. IEEE Transactions on Computers, Vol. 50, No. 5, May 2001, pp. 432-448.
- [11] Carmona J., Colom J., Cortadella J., García-Vallés F.: Synthesis of asynchronous controllers using integer linear programming. IEEE Transactions on Computer-Aided Design, 25(9), 2006, s. 1637-1651
- [12] Strona program DDCalc - <http://vlsi.colorado.edu/~fabio/>