

REPREZENTACJA HIERARCHICZNEGO GRAFU ZNAKOWAŃ Z WYKORZYSTANIEM FUNKCJI MONOTONICZNYCH

Piotr Miczulski

Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50
e-mail: P.Miczulski@iie.uz.zgora.pl

STRESZCZENIE

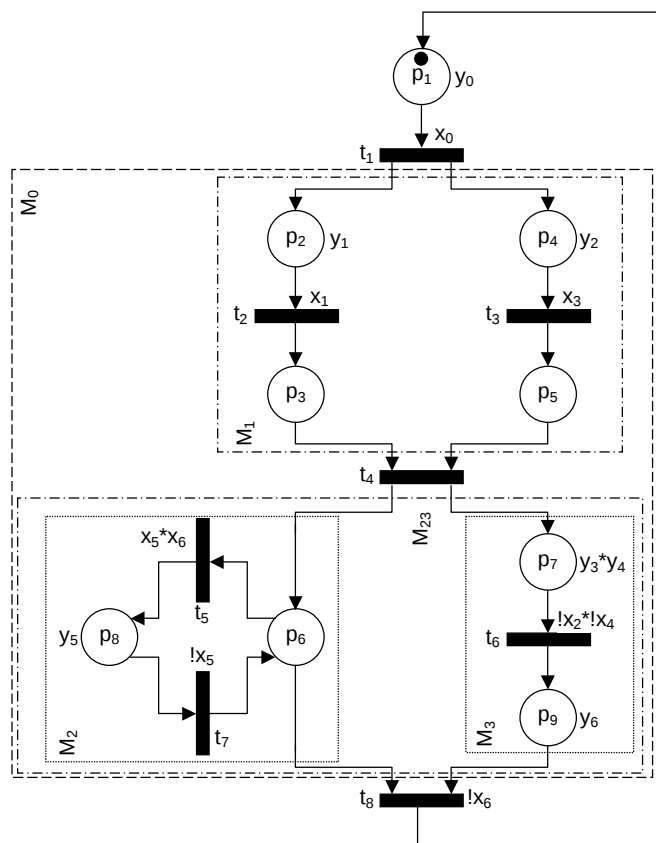
W dotychczas stosowanych metodach opisu przestrzeni stanów automatów współbieżnych, opisywanych jednopoziomowymi lub hierarchicznymi sieciami Petriego, graf znakowań reprezentowany był w postaci klasycznej funkcji charakterystycznej. W opracowanych rozwiązaniach funkcja charakterystyczna przedstawiana była jako pojedynczy diagram decyzyjny, co skutkowało brakiem możliwości efektywnego projektowania złożonych automatów współbieżnych. W referacie zaproponowano nowy sposób opisu przestrzeni stanów, w którym wykorzystano funkcje monotoniczne reprezentowane za pomocą połączonego systemu diagramów BDD.

1. WPROWADZENIE

W trakcie prowadzonych badań nad metodami analizy i syntezy współbieżnych automatów cyfrowych, modelowanych za pomocą jednopoziomowych i hierarchicznych sieci Petriego, pojawiła się konieczność odzwierciedlenia hierarchicznej budowy układu także na poziomie reprezentacji przestrzeni stanów. Wynikało to z konieczności zdecydowanego poprawienia sposobu jej opisu tak, aby umożliwić efektywne projektowanie złożonych automatów współbieżnych. Wykonane badania wykazały, że wykorzystanie w tym celu funkcji monotonicznych zwiększa wydajność reprezentacji przestrzeni stanów układu, a przy okazji daje dodatkowe zalety w postaci nowego wariantu metody dekompozycji automatów współbieżnych na procesy sekwencyjne. Rezultaty dekompozycji mogą zostać wykorzystane następnie do bezpośredniego kodowania stanów lokalnych układu, na potrzeby syntezy, jak i weryfikacji.

Podział grafu znakowań reprezentującego całą przestrzeń stanów na zbiór wzajemnie powiązanych grafów znakowań, opisujących znakowania poszczególnych poziomów hierarchii, łatwość z jaką można „rozwijać” i „zwijać” opis przestrzeni stanów z wykorzystaniem funkcji monotonicznych oraz ich reprezentacja za pomocą diagramów decyzyjnych BDD [4], stwarza możliwość efektywnej analizy i syntezy sterowników

cyfrowych. Przedstawioną metodę reprezentacji przestrzeni stanów omówiono na przykładzie hierarchicznej sieci Petriego, która została przedstawiona na rys. 1.



Rys. 1. Przykład hierarchicznej sieci Petriego

2. FUNKCJE MONOTONICZNE I ICH REPREZENTACJA W POSTACI DIAGRAMÓW BDD

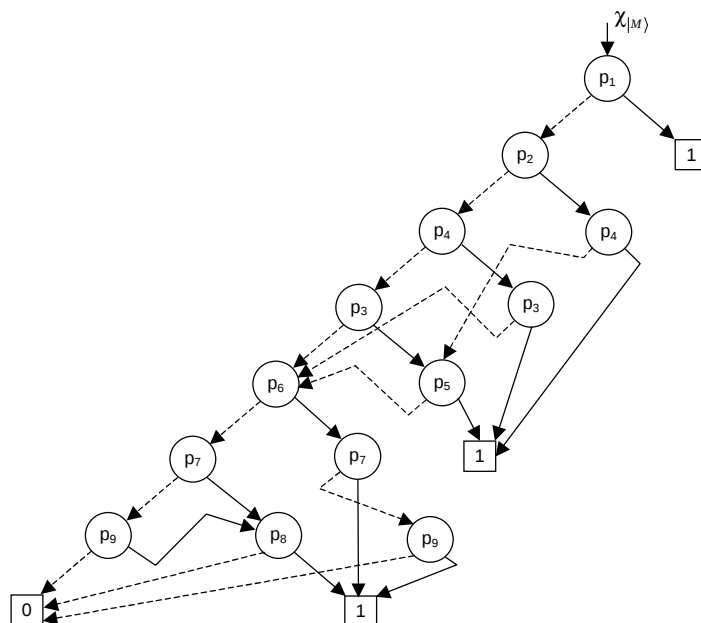
Funkcja boolowska f jest funkcją monotoniczną, jeżeli dla dowolnych wektorów boolowskich a i b takich, że $a \geq b$, funkcja f spełnia warunek $f(a) \geq f(b)$. Kombinacja zmiennych logicznych (wektor boolowski) jest nie mniejsza od innej kombinacji zmiennych, jeżeli wartość każdego argumentu jednej kombinacji jest większa lub równa wartości tego samego argumentu drugiej kombinacji. Kombinacje spełniające ten warunek nazywa się kombinacjami porównywalnymi. Zatem, w funkcji monotonicznej dla dowolnego wzrostu jej argumentów (wektorów boolowskich), wartości funkcji nie maleją, a warunek $f(a) \geq f(b)$ jest spełniony jedynie dla kombinacji porównywalnych. Funkcję monotoniczną można zatem zapisać w dysjunkcyjnej lub koniunkcyjnej postaci normalnej, wykorzystując w tym celu operatory sumy logicznej i iloczynu logicznego oraz nie używając operatora negacji.

Funkcja monotoniczna może być również reprezentowana jako diagram decyzyjny BDD, ZBDD lub inny. Na obecnym etapie prac skupiono się jednak na opracowaniu metod reprezentacji przestrzeni stanów za pomocą funkcji monotonicznych oraz analizie i syntezie tak opisanych układów. Wybór odpowiedniego rodzaju diagramu decyzyjnego zostanie przeprowadzony, w późniejszym czasie po wykonaniu koniecznych badań. Aktualnie w prowadzonych pracach zdecydowano się na wykorzystanie klasycznej odmiany diagramów decyzyjnych, jaką są diagramy ROBDD [4].

Przykładem funkcji monotonicznej jest funkcja opisująca całą przestrzeń stanów sieci Petriego przedstawionej na rys. 1:

$$\chi_{|M\rangle} = p_1 + p_2 p_4 + p_3 p_4 + p_2 p_5 + p_3 p_5 + p_6 p_7 + p_6 p_9 + p_7 p_8 + p_8 p_9 \quad (1)$$

Funkcja (1) może być reprezentowana w postaci diagramu decyzyjnego BDD (rys. 2).



Rys. 2. Diagram BDD reprezentujący funkcję monotoniczną (1)

Umownie przyjęto, że łuk (narysowany linią przerywaną), łączący węzeł z jego lewym następnikiem ma przypisaną wartość „0”, co oznacza, że zmienna skojarzona z węzłem początkowym tego łuku otrzymuje dla niego wartość „0”. Natomiast łuk (narysowany linią ciągłą) łączący węzeł z jego prawym następnikiem ma przypisaną wartość „1”, co oznacza, że zmienna z nim skojarzona otrzymuje wartość „1”. W praktyce diagram BDD zawiera tylko jeden węzeł terminalowy, reprezentujący wartość „1” funkcji boolowskiej, a zerowa wartość funkcji otrzymywana jest przez zanegowanie wartości „1”. Na rys. 2 węzły terminalowe zostały jednak powielone, aby zwiększyć jego czytelność.

Przeglądając diagram rekurencyjnie od korzenia do liści reprezentujących wartość „1” funkcji monotonicznej otrzymuje się mintermy funkcji logicznej. Pomijając na ścieżce węzły reprezentujące zmienne o wartości „0” i stosując regułę pochłaniania otrzymuje się zminimalizowaną funkcję monotoniczną w dysjunkcyjnej postaci normalnej.

3. REPREZENTACJA HIERARCHICZNEGO GRAFU ZNAKOWAŃ Z WYKORZYSTANIEM FUNKCJI MONOTONICZNYCH

Projektując współbieżny sterownik cyfrowy o dużej liczbie stanów lokalnych często nie ma potrzeby rozpatrywania jednocześnie ich wszystkich. Dlatego też abstrahowanie wybranych jego fragmentów, poprzez wprowadzenie w sieci Petriego poziomów hierarchii, w postaci makromiejsc (np. M_1 na rys. 1), pozwala na projektowanie w danej chwili tylko wybranego fragmentu zachowania układu. Podejście to jest dobrze znane z literatury [2], [5], [7].

Wykorzystując hierarchiczne sieci Petriego, na każdym etapie modelowania układu, może zachodzić potrzeba analizy, czy zaprojektowany dotychczas jego fragment, na danym poziomie szczegółowości, nie zawiera konstrukcji wadliwych. Weryfikacja ta może być przeprowadzana poprzez zbadanie podstawowych właściwości sieci Petriego takich, jak żywotność, bezpieczeństwo, czy determinizm. Wymienione właściwości mogą zostać sprawdzone za pomocą analizy przestrzeni stanów [1]. Analiza, jak również synteza, może być przeprowadzana oddzielnie dla każdego makromiejsca [3]. Tak więc celowe jest odzwierciedlenie hierarchicznej budowy układu również na poziomie jego przestrzeni stanów.

W pracy [1] przedstawiono metodę opisu przestrzeni stanów za pomocą klasycznej funkcji charakterystycznej, reprezentowanej w pamięci komputera za pomocą jednego diagramu decyzyjnego, BDD lub ZBDD. Ponieważ jednak utworzona funkcja opisywała całą przestrzeń stanów sterownika cyfrowego, to podejście to jest nie efektywne dla układów o dużej liczbie stanów globalnych. Dla sieci z rys. 1, funkcja charakterystyczna ma następujący wzór:

$$\begin{aligned} \chi_{|M\rangle} = & p_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 + \bar{p}_1 p_2 \bar{p}_3 p_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 + \bar{p}_1 \bar{p}_2 p_3 p_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 + \\ & \bar{p}_1 p_2 \bar{p}_3 \bar{p}_4 p_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 p_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 p_6 p_7 \bar{p}_8 \bar{p}_9 + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 p_9 + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 p_7 p_8 \bar{p}_9 + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 p_8 p_9 \end{aligned} \quad (2)$$

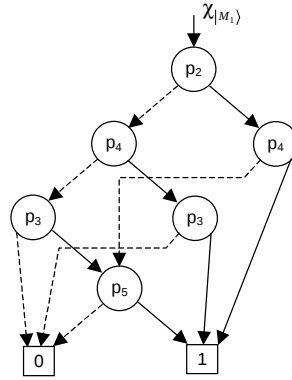
Każde znakowanie sieci Petriego jest opisywane przez składnik funkcji (2), którego poszczególne zmienne odpowiadają miejscom sieci Petriego. Jeżeli w danym składniku zmienna nie jest zanegowana, to oznacza to, że miejsce posiada znacznik. Jeżeli zmienna jest zanegowana, to odpowiadające jej miejsce nie zawiera znacznika.

Z drugie strony prowadzone prace badawcze w zakresie reprezentacji przestrzeni stanów w postaci hierarchicznego grafu znakowań, nie zajmowały się dostatecznie aspektem reprezentacji utworzonego grafu znakowań w pamięci komputera oraz sposobami jego analizy [5].

Autor w swoich pracach proponuje wykorzystanie monotonicznych funkcji boolowskich do opisu przestrzeni stanów układu. Graf znakowań każdego poziomu hierarchii reprezentowany jest za pomocą pojedynczej funkcji monotonicznej w dysjunkcyjnej postaci normalnej. Każdy składnik funkcji przedstawia pojedyncze znakowanie (konfigurację) makromiejsca, a zmienne funkcji odpowiadają miejscom i makromiejscem sieci Petriego. Występowanie nazwy zmiennej w danym składniku funkcji oznacza, że miejsce reprezentowane przez zmienną jest oznakowane w tej konfiguracji. Brak nazwy zmiennej w iloczynie oznacza, że miejsce to nie wchodzi w skład danej konfiguracji. Na przykład, przestrzeń stanów makromiejsca M_1 może zostać wyrażona w postaci następującej funkcji monotonicznej:

$$\chi_{|M_1\rangle} = p_2 p_4 + p_3 p_4 + p_2 p_5 + p_3 p_5 \quad (3)$$

Na rys. 3 przedstawiono diagram BDD reprezentujący funkcję monotoniczną (2).



Rys. 3. Funkcja monotoniczna reprezentująca przestrzeń stanów makromiejsca M_1

Aby odczytać znakowania układu na wybranym poziomie hierarchii należy wyznaczyć wszystkie mintermy proste funkcji monotonicznej. Istotną zaletą funkcji monotonicznych jest fakt, że przy superpozycji monotonicznych funkcji boolowskich i przy podstawianiu do tych funkcji zmiennych, uzyskane w ten sposób funkcje nadal są funkcjami monotonicznymi. Twierdzenie to jest istotne z punktu widzenia „rozwijania” i „zwijania” hierarchicznej reprezentacji przestrzeni stanów. Jeżeli zachodzi potrzeba rozwinięcia opisu przestrzeni stanów w taki sposób, aby funkcja (diagram BDD) przedstawiał całą przestrzeń stanów na wybranym poziomie szczegółowości, należy za odpowiednią zmienną podstawić funkcję monotoniczną reprezentującą niższy poziom hierarchii. Operacja ta może zostać wykonana za pomocą odpowiedniego przekształcenia diagramu decyzyjnego. Na przykład, jeżeli przestrzeń stanów najwyższego poziomu hierarchii opisywana jest funkcją $\chi_{|M\rangle} = p_1 + M_0$, to podstawiając za zmienną M_0 , funkcję reprezentującą przestrzeń stanów makromiejsca M_0 ($\chi_{|M_0\rangle} = M_1 + M_{23}$), otrzymuje się funkcję opisującą zbiór konfiguracji grafu znakowań na

wybrany poziomie abstrakcji: $\chi_{|M\rangle} = p_1 + M_1 + M_{23}$. Postępując w ten sposób można uzyskać całą przestrzeń stanów opisaną wzorem (1).

Ponieważ funkcja przedstawiająca w opisany sposób graf znakowań powinna wykazywać cechy funkcji monotonicznej, to możliwe jest wykrycie w nim niepoprawnych znakowań. Ten wstępny etap weryfikacji przestrzeni stanów może zostać przeprowadzony przed wykonaniem analizy podstawowych właściwości sieci Petriego. Algorytm badania monotoniczności funkcji, bazujący na analizie diagramu BDD, jest tematem aktualnie prowadzonych badań.

4. ZAKOŃCZENIE

Zaletą wykorzystania funkcji monotonicznych, do reprezentacji przestrzeni stanów, w postaci systemu skorelowanych ze sobą diagramów decyzyjnych jest zmniejszenie (od 20% do 30%) rozmiaru diagramu BDD. Odzwierciedlenie hierarchicznej struktury układu na poziomie przestrzeni stanów nie wymaga również operowania na jednym dużym diagramie decyzyjnym w czasie analizy lub syntezy układu. Ponadto zastosowanie funkcji monotonicznych stworzyło możliwość opracowania nowego wariantu metody dekompozycji współbieżnych automatów cyfrowych na procesy sekwencyjne. Przedstawione prace prowadzone są w ramach grantu KBN, 4T11C 006 24.

LITERATURA

- [1] K. Biliński: *Application of Petri Nets in parallel controller design*, PhD. Thesis, University of Bristol, Electrical and Electronic Department, 1996
- [2] P. Buchholz: *Hierarchical High Level Petri Nets for Complex System Analysis*, Proceedings of the 15th International Conference on Application and Theory of Petri Nets, ss. 119-138, 1994
- [3] P. Miczulski: *Weryfikacja poprawności opisu współbieżnych sterowników cyfrowych z wykorzystaniem diagramów decyzyjnych*, Materiały Konferencji RUC, 2004
- [4] S. Minato: *Binary Decision Diagrams and Applications for VLSI CAD*, Kluwer Academic Publishers, 1996
- [5] M. Notomi, T. Murata: *Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent Software Analysis*, Proceedings of IEEE Transactions on Software Engineering, Vol. 20, No 5, 1994
- [6] E. Pastor, O. Roig, J. Cortadella, M. Badia: *Petri Net Analysis Using Boolean Manipulation*, Proceedings of 15th International Conference, Application and Theory of Petri Nets, volume 815 of Lecture Notes in Computer Science, 1994
- [7] M. Węgrzyn: *Hierarchiczna implementacja współbieżnych kontrolerów cyfrowych z wykorzystaniem FPGA*, Rozprawa Doktorska, Warszawa, 1998