

WYKRYWANIE BLOKAD W SYSTEMACH PRIORYTETOWYCH

Andrzej Karatkiewicz

**Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50**

e-mail: a.karatkiewicz@iie.uz.zgora.pl

STRESZCZENIE

W artykule przedstawiono metodę wykrywania blokad we współbieżnych systemach priorytetowych, opisanych modelem sieci Petriego z priorytetami statycznymi. Metoda ta bazuje na zmodyfikowanym algorytmie „upartych zbiorów” wykrywania blokad w klasycznych sieciach Petriego.

1. WPROWADZENIE

Większość współczesnych systemów cyfrowych może być traktowana jako współbieżne. Dotyczy to zarówno części sprzętowej, jak i algorytmów, wykonywanych przez te systemy. Dlatego wiele języków opisu sprzętu (np. VHDL, Verilog) pozwala na modelowanie współbieżnych procesów dyskretnych. Powstał też szereg matematycznych modeli opisu systemów współbieżnych, takich jak np. sieci Petriego lub Statecharts.

Metody analizy systemów współbieżnych mają duże znaczenie praktyczne [1]. Analiza i weryfikacja takich systemów, w tym wykrywanie blokad, jest problemem skomplikowanym, co spowodowane jest wykładniczą zależnością pomiędzy ilością osiągalnych stanów globalnych a ilością stanów lokalnych.. Jedną ze znanych metod takiej analizy jest metoda „upartych zbiorów” [2], pozwalająca poprzez badanie części przestrzeni stanów, czasami dużo mniejszej niż całość, wyznaczać osiągalne blokady systemu oraz (z pewnymi ograniczeniami) analizować niektóre inne własności. Metoda ta została opracowana jednak dla klasycznego modelu sieci Petriego i nie uwzględnia uwarunkowań priorytetowych (istotnych np. dla diagramów Statechart [3]). Istnieją prace, opisujące przekształcenie ograniczonej sieci Petriego z priorytetami (to znaczy takiej, że na zbiorze tranzycji zadana jest relacja priorytetów) na klasyczną sieć Petriego z zachowaniem ekwiwalentnym [4]. W proponowanym artykule opisano algorytm badania własności współbieżnych, dyskretnych systemów priorytetowych, opracowany z wykorzystaniem metody przekształceń sieci do postaci klasycznej oraz algorytmu „upartych zbiorów”.

2. INFORMACJE PODSTAWOWE

2.1. Sieci Petriego i priorytety

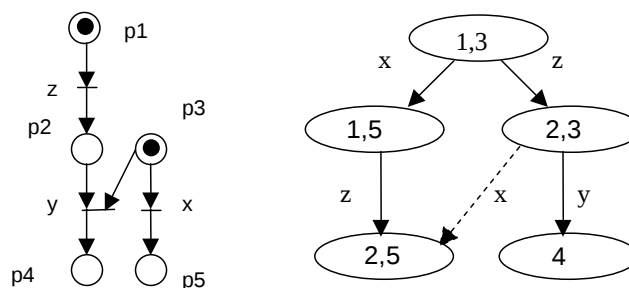
Sieci Petriego [5] są popularnym modelem matematycznym i graficznym, służącym do opisu systemów współbieżnych. Sieć Petriego może być przedstawiona jako dwudzielny graf skierowany z dwoma rodzajami wierzchołków, które nazywane są *miejscami* i *tranzycjami*. Stan sieci Petriego, nazywany *znakowaniem*, wyznaczany jest poprzez *znaczniki*, które mogą znajdować się w miejscach sieci. Jeśli znaczniki znajdują się we wszystkich wejściowych miejscach tranzycji, to taka tranzycja jest *aktywna* i może zostać *zrealizowana*. Realizacja tranzycji usuwa jeden znacznik z każdego z miejsc wejściowych i dodaje jeden znacznik do każdego z miejsc wyjściowych tranzycji.

Tak opisana sieć jest *siecią zwykłą*; czasem jednakże zachodzi potrzeba rozpatrzenia *sieci z łukami ważonymi* [5]. W takich sieciach do łuków są przypisane wagi całkowite, a warunkiem aktywności tranzycji jest obecność w jej każdym miejscu wejściowym ilości znaczników nie mniejszej, niż waga odpowiedniego łuku. Ilość znaczników, usuwanych i dodawanych do miejsc sieci podczas realizacji tranzycji, wyznacza się wagami odpowiednich łuków. Sieć zwykłą można traktować jako sieć, której wagi wszystkich łuków są równe 1.

Sieć jest *bezpieczną*, jeśli w każdym z osiągalnych znakowań żadne miejsce sieci nie zawiera więcej niż jeden znacznik. Sieć jest *ograniczoną*, jeśli istnieje granica górna ilości znaczników w każdym miejscu sieci dla wszystkich osiągalnych znakowań.

Siecią z priorytetami (statycznymi) [4] nazywana jest sieć Petriego taka, że na jej zbiorze tranzycji T jest zdefiniowana *relacja priorytetowa* ρ ; a $(t,u) \in \rho$ oznacza, że tranzycja t ma niższy priorytet od tranzycji u . Jeśli w znakowaniu M tranzycja t jest aktywna, to może zostać zrealizowana tylko wtedy, gdy żadna tranzycja o wyższym priorytecie nie jest aktywna.

Nałożenie na sieć priorytetów wyklucza niektóre jej ewolucje. Na rys. 1 pokazany jest przykład sieci (wzięty z [4]) oraz jej graf osiągalności.



Rys. 1. Przykład sieci Petriego i jej grafu osiągalności. Priorytet $\rho = \{(x,y)\}$ uniemożliwia zmianę znakowania, zaznaczoną przerywaną strzałką

2.2. Metoda „upartych zbiorów”

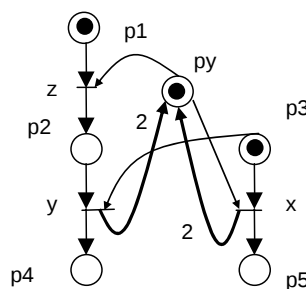
„Upartym zbiorem” (*stubborn set*) dla znakowania M nazywa się zbiór tranzycji T_S , który spełnia trzy warunki: (1) każda nieaktywna tranzycja w T_S ma puste miejsce wejściowe p takie, że wszystkie tranzycje, dla których p jest miejscem wyjściowym, należą do T_S ; (2) żadna aktywna tranzycja w T_S nie ma wspólnego miejsca wejściowego z tranzycją poza T_S ; (3) T_S zawiera aktywną tranzycję. Metoda polega na tym, że w każdym danym znakowaniu symuluje się realizację aktywnych tranzycji należących tylko do T_S , budując w ten sposób częściowy (zredukowany) graf osiągalności, pozwalający na określenie niektórych własności sieci. W swojej klasycznej wersji dla dowolnej sieci Petriego graf zredukowany zawiera wszystkie blokady sieci [2].

2.3. Konstruowanie sieci Petriego, odpowiadającej systemowi priorytetowemu

W [4] opisana została metoda, pozwalająca przekształcić sieć Petriego z priorytetami (ograniczoną) na klasyczną (nie priorytetową), ekwiwalentną behawioralnie. Ekwiwalencja ta w zasadzie nie jest pełna, ale zostaje zachowany szereg istotnych właściwości funkcjonalnych, również blokady.

Metoda polega na tym, że dla każdej tranzycji y , takiej że $\exists x: (x,y) \in p$, wprowadza się dodatkowe miejsce p_y , które jest wejściowym i wyjściowym dla tranzycji x i jest połączone z innymi tranzycjami sieci w taki sposób, że suma znaczników w p_y i miejscach wejściowych tranzycji y jest stała. Miejsce p_y jest puste wtedy i tylko wtedy, gdy tranzycja y jest aktywna, co uniemożliwia realizację tranzycji x . Niestety spełnienie warunków bezpieczeństwa sieci zwykłej z priorytetami, niekoniecznie pociąga ich spełnienie w sieci bez priorytetów.

Z powodu braku miejsca nie opisano tej metody szczegółowo. Na rys. 2 pokazany jest przykład jej zastosowania – sieć Petriego, odpowiadająca sieci z priorytetami (rys. 1) [4].



Rys. 2. Przykład zastosowania metody Besta-Koutnego (dla sieci z rys. 1). Pogrubione łuki mają przypisane wagi. Miejsce p_y jest miejscem dodatkowym.

Łatwo zauważyć, że graf osiągalności został taki sam, jak dla sieci z priorytetami z rys. 1; różnica jednak polega na tym, że tranzycje x i z , współbieżne dla sieci na rys. 1, znajdują się w konflikcie na rys. 2.

3. WYKRYWANIE BLOKAD W SIECIACH PETRIEGO Z PRIORYTETAMI

3.1. Metoda

Wszystkie blokady w sieci priorytetowej, jak w każdym systemie dyskretnym, mogą być wykryte przy pomocy konstruowania pełnej przestrzeni stanów. Jednak przy próbie zastosowania do sieci Petriego z priorytetami metody upartych zbiorów, skutecznie redukującej ilość badanych stanów, to okaże się, że można nie wykryć wszystkich blokad.

Wracając do przykładu z rys. 1; dla znakowania początkowego zbiorem upartym minimalnym (nie zawartym w innych) jest zbiór $\{z\}$. Ograniczając się do symulacji realizacji z , traci się możliwość wykrycia blokady $\{2,5\}$. Jest to wynikiem faktu, że realizacja z w stanie początkowym powoduje aktywność tranzycji y , po czym realizacja x staje się niemożliwa.

Może się zdarzyć nawet taka sytuacja, że w zbiorze tranzycji priorytetowej sieci Petriego, odpowiadającym definicji upartego zbioru, żadna tranzycja nie będzie mogła zostać zrealizowana, bo realizacji aktywnych tranzycji należących do zbioru, będzie przeszkadzać aktywna tranzycja o większym priorytecie spoza zbioru.

Wydaje się intuicyjnie oczywistym, że metoda upartych zbiorów może być stosowana dla systemów priorytetowych, bez rzeczywistego wykonania przekształcenia sieci do postaci klasycznej, ale definicja upartego zbioru wymaga w tym przypadku korekty.

Definicja. „Upartym zbiorem” dla znakowania M sieci priorytetowej nazywa się zbiór tranzycji T_S , który spełnia sześć warunków: (1) każda nieaktywna tranzycja w T_S ma puste miejsce wejściowe p takie, że wszystkie tranzycje, dla których p jest miejscem wyjściowym, należą do T_S ; (2) żadna aktywna tranzycja w T_S nie ma wspólnego miejsca wejściowego z tranzycją poza T_S ; (3) T_S zawiera aktywną tranzycję; (4) jeśli aktywna tranzycja w T_S nie może zostać zrealizowana dlatego, że są aktywne tranzycje o większym priorytecie, jedna z takich tranzycji należy do T_S ; (5) jeśli aktywna tranzycja w T_S ma miejsce wyjściowe, które jest miejscem wejściowym dla tranzycji y takiej, że $(x,y) \in \rho$, to $x \in T_S$; (6) jeśli dla aktywnej tranzycji $x \in T_S$ $(x,y) \in \rho$, to każda tranzycja, której miejsce wyjściowe jest miejscem wejściowym dla y , należy do T_S .

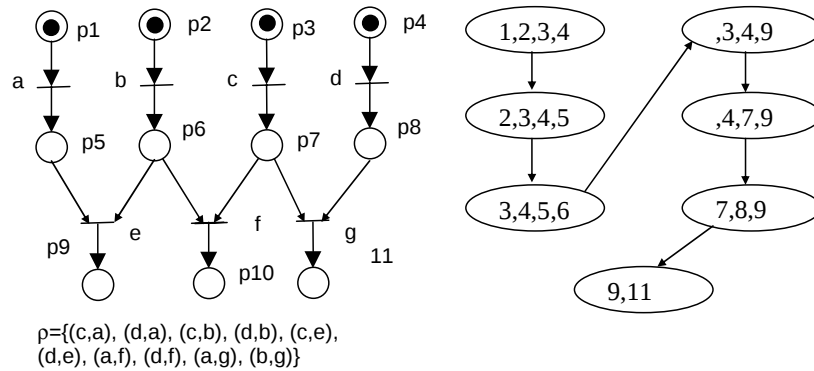
Twierdzenie. Zredukowany graf osiągalności sieci Petriego z priorytetami, skonstruowany w ten sposób, że dla każdego z rozpatrywanych znakowań symulowane są realizacje tylko tych aktywnych tranzycji, które należą do zbioru T_S dla danego znakowania, spełniającego wymienione powyżej sześć warunków, zawiera wszystkie osiągalne blokady sieci.

Poprawność twierdzenia wynika z poprawności metody upartych zbiorów dla klasycznej sieci Petriego (dowód patrz w [6]) i metody Besta-Koutnego przekształcenia sieci priorytetowej na sieć bez priorytetów (dowód patrz w [4]).

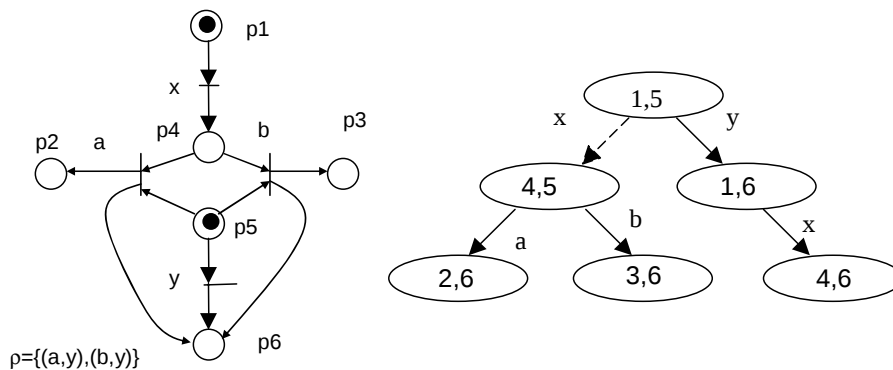
Można jednak zauważyć, że ostatni warunek (6) jest zbyteczny. Chodzi w nim o to, że jeśli realizacja aktywnej tranzycji (niech to będzie z) dodaje znaczniki do miejsc wejściowych tranzycji y takiej, że $(x,y) \in \rho$, może to uniemożliwić realizację tranzycji x . Jeśli x i y nie mają wspólnych miejsc wejściowych, takie „uniemożliwienie” okaże się tylko tymczasowym (do realizacji y), więc nie będzie miało wpływu na osiągalne blokady. Jeśli x i y mają wspólne miejsca wejściowe, to y znajduje się w T_S ze względu na warunek (2). Wobec powyższego warunek (6) można z definicji usunąć.

3.2. Przykłady

W przykładzie z rys. 1 skorygowanej definicji upartego zbioru odpowiada tylko zbiór $\{x,y\}$, wszystkie blokady zostaną więc wykryte. Poniżej podano dwa inne przykłady.



Rys. 3. Przykład zastosowania metody



Rys. 4. Przykład zastosowania metody

Na rys. 3 pokazano sieć priorytetową, która ma o jedną osiągalną blokadę mniej, niż odpowiednia sieć bez priorytetów ($\{5,8,10\}$) i zredukowany graf osiągalności, skonstruowany

za pomocą zaproponowanej metody. Na rys. 4 pokazano sieć priorytetową (zmodyfikowany przykład z [7]), do której stosowanie klasycznej metody upartych zbiorów nie pozwoliłoby wykryć blokad {2,6} oraz {3,6} (nie zostałaby symulowana realizacja tranzycji x w znakowaniu początkowym – strzałka przerywana na grafie). Stosowanie proponowanej metody pozwala wykryć wszystkie blokady.

4. PODSUMOWANIE

Przedstawiona metoda pozwala w sposób oszczędny wobec czasu i pamięci wykrywać blokady we współbieżnych systemach priorytetowych. Metoda jest łatwa do zaimplementowania programowego i może być wykorzystana w systemach komputerowego wspomagania projektowania układów sterowania logicznego, w programowaniu współbieżnym i innych zastosowaniach systemów priorytetowych jako część formalnej analizy i weryfikacji.

Dalsze badania w tym kierunku będą dotyczyły analizy innych behawioralnych właściwości systemów priorytetowych, poprzez konstruowanie zredukowanych przestrzeni osiągalności oraz strukturalnej analizy systemów priorytetowych.

Praca naukowa finansowana ze środków Komitetu Badań Naukowych jako projekt badawczy № 4 T11C 006 24.

LITERATURA

- [1] M. Heiner: *Petri net based system analysis without state explosion*, Proc. High Performance Computing '98, Boston, April 1998, session "Petri net Applications and HPC", pp. 394-403
- [2] A. Valmari: *State of the art report: Stubborn sets*, Petri Nets News., 46, 1994, pp. 6-14
- [3] A. Karatkevich: *Deadlock Analysis in Statecharts*, Proc. Forum on Specification & Design Languages '03, Frankfurt, September 2003, session "Property Specification and Analysis", pp. 414-424
- [4] E. Best, M. Koutny: *Petri net semantics of priority systems*, Theoretical Computer Science, 96, 1992, pp. 175-215
- [5] J. L. Peterson: *Petri net theory and the modeling of systems*, Prentice-Hall, Inc., 1981
- [6] A. Valmari: *The State Explosion Problem*, Lecture Notes on Computer Science, 1491, 1998, pp. 429-528
- [7] E. Best: *Petri net semantics of priorities*, Extended Abstracts of Concurrency and Compositability, S. Miniato, February-March 1990, pp. 11-14