Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej nr 36

XXIII Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE' 2013
Oddział Gdański PTETiS
Referat nr 12

REDUKCJA CZASU ANALIZY MZP PRZEZ OGRANICZENIE ROZMIARU ROZWIĄZANIA

Grzegorz GOŁASZEWSKI[1]

1. Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk
tel: +48 58 347 10 37 fax: +48 58 347 27 27 e-mail: grzo@eti.pg.gda.pl

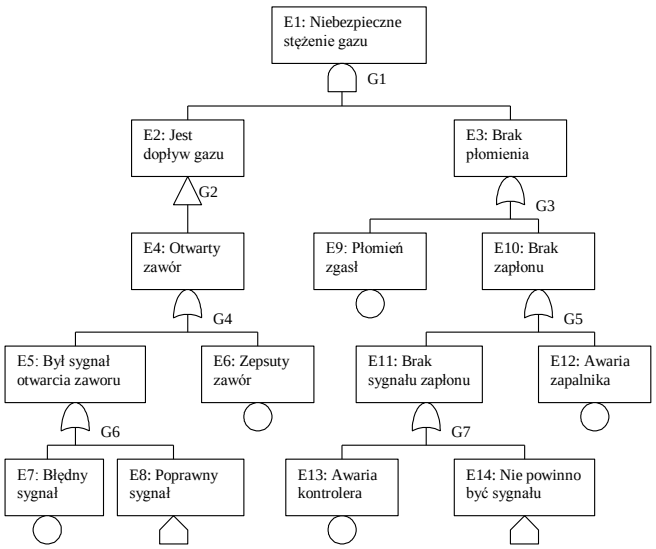
Streszczenie: Analiza drzew niezdatności jest uznaną metodą analizy bezpieczeństwa systemów. Notacja ECSDM pozwala definiować zależności czasowe między zdarzeniami drzewa oraz przeanalizować je w celu określenia zależności pomiędzy zdarzeniami z Minimalnych Zbiorów Przyczyn (MZP). Dzięki wprowadzeniu klasyfikacji zdarzeń z MZP można wyodrębnić zależności czasowe istotne dla zapobiegania wywoływania hazardu przez konkretny MZP. Pozostałe zależności czasowe są w tym podejściu odrzucane. Opracowano odmiany algorytmu analizy zależności czasowych w drzewach niezdatności, które są w stanie odrzucić część zależności czasowych na wcześniejszym etapie analizy. W tym artykule omówiony zostanie eksperyment określający redukcję czasu przetwarzania uzyskaną dzięki zastosowania tych zmian.

Słowa kluczowe: drzewa niezdatności, zależności czasowe, ECSDM

1. WSTĘP

W zależności od planowanego wykorzystania rozważanego systemu może on, poprzez swoje działanie, wyrządzić znaczne szkody swojemu otoczeniu, zarówno bezpośredniemu (np. obsługa), jak i dalszemu (np. środowisko naturalne). W sytuacji, gdy szkody są niezamierzone i mogą być poważne, system taki poddaje się różnorakim analizom. Mogą one mieć na celu określenie możliwych sposobów negatywnego wpływu systemu na otoczenie, skutków takich zdarzeń, a także mechanizmów ich występowania oraz możliwych sposobów zapobiegania im.

Sposobem na określenie przyczyn konkretnego niepożądanego zdarzenia (hazardu) jest m.in. analiza drzew niezdatności. Bazuje ona na graficznej notacji drzewa, w którego korzeniu znajduje się niepożądane zdarzenie. Do niego, poprzez bramki logiczne podczepiane są kolejne zdarzenia, stanowiące jego bezpośrednie przyczyny. Struktura ta rozwijana jest iteracyjnie, poprzez dodawanie bezpośrednich przyczyn do kolejnych zdarzeń, aż do uzyskania założonej szczegółowości. Przykładowe drzewo przedstawiono na rysunku 1. Tak zbudowane drzewo można poddać analizie. Podstawową jej formą jest określenie minimalnych zbiorów przyczyn (MZP). Minimalne zbiory przyczyn są to zbiory zdarzeń podstawowych (zdarzeń występujących w liściach drzewa), których łączne wystąpienie umożliwia wystąpienie hazardu. Ich minimalność polega na tym, że nie można z takiego zbioru usunąć żadnego ze zdarzeń, bez naruszenia właściwości umożliwiania wystąpienia hazardu. Możliwa jest również analiza probabilistyczna omawianego modelu, jednak wymaga to wprowadzenia dodatkowych informacji. Po uzupełnieniu drzewa o prawdopodobieństwa wystąpienia zdarzeń podstawowych, możliwe jest obliczenie prawdopodobieństwa wystąpienia hazardu albo określenie, które zdarzenia mają największy wpływ na jego wystąpienie [1].



Rys. 1 Drzewo niezdatności dla systemu palnika gazowego [2]

Problemem analizy drzew niezdatności jest jednak określenie relacji czasowych pomiędzy zdarzeniami z MZP. Przyjmuje się, że muszą one wystąpić równocześnie. W niektórych wypadkach może jednak zachodzić potrzeba precyzyjniejszego wyrażenia tej zależności i/lub uwzględnienie zdarzeń zachodzących nierównocześnie, a mogących przyczynić się do zdarzenia niepożądanego.

W celu rozwiązania tego problemu zaproponowano notacje precyzujące zależności czasowe pomiędzy zdarzeniami dla poszczególnych bramek. Jedno podejście polega na użyciu różnego rodzaju logik temporalnych (np. [3]), inne pozwala na ilościowe wyrażenie zależności czasowych pomiędzy zdarzeniami [4] – notacja Extended Common Safety

Description Model (ECSDM). Górski oraz Wardziński opisali sposób przedstawiania zależności czasowych opisanych tą notacją w formie grafu oraz zaproponowano algorytm wnioskowania [5]. Razem z Magotem przedstawili sposób konwersji relacji czasowych do postaci czasowej sieci Pertiego [6], co dało podstawę do zaprojektowania nowego algorytmu analizy zależności czasowych [7]. Główną różnicą pomiędzy powyższymi algorytmami jest to, iż ten pierwszy koncentruje się na zależnościach czasowych pomiędzy zdarzeniami z MZP, a drugi na zależnościach pomiędzy poszczególnymi zdarzeniami z MZP a hazardem.

W dalszej części referatu zostaną krótko omówione podstawowe pojęcia notacji ECSDM oraz zasada działania algorytmu przetwarzania zależności czasowych wyrażonych w tej notacji. Następnie wprowadzona zostanie idea klasyfikacji zdarzeń podstawowych w drzewie niezdatności i zostaną pokrótce omówione warianty algorytmu wnioskującego o zależnościach czasowych w drzewach niezdatności. W dalszej części opisany zostanie eksperyment mający na celu porównanie zaproponowanych przez autora algorytmów.

2. WNIOSKOWANIE O ZALEŻNOŚCIACH CZASOWYCH

Notacja ECSDM składa się z dwu części: statycznej i dynamicznej. Statyczna pozwala określić zdarzenia występujące w analizowanym systemie w kontekście stanów elementów systemu. Dynamiczna pozwala wypowiadać się o zależnościach czasowych występujących pomiędzy zdarzeniami. Główne pojęcia notacji to:

- E – zbiór zdarzeń, oznaczanych wielkimi literami, np. X, Y,
- L – zbiór etykiet opisujących poszczególne wystąpienie zdarzeń, oznaczanych np. l, m, n,
- A – zbiór akcji (etykietowanych zdarzeń),

$$A = \{E \times L\} \tag{1}$$

Akcje oznacza się jako np. x, y, z,

- T – zbiór tranzycji (początków i końców akcji),

$$T = \{x_s | x \in A\} \cup \{x_e | x \in A\} \tag{2}$$

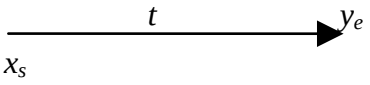
gdzie: x_s – moment rozpoczęcia akcji x, x_e – moment zakończenia akcji x.

Notacja ECSDM wprowadza konstrukcje pozwalające wyrazić zależności czasowe pomiędzy zdarzeniami, jednak dla celów analizy sprowadzane one są do postaci znormalizowanej:

$$start(x) + t \geq end(y) \tag{3}$$

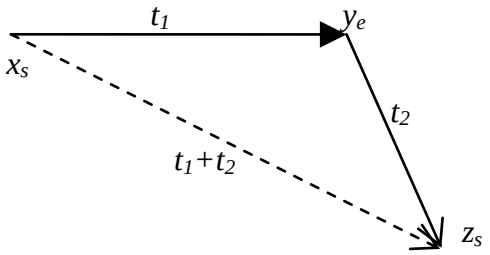
gdzie: t – stała, start(x) – nieznany czas rozpoczęcia akcji x (czas wystąpienia tranzycji x_s), end(y) – nieznany czas zakończenia akcji y.

W literaturze pokazano, że takie wyrażenie można reprezentować przez jako krawędź w grafie [5]:



Rys. 2 Grafowa reprezentacja zależności czasowej

Zdefiniowano tam również regułę wnioskowania, którą można przedstawić jako:



Rys. 3 Reguła wnioskowania

Krawędź narysowana przerywaną linią obrazuje wyrażenie wywnioskowane na podstawie pozostałych dwóch. Na podstawie tej reguły zdefiniowano algorytm MZPCALC_FULL, który schematycznie można przedstawić następująco:

- przedstaw wyrażenia czasowe dla szczytowej bramki drzewa niezdatności jako graf (lub grafy dla bramki LUB – po jednym dla każdej alternatywy),
- dodaj wszystkie możliwe krawędzie wynikające z przedstawionej reguły,
- usuń wierzchołki odpowiadające zdarzeniu wyjściowemu tej bramki,
- do uzyskanego grafu (grafów) po kolei dodawaj grafy obrazujące zależności z kolejnych bramek (jeśli dodawany jest więcej niż jeden graf powiel dotychczasowe grafy i łącz z nowymi na zasadzie każdy z każdym),
- dodaj krawędzie zgodnie z regułą,
- usuń z grafu (grafów) wierzchołki reprezentujące zdarzenie wyjściowe właśnie dodanej bramki.

Takie postępowanie doprowadzi do uzyskania zestawu grafów. Każdy z tych grafów będzie reprezentował minimalny zbiór przyczyn rozszerzony o zależności czasowe, które muszą wystąpić pomiędzy jego zdarzeniami, aby umożliwiał on wystąpienie hazardu.

Aby zminimalizować ryzyko wystąpienia hazardu można więc zapewnić, żeby dla jak największej liczby MZP chociaż jedna zależność czasowa w nim wyspecyfikowana nigdy nie wystąpiła w działającym systemie. W oparciu o tę ideę powstała metoda wyznaczania czasowych wymagań bezpieczeństwa wobec systemów związanych z bezpieczeństwem [2, 8]. Polega ona na oznaczeniu wśród zdarzeń podstawowych takich, których wystąpienie twórcy systemu mogą kontrolować (np. poprzez programowalne układy sterujące). Następnie wśród MZP wyszukiwane są zależności pomiędzy zdarzeniami kontrolowalnymi a pozostałymi typami zdarzeń (przewidywalne, obserwowalne, nieobserwowalne). Takie wyrażenia po ich zanegowaniu stanowią kandydujące wymagania wobec systemu.

Co istotne, dla MZP nie zawierających żadnego zdarzenia kontrolowalnego żadna zależność czasowa nie zostanie wytypowana. Wyznaczanie takich MZP w tym podejściu jest bezcelowe i powoduje jedynie zbędny narzut. Takie MZP należy więc jak najszybciej odrzucić już w fazie ich obliczania. Tak działa algorytm MZPCALC_CONTROLABLE, który każdorazowo po dodaniu do rozwiązania pośredniego nowej bramki i przetworzeniu wyników sprawdza powstałe grafy i usuwa ze zbioru rozwiązań te, które nie zawierają żadnych zdarzeń kontrolowalnych bądź zdarzeń posiadających w drzewie takich potomków.

Ponadto należy zauważyć, że zależności czasowe pomiędzy zdarzeniem kontrolowalnym a nieobserwowalnym mogą doprowadzić jedynie do wymagania opisującego minimalną częstotliwość występowania zdarzenia kontrolowalnego. Jeśli zdarzenia kontrolowalne w analizowanym systemie nie mogą być wyzwalane cyklicznie również MZP, w których poza jednym kontrolowalnym wszystkie zdarzenia są nieobserwowalne, mogą być odrzucone już na etapie wyznaczania. Tak działa algorytm MZPCALC_PAIR. Tak jak i poprzednia wersja po przetworzeniu każdej bramki sprawdza on wynik pośredni i odrzuca grafy nie zawierające pary: zdarzenie kontrolowalne oraz zdarzenie inne niż nieobserwowalne (lub ich przodkowie).

Algorytm ten można dalej zmodyfikować, aby nawet dla MZP zawierających parę opisaną powyżej nie przeprowadzał obliczeń dla bramek wpływających jedynie na zależności czasowe pomiędzy zdarzeniami niekontrolowanymi z danego MZP. Tak działa algorytm MZPCALC_PART, jednak do poprawnego wyznaczenia nieistotnych bramek po każdym kroku algorytmu musi on sprawdzić wszystkie ścieżki we wszystkich grafach wyniku pośredniego. Można oczekiwać, że będzie to powodowało spory narzut.

3. EKSPERYMENT

Jak można wywnioskować z powyższego opisu, przedstawione algorytmy pozwalają na uniknięcie części dotychczasowych obliczeń ale kosztem wprowadzenia nowych. Postanowiono więc sprawdzić eksperymentalnie czy wprowadzone modyfikacje algorytmów doprowadzą do redukcji czasu przetwarzania.

Wszystkie cztery algorytmy zostały zaimplementowane, a czasy ich wykonywania sprawdzono na testowym drzewie niezdatności. Drzewo to zaczerpnięto z [9]. Zawiera ono 12 bramek i 10 zdarzeń podstawowych.

Skalowalność testowanych rozwiązań sprawdzano przez przeprowadzanie obliczeń dla drzew składających się z wielu połączonych kopii wspomnianego drzewa. Natomiast aby ocenić wpływ odsetka zdarzeń nieobserwowalnych na wydajność algorytmów dla każdego rozmiaru drzewa przygotowano 3 wersje o różnej liczbie zdarzeń nieobserwowalnych.

Pomiary prowadzone były na komputerze z procesorem Intel Core i7 920 z 6GB pamięci operacyjnej oraz systemem Windows 7 Pro w wersji 64. Dla ograniczenia wpływu innych aplikacji na wynik pomiaru wyłączono aplikacje nie mające związku z pomiarami (również te działające w zasobniku systemowym (ang. systray) poza programem antywirusowym). Dla każdego drzewa i każdego algorytmu przeprowadzono po 10 pomiarów.

Poniżej przedstawione czasy pracy poszczególnych algorytmów (wyrażone w milisekundach) są więc średnią z 10 pomiarów. Liczba MZP za każdym przebiegiem algorytmu była stała.

Tablica 1. Zestawienie wyników eksperymentu

Drzewo	Liczba bram	Liczba zdarzeń kontrolowalnych	Liczba zdarzeń przewidywalnych + obserwowalnych	Liczba zdarzeń nieobserwowalnych	Liczba MZP - MZPCALC_FULL	Czas przetwarzania [ms] - MZPCALC_FULL	Liczba MZP - MZPCALC_CONTROLABLE	Czas przetwarzania [ms] - MZPCALC_CONTROLABLE	Liczba MZP - MZPCALC_PAIR	Czas przetwarzania [ms] - MZPCALC_PAIR	Liczba MZP - MZPCALC_PART	Czas przetwarzania [ms] - MZPCALC_PART
1A	12	1	1	8	170	388,5	34	68,7	16	52,6	16	53,0
1B	12	1	1	8	170	388,5	34	68,7	16	52,6	16	53,0
1C	12	2	2	6	170	394,9	98	298,7	64	274,6	60	236,0
2A	24	1	1	17	380	1006,5	76	169,0	16	83,2	16	80,6
2B	24	2	2	15	380	998,0	84	194,9	34	135,7	34	133,2
2C	24	4	4	11	380	1024,2	212	814,5	134	755,1	126	627,4
3A	36	1	1	26	1900	5153,0	76	244,8	16	99,0	16	99,9
3B	36	3	3	22	1900	5106,3	198	461,2	84	308,7	68	293,4
3C	36	6	6	16	1900	5272,9	1270	4173,3	964	3979,6	956	3649,2
4A	48	1	1	35	3572	13085,7	76	251,1	0	34,9	0	35,6
4B	48	4	4	29	3572	13269,7	316	974,6	134	694,6	110	602,8
4C	48	8	8	21	3572	13273,9	2396	10247,4	1826	9713,1	1806	8943,2
5A	60	1	1	44	5092	18537,7	76	261,5	0	40,3	0	40,8
5B	60	5	5	36	5092	18503,3	430	1344,5	184	952,1	144	842,0
5C	60	10	10	26	5092	18945,9	3454	14356,0	2656	13208,9	2628	12446,8

Dane do eksperymentu dobrano tak, że ciąg A miał po jednym zdarzeniu kontrolowalnym i obserwowalnym niezależnie od rozmiaru drzewa. Doprowadziło to do pewnych anomalii i dla dużych drzew zdarzenia te były zbyt "rozrzedzone" aby móc wystąpić naraz w jednym MZP. W pozostałych dwóch ciągach liczby zdarzeń rosły proporcjonalnie do rozmiaru drzewa.

Przedstawione wyniki pokazują, że liczba MZP dla algorytmu przetwarzającego wszystkie minimalne zbiory przyczyn jest, jak się należało spodziewać, niezależna od liczby poszczególnych rodzajów zdarzeń podstawowych w drzewie. Jednocześnie można zaobserwować niekorzystny wzrost czasu przetwarzania wraz ze wzrostem rozmiaru drzewa. Porównując drzewo najmniejsze z największym: 5-ciokrotny wzrost rozmiaru zadania spowodował prawie 50-ciokrotne wydłużenie czasu obliczeń. Przy czym otrzymano jednocześnie prawie 30-stokrotny wzrost liczby MZP. Stąd wniosek, że to właśnie rozmiar rozwiązania determinuje czas obliczeń.

Zmniejszenie rozmiaru rozwiązania, zgodnie z założeniem, przełożyło się na redukcję czasu przetwarzania. Przy czym w każdym przypadku uzyskana redukcja czasu obliczeń jest proporcjonalnie mniejsza niż odpowiadająca jej redukcja rozmiaru rozwiązania. Jest to zgodne z oczekiwaniami, gdyż znaczna część obliczeń jest wspólna dla więcej niż jednego MZP (np. bramkę szczytową przetwarza się raz, a wykorzystuje przy obliczaniu wszystkich MZP).

Największe ograniczenie czasu przetwarzania uzyskano przechodząc z odmiany algorytmu MZPCALC_FULL na MZPCALC_CONTROLABLE. Przechodzenie na pozostałe wersje w obu przypadkach dało dodatkowe korzyści, które nie były już jednak tak znaczące. Korzyści te, choć wystąpiły w prawie wszystkich przypadkach, zależały w znaczącym stopniu od liczby zdarzeń nieobserwowalnych w analizowanym drzewie. Im większa liczba tych zdarzeń, tym większego zysku z zastosowania zaproponowanych odmian algorytmu można się spodziewać.

4. PODSUMOWANIE

W przedstawionym referacie omówiono eksperyment mający na celu weryfikację założenia o możliwej redukcji czasu przetwarzania danych przez wprowadzenie modyfikacji ograniczających rozmiar tworzonego rozwiązania do algorytmu pracującego na tych danych.

Redukcję rozmiaru rozwiązania można było rozważać z powodu specyficznego zastosowania dla otrzymywanych z algorytmu wyników, a udało się je zaimplementować w algorytmie dzięki wyraźnie wydzielonym fazom pracy algorytmu.

Eksperyment potwierdził założenia, wszystkie trzy odmiany algorytmu charakteryzowały się redukcją czasu rozwiązania zgodną z stopniem redukcji wyniku przetwarzania (porównując pomiędzy sobą poszczególne wersje).

Ocenia się więc, że wprowadzenie tych modyfikacji do algorytmu jest korzystne i celowe dla przedstawionego w referacie sposobu wykorzystania wyników pracy omawianego algorytmu.

5. BIBLIOGRAFIA

1. Vesely W. E., Goldberg F. F., Roberts N. H., Haasl D. F.: Fault Tree Handbook, U.S. Nuclear Regulatory Commission, January 1981, NUREG–0492
2. Gołaszewski G.: Automatyzacja określania wymagań bezpieczeństwa na podstawie wyników analizy czasowej drzew błędów, Technologie Informacyjne, Maj 2006, Gdańsk, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, tom 11, str. 631-638, ISBN 83-917681-8-X
3. Thums A., Schellhorn G.: Formal Safety Analysis in Transportation Control, Workshop on Software specification for safety relevant transportation control tasks, 2002
4. Górski J.: Extending Safety Analysis Techniques with Formal Semantics, Technology and Assessment of Safety Critical Systems, (Redmil F. J., Anderson T., Eds.), Springer-Verlag, 1994, ISBN 978-3-540-19859-8
5. Górski J., Wardziński A.: Timing Aspects of fault tree analysis of safety critical system, Proceedings of the Fifth Safety-critical Systems Symposium, February 1997 Brighton, pp. 231-244, ISBN 978-3540761341
6. Górski J., Magott J. and Wardziński A.: Modeling Fault Trees Using Timed Petri Nets, Safe Systems (G Rabe, ed.), Springer Verlag, 1995, pp. 90-100, ISBN 978-3-540-19962-5
7. Magott J., J. Skrobanek J.: Method of time Petri net analysis for analysis of fault trees with time dependencies, IEE Proc. Computers and Digital Techniques vol. 149 No. 6, November 2002, ISSN 1350-2387
8. Gołaszewski G., Górski J.: Hazard prevention by forced time constraints, IEEE Computer Society "Conference on Dependability of Computer Systems DepCoS – RELCOMEX'06", May 2006 Szklarska Poręba, pp. 84-91, ISBN 978-0-7695-2565-5
9. Skrobanek P.: Analiza zależności czasowych w drzewach niezdatności (rozprawa doktorska), Instytut Informatyki, Automatyki i Robotyki Politechniki Wrocławskiej, Czerwiec 2005 Wrocław

REDUCTION OF MCS ANALYSIS TIME BY REDUCING SIZE OF THE RESULT

Key-words: fault trees, timing relationships, ECSDM

Fault tree analysis is an accepted systems safety analysis method. ECSDM notation allows for definition of timing relationships between events of the tree. These relationships are then analyzed in order to derive timing relationships between events in Minimal Cut Sets. Introduction of events classification allows for extraction from MCS timing relationships relevant to prevention of that given MCS. Timing relationships not selected in that step are ignored. Therefore a set of new algorithms for analysis of timing relationships in fault trees was proposed. Each of these algorithms filters out some of the irrelevant relationships in earlier steps of the analysis. In this article an experiment aimed at determining the gain (understood as processing time reduction) from applying these algorithms is described.