

Jacek TKACZ

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Wykrywanie zastoju w sterownikach logicznych metodą symbolicznego wnioskowania Gentzena

Mgr inż. Jacek TKACZ

Absolwent Uniwersytetu Zielonogórskiego. Od 2001 roku pracownik zatrudniony na stanowisku asystenta w Zakładzie Inżynierii Komputerowej. Prowadzi badania nad symboliczną metodą dowodzenia twierdzeń i jej praktycznymi zastosowaniami w informatyce oraz elektronice. Interesuje się nowoczesnymi technologiami projektowania i wytwarzania aplikacji. W latach 1997-2005 zajmował się projektowaniem profesjonalnych baz danych dla polskich bibliotek (PROLIB).

e-mail: J.Tkacz@iie.uz.zgora.pl



Streszczenie

W referacie przedstawiono koncepcję i sposób weryfikacji specyfikacji sterowników logicznych z wykorzystaniem wnioskowania Gentzena. Zastosowanie symbolicznej metody Gentzena w odróżnieniu od opisywanych w literaturze rozwiązań, pozwala na pominięcie szeregu kroków w procesie analizy symbolicznej sieci Petriego, opisującej funkcjonowanie sterownika, co prowadzi do pełnej automatyzacji procesu weryfikacji. Metoda dokładnie określa fragmenty specyfikacji, w których występują defekty (np. zastoje).

Słowa kluczowe: Gentzen, sekwent, wnioskowanie symboliczne, sterowniki logiczne, sieci Petriego.

Deadlocks detection in logic controllers by means of using Gentzen symbolic deduction

Abstract

The paper presents a concept and methodology for verification of logic controllers specification by means of using Gentzen symbolic deduction. This method omits some steps of algorithm of Petri net symbolic analysis, where Petri net specifies behavior of logic controller. This concept makes possible to automate full process of verification and determines places, for which specification has errors (deadlocks).

Keywords: Gentzen, sequent, symbolic deduction, logic controllers, Petri nets.

1. Wstęp

Ważną grupę układów cyfrowych stanowią układy współbieżne, do których należą sterowniki logiczne. Konstruowanie takich układów, zwłaszcza poprawna integracja i synchronizacja procesów może przysporzyć projektantom wiele kłopotów. Do modelowania układów współbieżnych można zastosować grafy stanów, sieci działań, diagramy przejść itp. Efektywnym sposobem modelowania takich układów są sieci Petriego, przy pomocy których w sposób naturalny można przedstawiać współbieżność. Teoria sieci Petriego stanowi obszerną i szybko rozwijającą się dziedzinę nauki, która odgrywa ważną rolę w projektowaniu systemów informatycznych, a także w planowaniu i sterowaniu przepływem produkcji. Analiza formalna sieci Petriego pozwala wykryć szereg niezamierzonych defektów realizowanego algorytmu sterowania. Badając żywotność sieci można sprawdzić, czy w projektowanym układzie nie występują zagnieżdżenia i zapętlenia. Jeżeli badana sieć nie jest żywa, to może nastąpić częściowe lub całkowite zablokowanie układu.

Większość algorytmów opisywanych w literaturze, służących do badania żywotności sieci daje odpowiedź, czy oczekiwana własność jest spełniona, czy też nie. Zastosowanie w tym przypadku metody analizy bazującej na logice Gentzena [1, 8] i symbolicznym przetwarzaniu danych, umożliwi dodatkowo zlokalizowanie tych fragmentów sieci, w których występują defekty.

2. System Gentzena

W systemie Gentzena zastosowano dziesięć reguł wnioskowania [1] dla następujących spójników logicznych: negacja „~”, dysjunkcja „+”, koniunkcja „*”, implikacja „->” i równoważność „<->”. Dla każdego spójnika zostały podane dwie reguły jego eliminowania. Jedna z nich dotyczy sytuacji, gdy spójnik logiczny znajduje się po lewej stronie sekwentu, a druga, gdy spójnik umieszczony po prawej stronie sekwentu. Wyboru reguły dokonuje się poprzez lokalizację głównego spójnika względem znaku wynikania logicznego „|-”. Reguły stosowane są zawsze do spójnika głównego formuły nieelementarnej w zredukowanym sekwenie. Proces redukcji powtarzany jest tak długo, aż otrzymane zostaną same sekwenty znormalizowane, to jest sekwenty nie zawierające żadnych spójników logicznych.

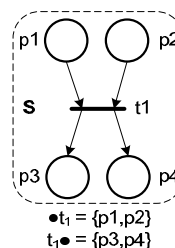
Do systemu Gentzena wprowadzono dodatkowo cztery reguły minimalizacji wyrażeń normalizowanych [1, 8]: tautologii, sklejania sekwentów, pochłaniania oraz consensusu. Minimalizacja w tym przypadku, poza zmniejszeniem liczby nieużytecznych wyników, ma znaczny wpływ na skrócenie przebiegu normalizacji. Odpowiednio wcześnie zlokalizowanie sklei i eliminacja tautologii pozwala na pominięcie analizy sekwentów, które nie mają wpływu na wartość logiczną badanego wyrażenia.

3. Wyznaczanie blokad i pułapek w sieciach Petriego

Na podstawie definicji blokady dla fragmentu sieci przedstawionego na rysunku (rys. 1) otrzymywane jest:

$$\text{jeżeli } p3 \in S, \text{ to } p1 \in S \text{ lub } p2 \in S \text{ oraz} \quad (1)$$

$$\text{jeżeli } p4 \in S, \text{ to } p1 \in S \text{ lub } p2 \in S \quad (2)$$



Rys. 1. Wycinek sieci Petriego

Fig. 1. Part of Petri net

Przedstawiając p_i za pomocą zmiennej logicznej x_i i stosując własność (3):

$$a+b \rightarrow c \equiv (a \rightarrow c) \wedge (b \rightarrow c) \quad (3)$$

zależności (1) i (2) można zapisać w postaci sekwentu:

$$|-(x_3 + x_4) \rightarrow (x_1 + x_2); \quad (4)$$

Po lewej stronie znaku implikacji znajduje się suma wszystkich zmiennych wyjściowych tranzycji t , natomiast po prawej stronie suma wszystkich zmiennych wejściowych:

$$|-\sum_{p_i \in t \bullet} x_i \rightarrow \sum_{p_j \in \bullet t} x_j; \quad (5)$$

Dla całej sieci wyrażenie opisujące wszystkie tranzycje przyjmie postać:

$$|-\prod_{t \in T} (\sum_{p_i \in t \bullet} x_i \rightarrow \sum_{p_j \in \bullet t} x_j); \quad (6)$$

Wszystkie wektory odpowiadające rozwiązaniom powyższego wyrażenia (6) odpowiadają blokadom w sieci Petriego.

W przypadku pułapek, na podstawie definicji pułapki, dla fragmentu sieci przedstawionego na rysunku (rys. 1) otrzymywane jest:

$$\text{jeżeli } p_1 \in Q, \text{ to } p_3 \in Q \text{ lub } p_4 \in Q \text{ oraz} \quad (7)$$

$$\text{jeżeli } p_2 \in Q, \text{ to } p_3 \in Q \text{ lub } p_4 \in Q \quad (8)$$

Przedstawiając p_i za pomocą zmiennej logicznej x_i i stosując własność (9):

$$a+b \rightarrow c \equiv (a \rightarrow c) \wedge (b \rightarrow c) \quad (9)$$

zależności (7) i (8) można zapisać w postaci sekwentu:

$$|(x_1+x_2) \rightarrow (x_3+x_4); \quad (10)$$

Po lewej stronie znaku implikacji znajduje się suma wszystkich zmiennych wejściowych tranzycji t , natomiast po prawej stronie suma wszystkich zmiennych wyjściowych:

$$|-\sum_{p_i \in \bullet t} x_i \rightarrow \sum_{p_j \in t \bullet} x_j; \quad (11)$$

Dla całej sieci wyrażenie opisujące wszystkie tranzycje przyjmie postać:

$$|-\prod_{t \in T} (\sum_{p_i \in \bullet t} x_i \rightarrow \sum_{p_j \in t \bullet} x_j); \quad (12)$$

Wszystkie wektory odpowiadające rozwiązaniom powyższego wyrażenia (12) odpowiadają pułapkom w sieci Petriego.

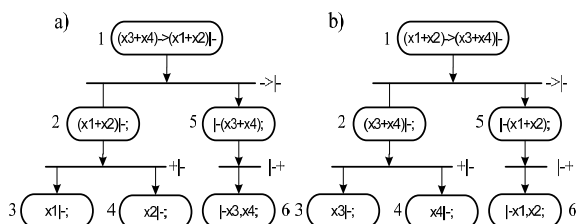
Struktura topologiczna sieci Petriego może być przedstawiona w postaci różnych wyrażeń logicznych, w tym sekwentów [1]. Innym sposobem przedstawionym w literaturze jest opisanie pewnych własności strukturalnych sieci w postaci formuł Horna [3]. Zastosowanie efektywnych algorytmów Thelena-Mathonego [5] do analizy tych własności zostało zaproponowane w pracy [10]. Właściwości systemu Gentzena umożliwiają analizę złożonych wyrażeń opisujących blokady i pułapki, bez konieczności dodatkowej transformacji ich do postaci formuł Horna.

Aby poddać normalizacji Gentzena wyrażenia opisujące blokady i pułapki, należy zapisać je po lewej stronie znaku wynikania logicznego. Po superpozycji lewych stron otrzymanych sekwentów otrzymuje się wyrażenie logiczne w postaci dysjunkcyjnej.

$$(x_3+x_4) \rightarrow (x_1+x_2); \quad (13)$$

$$(x_1+x_2) \rightarrow (x_3+x_4); \quad (14)$$

W sekwentach opisujących blokady i pułapki występują tylko trzy rodzaje spójników logicznych (dysjunkcja „+”, koniunkcja „*”, implikacja „ $\rightarrow$ ”), stąd wystarczy zastosować tylko trzy reguły wnioskowania. Wskazane jest użycie dodatkowych reguł wnioskowania, rozszerzających standardową logikę Gentzena, w celu doprowadzenia zespołu sekwentów do postaci minimalnej. Na rysunku 2 przedstawiono graficzne reprezentacje normalizacji sekwentów (13, 14) według reguł Gentzena, opisujących blokady i pułapki dla przykładowego fragmentu sieci (rys. 1). Zakłada się, że wyrażenia opisujące blokady lub pułapki są fałszywe, a sieć nie zawiera blokad (rys. 2a) lub pułapek (rys. 2b). Proces normalizacji będzie dowodem postawionych założeń.



Rys. 2. Przykład normalizacji sekwentów Gentzena
Fig. 2. Example of Gentzen sequent normalization

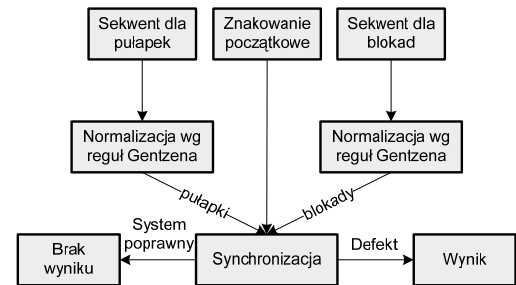
Sekwenty wynikowe będą stanowiły kontrprzykłady zdefiniowanych założeń. Blokady i pułapki w znormalizowanych sekwentach znajdują się po lewej stronie znaku wynikania logicznego (blokady: x_1 i x_2 ; pułapki: x_3 i x_4). Pułapki i blokady minimalne otrzymuje się na podstawie afirmacji zmiennych. Prawe strony sekwentów pomija się w dalszym procesie analizy.

4. Badanie żywotności sieci Petriego

Żywotność sieci można określić jako możliwość utrzymania żywego znakowania, czyli dla każdej tranzycji można z każdego znakowania osiągalnego dojść do takiego znakowania, w którym ta tranzycja będzie mogła zostać zrealizowana.

Algorytm wyznaczania wszystkich blokad i pułapek nadaje się do sprawdzania żywotności sieci Petriego [6, 7]. Aby stwierdzić, czy dana sieć jest żywa, bada się zależności między blokadami i pułapkami. Po wyznaczeniu blokad i pułapek, należy sprawdzić zgodnie z własnością Commonera [4], czy każda wyznaczona blokada zawiera oznakowaną pułapkę. W przypadku pułapek należy sprawdzić, czy każda wyznaczona pułapka zawiera oznakowaną blokadę.

Zastosowanie algorytmu Gentzena automatyzuje proces badania żywotności. Możliwe jest połączenie procesu wyznaczania blokad i pułapek z jednoczesnym badaniem zależności pomiędzy nimi.



Rys. 3. Schemat badania żywotności sieci
Fig. 3. Schema of Petri nets liveness testing

Pełna automatyzacja badania żywotności możliwa jest przez wprowadzenie niewielkich zmian do algorytmu Gentzena. Sekwenty blokad jak i pułapek można normalizować współbieżnie (rys. 3). Otrzymywane z procesów normalizacji sekwentów blokad i pułapek trafiają do procesu synchronizacji, który odrzuca prawe strony sekwentów, sprowadza je do postaci minimalnej poprzez odrzucenie rozwiązań nadmiarowych, sprawdza pokrycie blokad przez pułapki i pułapek przez blokady oraz weryfikuje znakowania początkowe. Pokryte wzajemnie pułapki i blokady zostają odrzucone. Jeżeli z całego procesu pozostaną blokady lub pułapki, to będą one wskazywały dokładnie miejsce defektu sieci Petriego opisującej projektowany system. Brak rozwiązań informuje o poprawnie zaprojektowanym systemie sterowania, w którym zastoje nie występują.

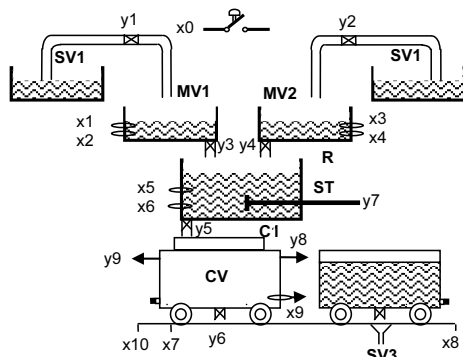
5. Analiza rzeczywistego systemu sterowania

Jako przykład posłuży znany z literatury [2] system sterowania procesem mieszania i transportu cieczy. W celu uwidocznienia problemu diagnostyki sieci zostanie przedstawiony system sterowania ze źle zaprojektowaną siecią (rys. 4), w której należy wskazać miejsca defektów [10].

System rozpoczyna działanie po naciśnięciu przycisku $x0$, co powoduje otwarcie zaworów $y1$ i $y2$. Naplnione zostają zbiorniki $MV1$ i $MV2$ do poziomu kontrolowanego przez czujniki $x1$ i $x3$. Naciśnięcie przycisku startującego proces powoduje również przemieszczenie wózka CV do pozycji początkowej, co sygnalizowane zostanie sygnałem $x10$. Zamierzona modyfikacja funkcjonowania systemu polega na tym, że wózek musi przejechać pozycję $x7$, gdyż specjalnie zmniejszono jego komorę wlewu. Po równoczesnym zamknięciu zaworów $y1$ i $y2$,

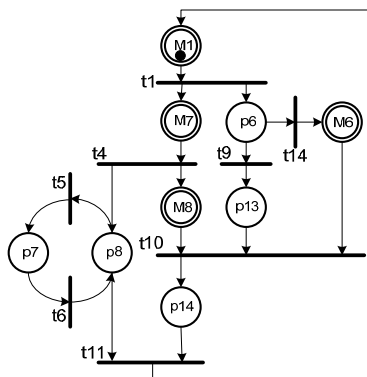
zostają otwarte zawory y_3 i y_4 i reaktor R zostaje napełniony cieczą z dwóch wcześniej napełnionych zbiorników (sygnały x_2 i x_4).

Gdy zbiornik reaktora zostanie napełniony do poziomu czujnika x_5 , uruchomione zostanie mieszadło ST . Opróżnienie zbiorników $MV1$ i $MV2$ powoduje zamknięcie zaworów y_3 i y_4 . Faza napełniania zbiornika CV nastąpi w momencie, gdy wózek $C1$ znajdzie się w lewym skrajnym położeniu, a zawór y_5 zostanie otwarty. Sygnał x_6 informuje o opróżnieniu reaktora. Ciecz zostaje przetransportowana do zbiornika $SV3$ wózkiem $C1$. Cały cykl zamyka się w stanie początkowym, a system czeka na ponowne naciśnięcie przycisku x_0 .



Rys. 4. Zmodyfikowany system mieszania cieczy
Fig. 4. Modified system of liquid mixing

Ze względu na dużą objętościową specyfikację systemu, zbada-na zostanie sieć w postaci zredukowanej - makrosieć (rys. 5). Uproszczenie sieci zostało wykonane zgodnie z opisywanymi w literaturze algorytmami [10], co powoduje zachowanie jej pierwotnych właściwości takich jak żywotność, czy ograniczoność [6]. Dzięki redukcji zmniejsza się rozmiary sekwentów, opisujących blokady i pułapki. Ma to znaczący wpływ na efektywność obliczeń analizowanych wyrażeń i powoduje znaczne skrócenie procesu normalizacji.



Rys. 5. Sieć Petriego dla zmodyfikowanego systemu mieszania cieczy
Fig. 5. Petri net for modified system of liquid mixing

Stosując zmodyfikowany wzór (6), dla analizowanej sieci Petriego wyznaczono sekwent reprezentujący blokady (tab. 1).

Tab. 1. Sekwent dla blokad
Tab. 1. Sequent for a deadlocks

Sekwent dla blokad
$((M7+p6) \rightarrow M1) * ((p8+M8) \rightarrow M7) * (p13 \rightarrow p6) * (M6 \rightarrow p6) * (p14 \rightarrow (M8+p13+M6)) * (p7 \rightarrow p8) * (p8 \rightarrow p7) * (M1 \rightarrow (p8+p14))$

W podobny sposób określa się sekwent reprezentujący pułapki (tab. 2).

Tab. 2. Sekwent dla pułapek
Tab. 2. Sequent for a traps

Sekwent dla pułapek
$((M1 \rightarrow (M7+p6)) * (M7 \rightarrow (p8+M8)) * (p6 \rightarrow p13) * (p6 \rightarrow M6) * ((M8+p13+M6) \rightarrow p14) * (p8 \rightarrow p7) * (p7 \rightarrow p8) * ((p8+p14) \rightarrow M1))$

W wyniku normalizacji sekwentów otrzymano minimalne blokady i pułapki (tab. 3). Wcześniej usunięte zostały zbyteczne, prawe strony sekwentów.

Tab. 3. Minimalne blokady i pułapki
Tab. 3. Minimal deadlocks and traps

Blokady	Pułapki
$M1, M7, M8, p14$	$M7, M8, p14, M1$
$M1, M7, p8, p7$	$M7, p8, p7, M1$
$M1, p6, p13, p14$	$p6, p13, M6, p14, M1$
$M1, p6, M6, p14$	-

W tabeli 3 zaznaczono dwie blokady ($M1, p6, p13, p14$ i $M1, p6, M6, p14$), które nie są pokryte przez minimalne oznakowane w znakowaniu początkowym pułapki. Wskazują one dokładnie miejsca zastoju w projektowanym systemie sterowania.

6. Podsumowanie

W artykule przedstawiono metodę wykrywania zastoju [6] w układach sterowania binarnego z wykorzystaniem algorytmu Gentzena. W odróżnieniu do innych rozwiązań wykorzystujących metody symboliczne dzięki użyciu algorytmu Gentzena pominięta została transformacja wyrażeń do postaci reguł Horna, a także zautomatyzowany został proces badania zależności pomiędzy blokadami i pułapkami.

Przydatność tego rozwiązania została poparta implementacją eksperymentalnego systemu detekcji zastoju i zapętleń. System ten został opracowany w języku C++, bazując na schemacie z rysunku 3 oraz wcześniej wykonanym uniwersalnym programie wnioskującym [9].

7. Literatura

- [1] Adamski M.: Projektowanie układów cyfrowych systematyczną metodą strukturalną, Wydawnictwo Wyższej Szkoły Inżynierskiej w Zielonej Górze, Zielona Góra 1990.
- [2] Adamski M.: Parallel Controller Implementation using Standard PLD Software. FPGAs. Abingdon, England 1991, pp 296-304.
- [3] Barkaoui K., Minoux M.: A polynomial-time graph algorithm to decide liveness of some basic classes of bounded Petri nets. Lecture Notes in Computer Science, Berlin 1992: Springer Verlag, Vol. 616, ss. 62-75.
- [4] Commoner F.: Deadlocks in Petri Nets. - Wakefield 1972: Applied Data Res. Inc.
- [5] Mathony H.J.: Universal logic design algorithm and its application the synthesis of two-level switching circuits. IEE Proceedings Letters, Elsevier Science Publishers (North Holland), Vol. 29, 1990, pp. 195-210.
- [6] Murata T.: Petri nets: properties, analysis and applications. Proceedings of the IEEE 1989, Vol. 77, No. 4, ss. 541-580.
- [7] Suraj Z., Szpyrka M.: Sieci Petriego I PN-Tools. Wydawnictwo Wyższej Szkoły Pedagogicznej, Rzeszów 1999.
- [8] Pawlak Z.: Automatyczne dowodzenie twierdzeń. Państwowe Zakłady Wydawnictw Szkolnych, Warszawa 1965.
- [9] Tkacz J.: Komputerowe wspomaganie przekształceń symbolicznych Gentzena. Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim : IV Krajowa konferencja. Kraków, Polska, 2003, ss. 113-116.
- [10] Węgrzyn A.: Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego. Wydawnictwo Politechniki Zielonogórskiej, Zielona Góra 1996.