

lującym analizę pól ramek ethernetowych oraz przeszukiwanie tablicy reguł. W praktyce sprowadziło się to do pobierania z pamięci ramkowej danych nagłówka, następnie wprowadzenia opóźnienia symulującego jego weryfikację, a na koniec zatwierdzania każdej z odebranych ramek.

■ Tabela 1. Wykorzystanie zasobów sprzętowych

Zajętość zasobów układowych Virtex II Pro XC2VP30-7		
	Liczba wykorzystanych elementów	Utylizacja procentowa
Liczba elementów Slices	222 z dostępnych 13 696	1%
Liczba elementów z przerzutnikami	260 z dostępnych 27 392	1%
Liczba czterowieściowych komórek LUT	409 z dostępnych 27 392	1%
Liczba bloków pamięci BRAM	70 z dostępnych 136	51%

Wykorzystanie zasobów układu FPGA dla dwóch bloków FW (z pominięciem kontrolerów sieciowych MAC) przedstawiono w tabeli 1. Orientacyjna maksymalna szybkość pracy pamięci ramkowej wynosi około 350 MHz.

Kolejnym istotnym etapem prowadzonych prac będzie opracowanie modułu weryfikującego reguły bezpieczeństwa.

Praca naukowa finansowana ze środków na naukę w latach 2006–2008 jako projekt badawczy.

LITERATURA

- [1] 802.3 IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks, 2002
- [2] Loinig J., Wolkerstorfer J., Szekely A. *Packet Filtering in Gigabit Networks Using FPGAs*, Austrochip 2007 – Proceedings of the 15th Austrian Workshop on Microelectronics, 2007
- [3] Moscola J., Lockwood J., Loui R. P., Pachos M., *Implementation of a Content-Scanning Module for an Internet Firewall*, Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines, 2003
- [4] Sułkowski G., Twardy M., Wiatr K.: *Implementacja systemu bezpieczeństwa typu Firewall dla potrzeb sieci Ethernet w oparciu o układy reprogramowalne FPGA*, Konferencja KNWS'07, Kwartalnik Pomiary, Automatyka i Kontrola, Warszawa, nr 5, 2007
- [5] Sułkowski G., Twardy M., Wiatr K.: *Implementacja standardu sieci Ethernet IEEE 802.3 w układach FPGA na potrzeby systemu bezpieczeństwa typu Firewall*, Konferencja: Reprogramowalne Układy Cyfrowe 2007, Kwartalnik Pomiary, Automatyka i Kontrola nr 7, 2007
- [6] Wiatr K.: *Akcylacja obliczeń w systemach wizyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2003

Artykuł recenzowany



Jacek TKACZ*, Marian ADAMSKI*

Wykorzystanie komputerowego wnioskowania w projektowaniu kombinacyjnych układów sterowania

Dotychczasowe badania, prowadzone nad komputerowym wnioskowaniem symbolicznym [1, 12, 13], uwiaryściły jego przydatność do rozwiązywania różnorodnych problemów kombinatorycznych metodami formalnymi. Dokładne algorytmy analizy funkcji logicznych silnie nieokreślonych, zwłaszcza przedstawionych w postaci symbolicznej, opisane w literaturze przedmiotu są bardzo pracochłonne [6, 7]. W artykule przedstawiono zastosowanie wnioskowania symbolicznego Gentzena do uzyskiwania już uproszczonego układu kombinacyjnego na podstawie przykładu o charakterze akademickim, zaczerpniętego z literatury. Jądem systemu wnioskującego jest system dowodzenia twierdzeń w zdaniowym rachunku sekwentów Gentzena [4].

Należy tu zaznaczyć, że celem badań nie jest poszukiwanie narzędzia, znacznie sprawniejszego od minimalizatorów zawartych w systemach profesjonalnych, gdzie poszukuje się rozwiązań suboptymalnych, wprowadzając wyrafinowaną heurystykę. Przedstawione w pracy oprogramowanie ma charakter uniwersalny, daje rozwiązania dokładne, nawet bez interakcji z projektantem, a na dodatek dokumentacja (protokół) wnioskowania jest formalnym dowodem poprawności analizy lub syntezy układu kombinacyjnego. Nie oznacza to jednak, że w trakcie badań eksperymentalnych projektant nie może wprowadzić elementów heurystyki, dzieląc na przykład skomplikowane zadania na podzadania [7]. Największą zaletą proponowanej metody jest jej elastyczność, nieoceniona zwłaszcza w badaniach naukowych.

SYSTEM WNISKOWANIA GENTZENA

Rozpatrywane są sekweny w logice zdań. Sekwentem nazywa się uporządkowaną parę (Γ, Δ) skończonych zbiorów formuł $\Gamma = \{A_1, A_2, \dots, A_m\}$, $\Delta = \{B_1, B_2, \dots, B_n\}$. W tradycyjnym ujęciu zbiory te są uporządkowane jako ustalone ciągi formuł. Zamiast pisać (Γ, Δ) stosuje się notację $\Gamma \vdash \Delta$. Intuicyjnie definiując sekwent $A_1, A_2, \dots, A_m \vdash B_1, B_2, \dots, B_n$ podaje się, że jest on spełniony dla wartościowań v wtedy i tylko wtedy, gdy dla tych samych wartościowań formuła $(A_1 * A_2 * \dots * A_m) \rightarrow (B_1 + B_2 + \dots + B_n)$ jest spełniona. W przyjętej notacji symbol $\rightarrow$ oznacza implikację, symbol $+$ dysjunkcję (alternatywę niewyłączającą), natomiast symbol $*$ jest symbolem koniunkcji.

Wykorzystując figury wnioskowania (zwane również regułami wnioskowania) eliminuje się stopniowo wszystkie spójniki logiczne występujące w formułach A i B, tworząc drzewo dowodu. Przykładowymi figurami wnioskowania są eliminacja spójnika koniunkcji po prawej stronie sekwentu i eliminacja spójnika dysjunkcji po lewej stronie sekwentu (1).

$$\frac{\Lambda / -\Theta\Phi * \Psi\Gamma}{\Lambda / -\Theta\Phi\Gamma} \quad \frac{\Theta\Phi + \Psi\Gamma / -\Pi}{\Theta\Phi\Gamma / -\Pi \quad \Theta\Psi\Gamma / -\Pi} \quad (1)$$

Intuicyjnie jasny jest sposób eliminacji spójnika negacji $\neg$, polegający na przeniesieniu negowanej formuły na przeciwną stronę sekwentu (2).

$$\frac{\Theta\Phi / \Psi\Gamma / -A}{\Theta\Phi\Gamma / -\Lambda\Psi} \quad \frac{\Lambda / -\Theta\Phi / \Psi\Gamma}{\Lambda\Psi / -\Theta\Phi\Gamma} \quad (2)$$

* Uniwersytet Zielonogórski, Instytut Informatyki i Elektroniki,
e-mail: J.Tkacz@iie.uz.zgora.pl, M.Adamski@iie.uz.zgora.pl

Rozpatrywany sekwent jest tautologią, gdy wszystkie liście w drzewie dowodu są tautologiami. W przypadku gdy niektóre liście w drzewie dowodu nie są tautologiami, sekwent nosi nazwę sekwentu warunkowego. Na podstawie tych sekwentów określa się wartościowania, dla których rozpatrywany sekwent macierzysty jest spełniony lub niespełniony.

Pierwszą monografią, dotyczącą wykorzystania rachunku sekwentów do formalnego projektowania układów cyfrowych, była książka [1]. Rachunek sekwentów Gentzena przedstawiono w języku polskim w książce Z. Pawłaka [9]. Jednymi z pierwszych książek opierających wykład logiki w całości na rachunku sekwentów były książki [3], [4]. Stosunkowo niedawno rachunek sekwentów znalazł właściwe sobie miejsce w wykładach logiki dla informatyków [2], [5].

Podstawy teoretyczne metody minimalizacji funkcji logicznej metodą wnioskowania Gentzena

W książce [4] wykazano, że dla każdej formuły zdaniowej A można znaleźć formułę A' w równoważnej normalnej postaci koniunkcyjnej, startując od sekwentu $\vdash A$. Podobnie, biorąc jako punkt wyjścia sekwent $A \vdash$, dochodzi się do normalnej postaci dysjunkcyjnej. Należy tu podkreślić, że w przypadku ogólnym formuła A może zawierać nawiasy oraz funktory logiczne takie jak $\neg$ (*), $\vee$ (+), $\neg$ (/), IMPLIKACJA ($\rightarrow$), RÓWNOWAŻNOŚĆ ($\leftrightarrow$), ALTERNATYWA WYŁĄCZAJĄCA ($\leftrightarrow$).

W nowym, oryginalnym systemie wnioskującym, w celu optymalizacji wyniku i uzyskania wyłącznie formuł odpowiadających implikantom lub implicentom prostym, dodatkowo zastosowano sposób wykorzystujący zasadę rezolucji, opierający się na regułach cięcia, consensusu i pochłaniania. W celu usprawnienia obliczeń wprowadzono reguły obcinania gałęzi drzewa po zastosowaniu skracania, wzorując się na pracach Thelena-Mathonego [8].

Przykład projektowania kombinacyjnego układu sterującego

W celu zademonstrowania przydatności proponowanej metody do celów badawczych, projektowych lub edukacyjnych zaadaptowano przykład zaczerpnięty z nowego wydania [11] klasycznego podręcznika wydanego przez zespół prof. J. Siwińskiego. Binarny kombinacyjny układ sterujący opisano w następujący sposób (Zad. 2.17): Zaprojektować układ przełączający, umożliwiający sterowanie pracą pomp P1, P2, P3, P4 w zależności od sygnałów wejściowych $x_1, x_2, x_3, r_1, r_2, r_3, r_4$. Stan pomp w zależności od tych sygnałów, jest następujący:

- dla $x_1=1$ — powinna pracować 1 pompa,
- dla $x_2=1$ — powinny pracować 2 pompy,
- dla $x_3=1$ — powinny pracować 3 pompy,
- dla $x_1=x_2=x_3=0$ — żadna z pomp nie pracuje,
- dla $r_1=1$ — odstawiona do remontu pompa P1,
- dla $r_2=1$ — odstawiona do remontu pompa P2,
- dla $r_3=1$ — odstawiona do remontu pompa P3,
- dla $r_4=1$ — odstawiona do remontu pompa P4,
- dla $r_1=r_2=r_3=r_4=0$ — żadna z pomp nie jest odstawiona do remontu.

Należy przyjąć, że do remontu może być odstawiona co najwyżej jedna pompa. Silniki pomp P1, P2, P3, P4 są załączone do sieci stycznikami Z1, Z2, Z3, Z4. Przy pojawieniu się sygnałów x_1, x_2, x_3 pompy powinny się włączać według narastających indeksów (z pominięciem pompy odstawionej do remontu).

W podręczniku zastosowano opis w postaci klasycznej tablicy decyzyjnej (tablicy zespołu czterech funkcji logicznych), zamieszczając 20 zależności między wektorami wejściowymi i wyjściowymi oraz pomijając 108 wierszy, dla których sygnały wyjściowe są nieokreślone (tabela 1).

■ Tabela 1. Tabela decyzyjna rozpatrywanego układu

	x_1	x_2	x_3	r_1	r_2	r_3	r_4	z_1	z_2	z_3	z_4
0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0
64	1	0	0	0	0	0	0	1	0	0	0
72	1	0	0	1	0	0	0	0	1	0	0
68	1	0	0	0	1	0	0	1	0	0	0
66	1	0	0	0	0	1	0	1	0	0	0
65	1	0	0	0	0	0	1	1	0	0	0
32	0	1	0	0	0	0	0	1	1	0	0
40	0	1	0	1	0	0	0	0	1	1	0
36	0	1	0	0	1	0	0	1	0	1	0
34	0	1	0	0	0	1	0	1	1	0	0
33	0	1	0	0	0	0	1	1	1	0	0
16	0	0	1	0	0	0	0	1	1	1	0
24	0	0	1	1	0	0	0	0	1	1	1
20	0	0	1	0	1	0	0	1	0	1	1
18	0	0	1	0	0	1	0	1	1	0	1
17	0	0	1	0	0	0	1	1	1	1	0
3	0	0	0	0	0	1	1	-	-	-	-
pozostałe								...			
127	1	1	1	1	1	1	1	-	-	-	-

W rozpatrywanym przypadku liczba wygenerowanych implikantów i ich selekcja (wyrażenie logiczne ma aż 11 zmiennych) powoduje nadmierne wydłużenie się czasu obliczeń.

Jak pokazano w pracach [1, 13], w wielu przypadkach wygodniej jest przedstawić poszczególne zależności w układzie kombinacyjnym w zwarty sposób, odzwierciedlający opis słowny (tabela 2), a następnie przekształcić je systemem wnioskującym do postaci zawsze znacznie uproszczonych sekwentów-klauzul.

■ Tabela 2. Specyfikacja behawioralna (założenie o otwartości świata)

Specyfikacja behawioralna (OWA)
$\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee \neg z_1 \vee z_2 \vee z_3 \vee z_4$; $x_1 \vee \neg(r_1 \leftrightarrow z_1) \vee (r_1 \leftrightarrow z_2) \vee z_3 \vee z_4$; $x_2 \vee \neg(r_1 \leftrightarrow z_1) \vee (r_2 \leftrightarrow z_2) \vee ((r_1 \vee r_2) \leftrightarrow z_3) \vee z_4$; $x_3 \vee \neg(r_1 \leftrightarrow z_1) \vee (r_2 \leftrightarrow z_2) \vee (r_3 \leftrightarrow z_3) \vee ((r_1 \vee r_2 \vee r_3) \leftrightarrow z_4)$;

Po zakończeniu procesu wnioskowania (przekształceniu sekwentów do postaci zbioru sekwentów-klauzul, niebędących aksjomatami logicznymi, przypisanych do liści drzewa dowodu) otrzymuje się uproszczony zbiór reguł typu *if...then*, opisujący każdy sterownik pompy oddzielnie.

■ Tabela 3. Definicje typu zmiennych

Definicje zmiennych
IN: $x_1, x_2, x_3, r_1, r_2, r_3, r_4$; OUT: z_1, z_2, z_3, z_4 ;

■ Tabela 4. Specyfikacja znormalizowana i uporządkowana

Regułowy opis sterownika
$x_1 \vee r_1 \vee x_2 \vee r_1 \vee x_3 \vee r_1 \vee \neg z_1$; $x_1 \vee r_1 \vee x_2 \vee r_2 \vee x_3 \vee r_2 \vee \neg z_2$; $x_2 \vee r_1 \vee x_2 \vee r_2 \vee x_3 \vee r_3 \vee \neg z_3$; $x_3 \vee r_1 \vee x_3 \vee r_2 \vee x_3 \vee r_3 \vee \neg z_4$; $\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee r_1 \vee x_1 \vee r_1 \vee x_2 \vee r_1 \vee x_3 \vee \neg z_1$; $\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_1 \vee r_1 \vee r_2 \vee x_2 \vee r_2 \vee x_3 \vee \neg z_2$; $\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_1 \vee x_2 \vee r_1 \vee r_2 \vee r_3 \vee x_3 \vee \neg z_3$; $\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_1 \vee x_2 \vee x_3 \vee r_1 \vee r_2 \vee r_3 \vee \neg z_4$;

Wprowadzenie dodatkowego określenia typu zmiennych (wejściowe/wyjściowe) (tabela 3) umożliwia uporządkowanie opisu i pogrupowanie znormalizowanych klauzul w grupy definiujące poszczególne wyjścia układu sterującego (tabela 4).

Na podstawie podanej specyfikacji (tabela 1) wyznacza się dwie równoznaczne formy opisu układu (tabela 2): dla wyjść aktywnych (z_1, z_2, z_3, z_4) oraz dla wyjść nieaktywnych ($/z_1, /z_2, /z_3, /z_4$). Forma ta jest bardzo dogodna do wprowadzenia do systemu ESPRESSO i umożliwia bezpośrednią współpracę programu wnioskującego z uniwersyteckimi programami do analizy i syntezy układów cyfrowych np. Demain [6].

Analiza i symulacja logiczna

Znormalizowane klauzule odzwierciedlające prace poszczególnych pomp mogą być w sposób automatyczny transformowane na język VHDL, symulowane w komfortowym środowisku projektowym, a następnie jednoznacznie odwzorowane w strukturze matrycowej rekonfigurowanego układu cyfrowego.

Stosując założenie o otwartym świecie (OWA), można wprowadzać do znormalizowanej specyfikacji układu dodatkowe założenia i badać jego zachowanie. Analiza jest bardzo przejrzysta dla człowieka, gdyż odzwierciedla naturalny tok jego myślenia. Przykładowo, chcąc zbadać, jak zachowa się układ po wciśnięciu przycisku x_1 , należy dodać do znormalizowanej specyfikacji wyrażenie (3).

$$I \rightarrow x_1 * x_2 * x_3; \quad (3)$$

W wyniku ponownej normalizacji systemem wnioskującym i uporządkowania wyrażeń ze względu na typy zmiennych otrzymuje się równania opisujące zachowanie układu (Tab. 5).

Przyjęte założenie otwartego świata spowodowało wyznaczenie wszystkich możliwych przypadków wynikających z pierwotnej specyfikacji kombinacyjnego układu sterującego, przy wciśniętym

■ Tabela 5. Symulacja logiczna dla wciśniętego przycisku pierwszego

Wyznaczone zachowanie układu	
$x_1 * r_1 \mid -z_1;$	$x_1 * r_1 \mid -z_2;$
$r_1 * x_1 \mid -z_1;$	$x_1 * r_1 \mid -z_2;$
$x_1 \mid -z_3;$	$x_1 \mid -z_4;$
$/x_1 + x_2 + x_3 \mid -;$	

pierwszym przycisku uruchamiającym jedną pompę. Z analizy sekwentów opisujących zachowanie układu dowiadujemy się, że jeżeli jest wciśnięty przycisk pierwszy i pompa pierwsza nie jest odstawiona do remontu, to będzie działać pierwsza pompa ($x_1 * r_1 \mid -z_1$). Jeśli pierwsza pompa jest odstawiona do remontu, a przycisk pierwszy jest wciśnięty, to będzie działać druga pompa ($x_1 * r_1 \mid -z_2$). Pompa pierwsza nie działa, jeśli przycisk pierwszy jest wciśnięty, a pompa jest remontowana ($r_1 * x_1 \mid -z_1$). Pompa druga nie działa, jeśli przycisk pierwszy jest wciśnięty i pompa druga jest remontowana ($x_1 * r_1 \mid -z_2$). Jeżeli przycisk pierwszy jest wciśnięty, to pompa trzecia i czwarta nie działają ($x_1 \mid -z_3; x_1 \mid -z_4$). Ostatni sekwent ($/x_1 + x_2 + x_3 \mid -$) informuje o zachowaniu układu w przypadku, gdy przycisk pierwszy nie jest wciśnięty, a może być wciśnięty drugi lub trzeci. Widać tutaj, że symulacja zgadza się ze specyfikacją i sterownik kombinacyjny działa prawidłowo.

Istnieje również możliwość symulowania sytuacji nieprzewidzianych w specyfikacji, np. dwa przyciski wciśnięte jednocześnie. Analizując w ten sposób układ, można znaleźć błędy w specyfikacji lub miejsca, w których projektowany układ jest niedospecyfikowany. W bardzo łatwy sposób można rozbudowywać też specyfikację (tabela 1), wprowadzając dodatkowe reguły lub rozbudowując istniejące, które zabezpieczą układ przed nieprzewidzianym zachowaniem.

W rozbudowanej specyfikacji wprowadzono dodatkowe warunki zabezpieczające przed wciśnięciem więcej niż jednego przycisku. Zabezpieczenie takie można zweryfikować wtórnie normaliza-

■ Tabela 6. Specyfikacja behawioralna z zabezpieczeniem

Specyfikacja behawioralna (OWA)
$/x_1, /x_2, /x_3 \mid -z_1 * z_2 * z_3 * z_4;$
$x_1 * x_2 * x_3 \mid -(/r_1 < -> z_1) * (r_1 < -> z_2) * z_3 * z_4;$
$x_2 * x_1 * x_3 \mid -(/r_1 < -> z_1) * (/r_2 < -> z_2) * ((r_1 + r_2) < -> z_3) * z_4;$
$x_3 * x_1 * x_2 \mid -(/r_1 < -> z_1) * (/r_2 < -> z_2) * (/r_3 < -> z_3) * ((r_1 + r_2 + r_3) < -> z_4);$

cją z dodaniem wyrażenia definiującego wciśnięcie więcej niż jednego przycisku (4).

$$I \rightarrow x_1 * x_2 * x_3; \quad (4)$$

W wyniku wnioskowania powstaje tylko jeden sekwent (5) informujący o zachowaniu układu w przypadku, gdy przyciski pierwszy i trzeci nie są wciśnięte (5).

$$I \rightarrow x_1 + x_2 + x_3; \quad (5)$$

Brak jakichkolwiek rozwiązań informuje, że układ nie zadziała przy wciśniętym więcej niż jednym przycisku.

* * *

Artykuł jest ilustracją możliwości zastosowania komputerowego wnioskowania symbolicznego Gentzena do rozwiązywania skomplikowanych problemów logicznych, jako formalna procedura dowodzenia. Zawile zależności między parametrami mogą być przekształcone do prostszych, łatwiejszych do weryfikacji relacji binarnych, co jest szczególnie istotne w projektowaniu sterowników bezpiecznych metodą formalnego przekształcania specyfikacji logicznej na opis implementacji. Dokumentacja w postaci drzewa jest równocześnie dowodem formalnym, wykazującym poprawność wykonanych przekształceń. Eksperymentalne oprogramowanie może być wykorzystywane samodzielnie lub stanowić nakładkę do systemu ESPRESSO. Rezultaty projektowania w postaci regułowej w łatwy sposób są transformowane na opis w języku VHDL.

LITERATURA

- [1] Adamski M.: *Projektowanie układów cyfrowych systematyczną metodą strukturalną*, Wydawnictwo Wyższej Szkoły Inżynierskiej w Zielonej Górze, Zielona Góra 1990.
- [2] Ben-Ari M.: *Logika matematyczna w informatyce*, Wydawnictwa Naukowo-Techniczne, Warszawa 2005.
- [3] Ershov Yu. L., Palyutin E. A.: *Mathematical Logic*, Mir Publisher, Moscow 1984.
- [4] Gallier J. H.: *Logic for computer science. Foundations of Automatic Theorem Proving*, Harper & Row 1986.
- [5] Huzar Z.: *Elementy logiki dla informatyków*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2002.
- [6] Łuba T., Rawski M., Tomasiewicz P., Zbierchowski B.: *Synteza układów cyfrowych*, Praca zbiorowa pod redakcją prof. Tadeusza Łuby, Wydawnictwa Komunikacji i Łączności, Warszawa 2003.
- [7] Kamionka-Mikuła H., Małysiak H., Pochopień B.: *Synteza i analiza układów cyfrowych*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 2006.
- [8] Mathony H. J.: *Universal logic design algorithm and its application to the synthesis of two-level switching circuits*, IEE Proceedings-E, Computers And Digital Techniques, Vol. 136, Part E, Number 3, May 1989.
- [9] Pawlak Z.: *Automatyczne dowodzenie twierdzeń*, Państwowe Zakłady Wydawnictw Szkolnych, Warszawa 1965.
- [10] Perkowski M., Rydzewski A., Misiurewicz P.: *Teoria układów logicznych – zagadnienia wybrane*, Wydawnictwa Politechniki Warszawskiej, Warszawa 1977.
- [11] Siwiński J., Małysiak H. (red.): *Zbiór zadań z układów przełączających*, Wydawnictwo Politechniki Śląskiej, Gliwice 2003.
- [12] Szajna J.: *Projektowanie układów cyfrowych z wykorzystaniem programowania logicznego*, Wydawnictwo Politechniki Zielonogórskiej, Zielona Góra 1996.
- [13] Tkacz J.: *Komputerowe wspomaganie przekształceń symbolicznych Gentzena*, IV Krajowa Konferencja: Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, Kraków, Polska, 2003.

Artykuł recenzowany