Autoreferat

Andrzej Zbrzezny
Instytut Matematyki i Informatyki
Akademia im. Jana Długosza w Częstochowie

Załącznik 2

1. Imię i nazwisko:
Andrzej Zbrzezny

2. Posiadane dyplomy, stopnie naukowe i artystyczne – z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej:

- dyplom magistra informatyki, specjalność: oprogramowanie; Uniwersytet Jagielloński, 1981; tytuł pracy magisterskiej: „Optymalizacja kodu niezależna maszynowo".
- dyplom doktora nauk humanistycznych w zakresie filozofii, specjalność: logika; Uniwersytet Wrocławski, Wydział Nauk Społecznych, 1991; tytuł rozprawy doktorskiej: „Systemy logiczne związane ze strukturami częściowymi".

3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych i artystycznych:

- od 01.12.1981 r. do 30.09.1983 r. – asystent stażysta, Instytut Matematyki, Wyższa Szkoła Pedagogiczna w Częstochowie;
- od 01.10.1983 r. do 30.09.1991 r. – asystent, Instytut Matematyki, Wyższa Szkoła Pedagogiczna w Częstochowie;
- od 01.10.1991 r. do 30.09.1998 r. – adiunkt, Instytut Matematyki, Wyższa Szkoła Pedagogiczna w Częstochowie;
- od 01.10.1994 r. do 30.09.1995 r. – adiunkt, Wydział Zarządzania, Politechnika Częstochowska;
- od 01.10.1998 r. do 30.09.2004 r. – adiunkt, Instytut Matematyki i Informatyki, Wyższa Szkoła Pedagogiczna w Częstochowie;
- od 01.10.2004 r. do teraz – adiunkt, Instytut Matematyki i Informatyki, Akademia im. Jana Długosza w Częstochowie.

4. Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. nr 65, poz. 595 ze zm.):

a) tytuł osiągnięcia naukowego/artystycznego:

Wybrane aspekty ograniczonej weryfikacji modelowej systemów współbieżnych

b) (autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa):

[H1] **Andrzej Zbrzezny**. Improvements in SAT-based Reachability Analysis for Timed Automata. *Fundamenta Informaticae*, 60(1-4):417–434, 2004.

[H2] **Andrzej Zbrzezny**. SAT-based Reachability Checking for Timed Automata with Diagonal Constraints. *Fundamenta Informaticae*, 67(1-3):303–322, 2005.

[H3] Bożena Woźna, **Andrzej Zbrzezny**. Bounded Model Checking for the Existential Fragment of TCTL-G and Diagonal Timed Automata. *Fundamenta Informaticae*, 79(1-2):229–256, 2007.

[H4] **Andrzej Zbrzezny**, Agata Półrola. SAT-based Reachability Checking for Timed Automata with Discrete Data. *Fundamenta Informaticae*, 79(3-4):579–593, 2007.

[H5] **Andrzej Zbrzezny**, Bożena Woźna. Towards Verification of Java Programs in VerICS. *Fundamenta Informaticae*, 85(1-4):533–548, 2008.

[H6] **Andrzej Zbrzezny**. Improving the Translation from ECTL to SAT. *Fundamenta Informaticae*, 85(1-4):513–531, 2008.

[H7] **Andrzej Zbrzezny**. A New Translation from ECTL* to SAT. *Fundamenta Informaticae*, 120(3-4):377–397, 2012.

[H8] Bożena Woźna-Szcześniak, **Andrzej Zbrzezny**. A Translation of the Existential Model Checking Problem from MITL to HLTL. *Fundamenta Informaticae*, 122(1-2):401–420, 2013.

c) omówienie celu naukowego/artystycznego ww. pracy/prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania:

Wstęp

Podstawą mojego wniosku habilitacyjnego jest jednolity cykl publikacji dotyczących wybranych, teoretycznych i praktycznych, aspektów ograniczonej weryfikacji modelowej systemów współbieżnych. Tematyką weryfikacji modelowej, a zwłaszcza ograniczonej weryfikacji modelowej, zainteresowałem się w 2001 roku po nawiązaniu współpracy z grupą profesora Wojciecha Penczka z Instytutu Podstaw Informatyki Polskiej Akademii Nauk (IPI PAN). W ramach tej współpracy byłem podwykonawcą realizowanego w IPI PAN, w okresie od 01.07.2000 r. do 30.06.2003 r., grantu zatytułowanego „Automatyczna weryfikacja systemów zależnych od czasu" (nr 8 T11C 014 19).

Weryfikacja modelowa [D4, D19, D20] (ang. model checking) jest automatyczną techniką weryfikowania własności systemów współbieżnych takich jak: układy cyfrowe, systemy rozproszone, systemy czasu rzeczywistego, systemy wieloagentowe, protokoły komunikacyjne, protokoły kryptograficzne, programy współbieżne oraz wielu innych. Aby móc algorytmicznie sprawdzić czy dany system spełnia daną własność należy najpierw stworzyć model tego systemu, a następnie opisać w języku formalnym zarówno stworzony model, jak i tę własność.

Jedną z wielu możliwości modelowania systemów współbieżnych jest zastosowanie sieci automatów skończonych komunikujących się poprzez wspólne akcje [D45]. Inną możliwością jest zastosowanie sieci Petriego [D37]. Analogicznie, współbieżne systemy czasu rzeczywistego można modelować przy pomocy sieci automatów czasowych [D2, D31] lub sieci Petriego z czasem [D27, D31]. Niezależnie jednak od formalizmu użytego do modelowania systemu współbieżnego, podstawą do zastosowania weryfikacji modelowej jest zdefiniowanie systemu tranzycyjnego dla użytego modelu. Przez *system tranzycyjny* [D4] (ang. transition system) rozumiemy układ $(S, Act, \longrightarrow, I, AP, L)$ składający się ze zbioru stanów S, zbioru akcji Act, relacji przejścia $\longrightarrow \subseteq S \times Act \times S$, zbioru stanów początkowych $I \subseteq S$, zbioru zmiennych zdaniowych AP reprezentujących zdania proste oraz funkcji etykietującej $L : S \to 2^{AP}$. Systemy tranzycyjne nazywane są także modelami Kripkego.

Do formułowania własności systemów stosuje się odpowiednią logikę temporalną. Najczęściej w tym celu wykorzystuje się liniową logikę temporalną (ang. linear temporal logic, w skrócie LTL), temporalną logikę czasu rozgałęzionego (ang. computation tree logic, w skrócie CTL), pełną logikę czasu rozgałęzionego CTL*, zawierającą logiki LTL i CTL, uniwersalne i egzystencjalne fragmenty wymienionych logik, a także logiki będące ich modyfikacjami i rozszerzeniami.

Ograniczona weryfikacja modelowa (ang. bounded model checking, w skrócie BMC) [D6, D7, D9] jest jedną z metod symbolicznej weryfikacji modelowej. Wykorzystuje ona redukcję problemu prawdziwości formuł temporalnych w systemie tranzycyjnym do problemu spełnialności formuł klasycznego rachunku zdań (w skrócie: do problemu SAT). Wspomnianą redukcję

uzyskuje się poprzez translację relacji przejścia systemu tranzycyjnego oraz translację badanej własności do formuł klasycznego rachunku zdań. Podkreślmy, że dla danej logiki temporalnej ograniczona weryfikacja modelowa wykorzystywana jest głównie do obalania własności żywotności oraz do potwierdzania własności bezpieczeństwa wyrażalnych w tej logice.

Dla wybranej logiki temporalnej TL stosowanie metody BMC wymaga udowodnienia twierdzenia, które daje podstawę do weryfikacji prawdziwości formuł tej logiki w dowolnym, ale ustalonym, systemie tranzycyjnym na skończonych prefiksach ścieżek. Takie skończone prefiksy ścieżek o długości $k \geqslant 0$ nazywane są k-ścieżkami. Wspomniane twierdzenie orzeka, iż dowolna formuła φ rozważanej logiki temporalnej TL jest prawdziwa w systemie tranzycyjnym $\mathcal{M}$ wtedy i tylko wtedy, gdy istnieje liczba naturalna k, taka że formuła zdaniowa $[\mathcal{M}, \varphi]_k$ będąca koniunkcją formuły kodującej skończony zbiór k-ścieżek o mocy $f_k(\varphi)$ oraz formuły będącej translacją formuły φ jest spełnialna. Funkcja f_k, której postać zależy od konkretnej logiki temporalnej, wyznacza minimalną liczbę k-ścieżek wystarczających, niezależnie od systemu tranzycyjnego, do weryfikacji danej formuły.

Przytoczone powyżej twierdzenie uzasadnia poprawność standardowego algorytmu BMC. Zaczynając od $k = 0$, algorytm ten, dla danego systemu tranzycyjnego $\mathcal{M}$ oraz danej formuły φ, tworzy formułę zdaniową $[\mathcal{M}, \varphi]_k$. Następnie przekształca ją do równospełnialnej formuły zdaniowej w koniunkcyjnej postaci normalnej i przekazuje do sprawdzenia przez SAT-tester (ang. SAT-solver). Jeżeli SAT-tester stwierdzi niespełnialność badanej formuły, to k jest zwiększane (zwykle o 1) i proces zostaje powtórzony. Algorytm BMC kończy się wykonywać w przypadku, gdy dla pewnego k, formuła $[\mathcal{M}, \varphi]_k$ okazuje się być spełnialną lub gdy k stanie się większe od pewnego, zależnego od systemu tranzycyjnego $\mathcal{M}$, ograniczenia (np. od liczby stanów systemu tranzycyjnego $\mathcal{M}$). Przekroczenie tego ograniczenia oznacza, że formuła φ nie jest prawdziwa w systemie tranzycyjnym $\mathcal{M}$. Natomiast spełnialność formuły $[\mathcal{M}, \varphi]_k$ dla pewnego k oznacza, iż formuła φ jest prawdziwa w systemie tranzycyjnym $\mathcal{M}$, a ponadto znalezione przez SAT-tester wartościowanie pozwala na wyznaczenie świadka, którym jest zbiór k-ścieżek.

Zauważmy, że algorytm BMC kończy pracę również wtedy, gdy dla pewnego k, dostępne zasoby (pamięć lub czas) nie wystarczają do wygenerowania formuły $[\mathcal{M}, \varphi]_k$ lub nie są wystarczające dla SAT-testera. W tym przypadku oznacza to, że ze względu na ograniczoność dostępnych zasobów, algorytm BMC nie jest w stanie sprawdzić, czy własność wyrażona formułą φ jest prawdziwa w systemie tranzycyjnym $\mathcal{M}$.

W pracach, które zapoczątkowały metodę ograniczonej weryfikacji modelowej, do specyfikacji własności systemów współbieżnych stosowana była liniowa logika temporalna, w przypadku której do weryfikacji dowolnej formuły potrzebna jest tylko jedna k-ścieżka. Także w późniejszych pracach rozwijających oraz doskonalących różne aspekty metody BMC [D8, D14, D15] jako języka specyfikacji własności systemów używano języka liniowej logiki temporalnej.

W 2001 roku profesor Wojciech Penczek oraz jego ówczesna doktorantka Bożena Woźna rozpoczęli opracowywanie metody BMC dla egzystencjalnego fragmentu temporalnej logiki

czasu rozgałęzionego (ang. existential computation tree logic, w skrócie ECTL). Istotne różnice pomiędzy metodą BMC dla logiki LTL oraz metodą BMC dla logiki ECTL wynikają w szczególności z tego, iż w przypadku logiki ECTL liczba ścieżek wymaganych do weryfikacji danej formuły zależy od liczby i sposobu zagnieżdżenia operatorów temporalnych w tej formule. W trakcie prac nad opracowywaniem metody BMC dla logiki ECTL i elementarnych sieci Petriego brałem udział w rozwiązywaniu zagadnień związanych z kodowaniem relacji przejścia oraz tworzyłem implementację dla tej metody. Opis opracowanej metody oraz uzyskane przy jej użyciu wyniki eksperymentalne zostały zawarte w pracy [D33].

Jednym z kolejnych tematów podjętych przez grupę profesora Wojciecha Penczka było rozwiązanie problemu osiągalności, w systemach tranzycyjnych dla automatów czasowych oraz sieci automatów czasowych, wykorzystujące redukcję do problemu spełnialności. Wynikiem wspólnej pracy nad powyższym tematem były dwa artykuły: [D34] oraz [D41]. W trakcie prac nad wymienionymi artykułami brałem udział w rozwiązywaniu zagadnień związanych z kodowaniem relacji przejścia dla automatów czasowych (artykuł [D34]) oraz dla sieci automatów czasowych (artykuł [D41]). Ponadto, moim istotnym wkładem do obu tych prac było zaimplementowanie modułu, który po kolejnych ulepszeniach i modyfikacjach stał się częścią rozwijanego w IPI PAN systemu VerICS. W module tym zostały zaimplementowane, zdefiniowane w pracy [D1], zredukowane układy Boolowskie (ang. reduced Boolean circuits, w skrócie RBC), które zapewniają bardzo efektywną reprezentację formuł zdaniowych. Tę wysoką efektywność uzyskuje się dzięki temu, że RBC dla danej formuły zdaniowej jest acyklicznym grafem skierowanym, w którym każda podformuła jest reprezentowana przez dokładnie jeden wierzchołek, niezależnie od liczby jej wystąpień w danej formule.

Podjęte w roku 2001 badania były przeze mnie kontynuowane w następnych latach. Te z nich, których rezultatem są artykuły [H2–H6], były prowadzone w ramach grantu Ministerstwa Nauki i Informatyzacji nr 3 T11C 011 28 zatytułowanego „Automatyczna symboliczna weryfikacja programów i protokołów kryptograficznych w systemach rozproszonych”. Grant ten, którego byłem jednym z wykonawców, realizowany był w IPI PAN w okresie od 25.05.2005 r. do 31.12.2008 r.

Natomiast badania, których rezultatem są artykuły [H7, H8], były prowadzone w ramach grantu Narodowego Centrum Nauki nr 2011/01/B/ST6/05317 zatytułowanego „Opracowanie oraz implementacja metod weryfikacji modelowej dla systemów czasu rzeczywistego i wieloagentowych”. Grant ten, którego jestem jednym z głównych wykonawców, realizowany jest w IMI AJD od 08.12.2011 r., a jego zakończenie zaplanowane jest na 07.12.2014 r.

Omówienie artykułów wybranych do osiągnięcia habilitacyjnego

Do cyklu będącego podstawą mojego wniosku habilitacyjnego wybrałem te publikacje, w których mój szacunkowy udział wyniósł co najmniej 50% i które stanowią istotny wkład w rozwój metod i algorytmów ograniczonej weryfikacji modelowej dla systemów współbieżnych, jak również w rozszerzenie zakresu stosowalności rozważanej metody.

Omawiany cykl rozpoczyna artykuł [H1], w którym przedstawiłem sposoby zwiększenia efektywności wprowadzonego w pracy [D41] algorytmu BMC służącego do rozpoznawania osiągalności w systemach tranzycyjnych dla automatów czasowych lub sieci automatów czasowych. Ponadto, zmodyfikowałem ten algorytm tak, aby potrafił on w określonych przypadkach rozstrzygać, a nie tylko rozpoznawać, problem osiągalności. Zwiększenie efektywności rozważanego algorytmu zostało uzyskane poprzez zastosowanie udoskonalonej dyskretyzacji, nowe, znacznie efektywniejsze niż w pracy [D41], zakodowanie relacji przejścia czasowego oraz wyodrębnienie z przejścia akcyjnego tak zwanego przejścia dostosowawczego.

Udoskonalenie dyskretyzacji polegało na zmianie kroku dyskretyzacji z $d = \frac{1}{2 \cdot n}$ na $d = \frac{1}{m}$, gdzie n jest liczbą zegarów automatu czasowego, a m jest najmniejszą potęgą dwójki, taką że $2 \cdot n \leqslant m$. Taki krok dyskretyzacji pozwala na to, aby przyrost czasu w przejściu czasowym w systemie tranzycyjnym z semantyką dyskretną mógł być dowolną wielokrotnością liczby d mniejszą od powiększonej o jeden największej stałej występującej w warunkach zegarowych. W konsekwencji, pozwoliło to na znacznie efektywniejsze zakodowanie relacji przejścia czasowego. Natomiast wyodrębnienie z przejścia akcyjnego tak zwanego przejścia dostosowawczego sprawiło, że otrzymywane formuły Boolowskie okazały się być, pomimo zwiększenia długości świadka, znacznie łatwiejsze dla SAT-testera w procesie rozstrzygania ich spełnialności.

Efektywność algorytmu BMC dla sieci automatów czasowych może być scharakteryzowana ze względu na takie parametry sieci, jak liczba jej komponentów oraz wartości stałych występujących w warunkach zegarowych. Przeprowadzone eksperymenty potwierdziły, że zgodnie z przewidywaniami, zaproponowane sposoby istotnie zwiększyły efektywność rozważanego algorytmu. W przypadku użytego do testowania modelu, którym był protokół Fischera dla problemu wzajemnego wykluczania, o rząd wielkości ze względu na liczbę komponentów oraz o kilka rzędów wielkości ze względu na stałe występujące w warunkach zegarowych.

Udowodnione w omawianym artykule główne twierdzenie orzeka, iż problem osiągalności w systemie tranzycyjnym z semantyką gęstą dla danego automatu czasowego jest równoważny problemowi osiągalności w systemie tranzycyjnym z semantyką dyskretną dla tego automatu czasowego. Dowód tego twierdzenia został przeprowadzony bez użycia tzw. grafu regionów.

Jak wiadomo, standardowy algorytm BMC służy tylko do rozpoznawania osiągalności. Jeżeli algorytm ten stwierdzi osiągalność w systemie tranzycyjnym dla danego modelu jakiegokolwiek niepożądanego stanu globalnego, to oznacza to, że przy pomocy tego algorytmu w rozważanym modelu został znaleziony błąd. Przykładowo, w systemie tranzycyjnym dla sieci automatów realizującej protokół Fischera dla problemu wzajemnego wykluczania osiągalne będą stany naruszające własność wzajemnego wykluczania, o ile stałe występujące w warunkach zegarowych zostaną niewłaściwie dobrane. Jeżeli natomiast w modelu nie ma błędu, a zatem w systemie tranzycyjnym dla tego modelu nie jest osiągalny żaden stan naruszający własność wzajemnego wykluczania, to tego faktu nie da się stwierdzić przy pomocy standardowego algorytmu BMC.

W związku z omówionym powyżej ograniczeniem standardowego algorytmu BMC, w oma-

wianym artykule zostało zaproponowane takie jego rozszerzenie, które pozwala – chociaż tylko dla modeli spełniających pewną własność – na rozstrzyganie problemu osiągalności. Oznacza to, że rozszerzony algorytm BMC umożliwia stwierdzenie, iż żaden stan z rozważanego zbioru niepożądanych stanów globalnych nie jest osiągalny. W szczególności, dla protokołu Fischera z właściwie dobranymi stałymi w warunkach zegarowych, algorytm ten potrafi stwierdzić, iż zachowany jest warunek wzajemnego wykluczania. Wspomnianą własnością systemu tranzycyjnego umożliwiającą rozstrzyganie problemu osiągalności jest nieistnienie cykli w zbiorze tych stanów nieosiągalnych, z których osiągalny jest testowany zbiór stanów niepożądanych. W przypadku, gdy we wspomnianym zbiorze cykle istnieją, rozszerzony algorytm BMC działa tak jak standardowy algorytm BMC: potrafi tylko stwierdzić osiągalność niepożądanego stanu, o ile jakiś niepożądany stan jest faktycznie osiągalny w danym modelu.

Po złożeniu omawianego artykułu do druku wygłosiłem na seminarium grupy realizującej grant europejski „Advanced methods for timed systems" („Ametist") referat prezentujący omówione powyżej rezultaty. Na seminarium to, które odbyło się w grudniu 2003 roku w Monachium, zostaliśmy zaproszeni wraz z prof. Wojciechem Penczkiem przez kierownika grantu „Ametist" prof. Fritsa Vaandragera.

Zaproponowane w omówionej pracy sposoby zwiększenia efektywności algorytmu BMC oraz jego rozszerzenie zostały przeze mnie zaimplementowane, a następnie dołączone do systemu VerICS jako moduł BMC4DFTA.

Kontynuacją i rozszerzeniem badań podjętych w omówionym powyżej artykule [H1] jest artykuł [H2] dotyczący rozpoznawania osiągalności w systemach tranzycyjnych dla automatów czasowych z warunkami diagonalnymi, jak i sieci automatów czasowych z warunkami diagonalnymi. Warunki diagonalne pozwalają na porównywanie różnicy dwóch zegarów ze stałą będącą liczbą naturalną. Aby umożliwić zastosowanie metody rozpoznawania, jak i rozstrzygania osiągalności, w omawianej pracy została wprowadzona nowa dyskretyzacja, w której krok dyskretyzacji jest uzależniony od numeru przejścia czasowego na danej k-ścieżce. Mianowicie, dla przejścia czasowego o numerze m, gdzie $1 \leqslant m \leqslant \lfloor \frac{k+1}{2} \rfloor$, krok dyskretyzacji wynosi $d = \frac{1}{2^m}$. Oznacza to, że przyrost czasu w przejściu czasowym w systemie tranzycyjnym z semantyką dyskretną może być dowolną wielokrotnością liczby d mniejszą od powiększonej o jeden największej stałej występującej w warunkach zegarowych. Ponadto, zaproponowana dyskretyzacja pozwala wyeliminować potrzebę stosowania przejść dostosowawczych.

Udowodnione w pracy główne twierdzenie orzeka, iż problem osiągalności w systemie tranzycyjnym z semantyką gęstą dla danego automatu czasowego jest równoważny problemowi osiągalności w systemie tranzycyjnym z semantyką dyskretną dla tego automatu czasowego. Analogicznie jak w przypadku automatów czasowych bez warunków diagonalnych, zaproponowany algorytm pozwala rozstrzygać problem osiągalności, o ile nie istnieją cykle w zbiorze stanów nieosiągalnych, z których osiągalny jest testowany zbiór stanów niepożądanych.

Wprowadzona metoda została przeze mnie zaimplementowana, a przeprowadzone eksperymenty potwierdziły jej efektywność. Do eksperymentów został użyty, zmodyfikowany przeze

mnie na potrzeby omawianej pracy protokół Fischera, w którym warunki zegarowe, będące warunkami umożliwienia tranzycji akcyjnych, są warunkami diagonalnymi. Ponadto, zaimplementowana metoda została przetestowana dla automatu czasowego wykorzystanego w pracy [D10] przez Patrycję Bouyer. Za pomocą tego automatu wykazała ona, iż ówczesne algorytmy zastosowane w weryfikatorach Kronos i Uppal do testowania osiągalności w systemach tranzycyjnych dla automatów czasowych z warunkami diagonalnymi działają niepoprawnie. Jak należało oczekiwać, lokacja, która według wspomnianych weryfikatorów była identyfikowana jako nieosiągalna, została poprawnie zidentyfikowana przez metodę BMC jako osiągalna.

Implementacja związana z omawianą pracą została również dołączona do systemu VerICS jako moduł BMC4DTA.

Na bazie wyników uzyskanych w pracy [H2] oraz w pracy [D26] powstał kolejny artykuł poświęcony ograniczonej weryfikacji modelowej dla sieci automatów czasowych z warunkami diagonalnymi [H3]. W artykule tym wspólnie z dr Bożeną Woźną rozszerzyliśmy metodę BMC dla automatów czasowych z warunkami diagonalnymi oraz egzystencjalnego fragmentu czasowej logiki czasu rozgałęzionego bez operatora $\mathbf{EG}$ (TECTL-G). W tym celu zaproponowaliśmy dla automatów czasowych z warunkami diagonalnymi system tranzycyjny w postaci nieskończonego grafu regionów. Graf ten zdefiniowany jest w oparciu o wprowadzoną w pracy [H2] relację słabej równoważności, która określona jest w zbiorze wartościowań zegarów. O tym systemie tranzycyjnym udowodniliśmy twierdzenie mówiące, że zachowuje on logikę TECTL-G. Natomiast dzięki dyskretyzacji wprowadzonej w pracy [H2] można było symbolicznie (to znaczy przy pomocy formuł Boolowskich) zakodować relację przejścia tego systemu tranzycyjnego, co umożliwiło zastosowanie metody BMC do weryfikacji formuł rozważanej logiki.

W omawianej pracy zdefiniowaliśmy również wzbogacony nieskończony graf regionów dla automatów czasowych. Graf ten zachowuje fragment logiki TECTL-G, w którym nie pozwala się na zagnieżdżenie operatorów temporalnych. Skutkuje to zwiększeniem efektywności metody BMC dla tych formuł, których prawdziwość da się wykazać na ścieżkach zawierających wielokrotnie po sobie następujące przejścia czasowe.

W kolejnym artykule [H4] z omawianego cyklu podjęliśmy wspólnie z dr Agatą Półrolą problem rozpoznawania i rozstrzygania problemu osiągalności dla automatów czasowych wzbogaconych o zmienne dyskretne, to jest zmienne, których zakresem jest skończony podzbiór zbioru liczb całkowitych. Takie automaty nazywa się automatami ze zmiennymi dyskretnymi (ang. timed automata with discrete data, w skrócie TADD). Zmienne dyskretne mogą być używane w niezmiennikach lokacji, w warunkach umożliwienia tranzycji oraz w instrukcjach przypisania związanych z tranzycjami. Do zmiennej w instrukcjach przypisania można przypisać wyrażenie arytmetyczne, przy czym w omawianej pracy rozważane są tylko wyrażenia arytmetyczne zbudowane ze stałych, zmiennych oraz operatorów dodawania i odejmowania. Dla rozważanej klasy automatów rozszerzyliśmy, bazujący na translacji do problemu spełnialności, algorytm rozstrzygania osiągalności oraz wykazaliśmy jego poprawność. Wyniki eksperymentalne, uzyskane z wykorzystaniem stworzonej przeze mnie implementacji, potwierdziły praktyczną

stosowalność metody.

Ograniczenie klasy wyrażeń arytmetycznych poprzez dopuszczenie tylko operatorów dodawania i odejmowania istotnie zawęża klasę systemów, które można modelować przy pomocy automatów czasowych ze zmiennymi dyskretnymi. Dlatego wkrótce po zakończeniu prac nad omawianym artykułem podjąłem próbę Boolowskiego zakodowania pozostałych podstawowych operacji arytmetycznych: mnożenia, dzielenia całkowitego oraz reszty z dzielenia całkowitego. Kodowanie to opisałem w raporcie technicznym [D44]. Oprócz kodowania wspomnianych podstawowych operacji arytmetycznych w raporcie tym zostało opisane także Boolowskie kodowanie operacji całkowitego pierwiastka kwadratowego z liczby naturalnej oraz potęgowania z wykładnikiem naturalnym. Boolowskie kodowanie podstawowych operacji arytmetycznych pozwoliło rozszerzyć stworzoną na potrzeby artykułu [H4] implementację, a powstały moduł BMC4TADD został włączony do systemu VerICS.

W późniejszym okresie na bazie modułu BMC4TADD powstał moduł BMC4TADDPA, w którym dodana została możliwość używania parametrów całkowitych oraz parametrycznych przypisań. Zastosowania tego modułu zostały opisane w pracach [D21, D28, D32], których jestem współautorem.

Opracowanie i zaimplementowanie metody testowania osiągalności dla automatów czasowych ze zmiennymi dyskretnymi pozwoliło na rozważenie jej wykorzystania do weryfikacji programów współbieżnych napisanych w wybranym niewielkim podzbiorze języka Java. W artykule [H5] wspólnie z dr Bożeną Woźną zaproponowaliśmy metodę modelowania wybranych konstrukcji oraz mechanizmów współbieżności języka Java przy pomocy sieci automatów czasowych ze zmiennymi dyskretnymi. Modelowany podzbiór języka Java obejmuje definicje zmiennych całkowitych, standardowe konstrukcje, takie jak instrukcje przypisania, instrukcje warunkowe i iteracyjne, definicje klas i obiektów, definicje metod synchronizowanych oraz metody biblioteczne używane w programowaniu wielowątkowym: `wait()`, `notify()`, `sleep()` i `random()`. Wielowątkowy program w Javie jest modelowany za pomocą sieci automatów czasowych ze zmiennymi dyskretnymi, przy czym każdy wątek jest modelowany przez jeden automat. Stany automatu są abstrakcją stanów wątku, a tranzycje są abstrakcją jego instrukcji. Każda z metod `wait()`, `notify()` oraz `random()` jest modelowana przez odpowiedni automat, natomiast do modelowania metody `sleep()` używany jest zegar występujący w tym automacie czasowym, który został użyty do modelowania wątku zawierającego wywołanie metody `sleep()`.

Omawiana praca dała podwaliny do stworzenia translatora dla wybranego podzbioru języka Java. Program napisany w tym podzbiorze jest przekształcany na sieć automatów czasowych ze zmiennymi dyskretnymi co umożliwia zastosowanie modułu BMC4TADD do testowania takich błędów programu jak: sytuacja wyścigu, brak wzajemnego wykluczania oraz istnienie możliwości blokady. Dodatkowo, stworzony translator umożliwia wygenerowanie sieci automatów czasowych ze zmiennymi dyskretnymi w formacie stosowanym w weryfikatorze UPPAL [D5, D25]. Konstrukcja i możliwości translatora zostały opisane w pracy [D36].

W artykule [H6] przedstawiłem sposób istotnego zwiększenia efektywności metody ograniczonej weryfikacji modelowej dla logiki ECTL poprzez ulepszenie translacji problemu prawdziwości formuł tej logiki w danym modelu do problemu SAT. Pierwsza translacja z ECTL do SAT została opisana w pracy [D33]. Źródłem względnej nieefektywności tej translacji jest fakt, że do translacji danej formuły φ, jak również każdej jej podformuły ψ, używa się wszystkich ścieżek symbolicznych o indeksach ze zbioru $F_k(\varphi) = \{1, \ldots, f_k(\varphi)\}$. Podstawą ulepszenia translacji z omawianego artykułu [H6] jest naturalna idea, aby do translacji danej podformuły ψ użyć tylko pewnego podzbioru o mocy $f_k(\psi)$ zbioru $F_k(\varphi)$.

Idea ta została zrealizowana poprzez wprowadzoną metodę wyznaczania w zbiorze $F_k(\varphi)$ podzbiorów potrzebnych do translacji właściwych podformuł formuły φ. W szczególności, do translacji podformuły postaci $\mathbf{EO}\psi$, gdzie $\mathbf{O} \in \{\mathbf{X}, \mathbf{G}, \mathbf{U}\}$ używana jest jedna ścieżka do translacji operatora $\mathbf{EO}$ oraz $f_k(\psi)$ innych ścieżek do translacji formuły ψ. Zauważmy, że translacja z pracy [D33] używa zawsze $f_k(\varphi)$ ścieżek do translacji dowolnej podformuły postaci $\mathbf{EO}\psi$.

W omawianym artykule wykazano poprawność i zupełność zmodyfikowanej translacji. Jak należało oczekiwać, przeprowadzone eksperymenty (dla sieci automatów modelującej problem ucztujących filozofów) wykazały, że nowa translacja jest znacznie efektywniejsza od poprzedniej translacji. W zależności od badanej formuły czas translacji jest krótszy nawet o trzy rzędy wielkości, a czas zużyty przez SAT-tester krótszy o dwa rzędy wielkości. Także wymagania pamięciowe dla procesu translacji, jak również dla SAT-testera, uległy kilkakrotnemu zmniejszeniu. Wykonana przez mnie na potrzeby omawianej pracy implementacja została włączona do systemu VerICS jako moduł BMC4ECTL.

Powyższa translacja z ECTL do SAT, a także związana z nią implementacja, zostały wykorzystane i rozszerzone w pracach [D23] oraz [D42].

W pracy [D23] rozszerzono translację z ECTL do SAT do translacji z logiki PRTECTL, która jest parametrycznym rozszerzeniem egzystencjalnego fragmentu logiki CTL. Bazująca na tej translacji metoda BMC została zastosowana do weryfikacji własności czasowych sieci Petriego z semantyką dyskretną. Użyty do eksperymentów moduł o nazwie BMC4PRTECTL powstał na bazie wcześniej zaimplementowanych przeze mnie modułów BMC4ECTL oraz BMC4DTPN.

W pracy [D42], której jestem współautorem, wprowadzona przeze mnie translacja z ECTL do SAT została rozszerzona dla logiki RTECTLK, w której występują operatory temporalne z przedziałami oraz operatory wiedzowe. Wyniki eksperymentalne zostały wykonane przy pomocy zaimplementowanego przeze mnie modułu BMC4RTECTLK, który jest rozszerzeniem modułu BMC4ECTL.

W artykule [H7] przedstawiłem sposób istotnego zwiększenia efektywności metody ograniczonej weryfikacji modelowej dla logiki ECTL* poprzez ulepszenie translacji problemu prawdziwości formuł tej logiki w danym modelu do problemu SAT. Pierwsza poprawna translacja z ECTL* do SAT została opisana w pracy [D40]. Podobnie jak w przypadku translacji z ECTL do SAT, źródłem znacznej nieefektywności translacji z pracy [D40] jest fakt, że do translacji

danej formuły φ, jak również każdej jej podformuły ψ, używa się wszystkich ścieżek symbolicznych o indeksach ze zbioru $F_k(\varphi) = \{1, \ldots, f_k(\varphi)\}$. Podstawą artykułu [H7] są: nowe kodowanie ścieżek symbolicznych oraz idea, aby do translacji dowolnej właściwej podformuły ψ formuły φ użyć tylko pewnego podzbioru o mocy $f_k(\psi)$ zbioru $F_k(\varphi)$.

We wspomnianym nowym kodowaniu ścieżka symboliczna o długości k jest parą uporządkowaną, której pierwsza składowa jest ciągiem stanów symbolicznych, a druga składowa jest symboliczną liczbą naturalną l, taką że $0 \leqslant l \leqslant k$. Takie kodowanie pozwala na zdefiniowanie translacji w sposób, który dopuszcza aby w zbiorze ścieżek będącym świadkiem dla danej formuły pewne ścieżki były k-pętlami, a pewne inne nie. Natomiast translacja z pracy [D40] wymusza, aby w zbiorze ścieżek będącym świadkiem dla danej formuły albo wszystkie ścieżki były k-pętlami albo wszystkie ścieżki nie były k-pętlami.

Idea, aby do translacji właściwych podformuł danej formuły używać właściwych podzbiorów zbioru $F_k(\varphi)$ została zrealizowana poprzez wyznaczanie w zbiorze $F_k(\varphi)$ podzbiorów potrzebnych do translacji właściwych podformuł formuły φ. W szczególności, do translacji podformuły postaci $\mathbf{E}\psi$ używana jest jedna ścieżka do translacji operatora $\mathbf{E}$ oraz $f_k(\psi)$ innych ścieżek do translacji formuły ψ. Zauważmy, że translacja z pracy [D40] używa zawsze $f_k(\varphi)$ ścieżek do translacji dowolnej podformuły postaci $\mathbf{E}\psi$.

Wykonane przez mnie na potrzeby omawianego artykułu dwie implementacje, jedna dla translacji z pracy [D40] oraz druga dla translacji z pracy [H7], pozwoliły na przeprowadzenie eksperymentów, które wykazały, że nowa translacja jest zdecydowanie bardziej efektywna. W zależności od badanej formuły czas translacji jest krótszy o dwa rzędy wielkości, a czas zużyty przez SAT-tester krótszy o co najmniej jeden rząd wielkości. Także wymagania pamięciowe dla procesu translacji, jak również dla SAT-testera, uległy kilkakrotnemu zmniejszeniu.

Ponadto warto zauważyć, że każda translacja z ECTL* do SAT może być zastosowana do formuł logiki ECTL. Przeprowadzone wyniki eksperymentalne pokazały dla formuł, w których nie występuje operator $\mathbf{EX}$, pewien wzrost efektywności w stosunku do translacji zdefiniowanej tylko dla logiki ECTL, co spowodowane jest zapewne nowym kodowaniem ścieżek. Natomiast dla formuł logiki ECTL, które zawierają operator $\mathbf{EX}$, translacja z ECTL* do SAT daje w wyniku formuły zdaniowe trudniejsze do rozstrzygnięcia przez SAT-solver. Jest to spowodowane tym, że w translacji z ECTL do SAT nie trzeba wymagać, aby k-ścieżka wybrana do translacji operatora $\mathbf{EX}$ była k-pętlą, natomiast w przypadku translacji z ECTL* do SAT nie można zrezygnować z tego wymagania.

Opisana translacja z ECTL* do SAT została zastosowana i rozszerzona w pracy [D43] do translacji z logiki EMTLKD oraz deontycznych przeplotowych systemów interpretowanych. Logika EMTLKD jest egzystencjalnym fragmentem metrycznej logiki temporalnej wzbogaconym o operatory wiedzowe i operatory deontyczne.

W artykule [H8] razem dr Bożeną Woźną-Szcześniak zaproponowaliśmy redukcję problemu egzystencjalnej weryfikacji modelowej dla logiki MITL (ang. Metric Interval Temporal Logic) z semantyką gęstą do problemu ograniczonej weryfikacji modelowej dla języka

HLTLz semantyką dyskretną. Logika MITL jest fragmentem logiki MTL, która rozszerza liniową logikę temporalną poprzez nałożenie na operatory temporalne ograniczeń czasowych w postaci, ograniczonych lub nieograniczonych, przedziałów liczb rzeczywistych, przy czym w logice MITL dopuszcza się tylko przedziały niezdegenerowane. Od czasu wprowadzenia logiki MTL w pracy [D24] oraz logiki MITL w artykule [D3] stały się one, jak również problem weryfikacji modelowej dla tych logik, przedmiotem rozlicznych badań – przykładowo w pracach [D11–D13, D29, D30].

Wprowadzona w omawianym artykule logika HLTL jest wariantem liniowej logiki temporalnej, w którym operator $\mathbf{X}$ (to jest operator następnego kroku) został zastąpiony rodziną indeksowanych operatorów resetujących $\mathbf{H}_k$, dla $k \in \mathbb{N}$. Rozważana redukcja zrealizowana jest poprzez translację dowolnej formuły MITL do formuły HLTL poprzez zastąpienie każdego wystąpienia operatora temporalnego $\mathbf{O}_{I_k}$ (gdzie $\mathbf{O}_{I_k} \in \{\mathbf{U}_{I_k}, \mathbf{R}_{I_k}\}$) odpowiednim złożeniem operatorów $\mathbf{H}_k$ oraz $\mathbf{O}$, przy czym liczba operatorów $\mathbf{H}_k$ w wynikowej formule jest równa liczbie wystąpień operatorów temporalnych $\mathbf{U}_{I_k}$ oraz $\mathbf{R}_{I_k}$ w formule poddawanej translacji. Celem nowego operatora jest „wyzerowanie" wartości zegara związanego z danym przedziałem I_k. Jest to zapewnione przez semantykę operatora $\mathbf{H}_k$ i służy właściwemu odmierzaniu czasu od odpowiednich punktów na dowolnej, rozważanej ścieżce wykonania w modelu dyskretnym dla automatu czasowego.

W artykule dowodzimy, że wprowadzona translacja jest poprawna w tym sensie, iż dowolna formuła MITL jest prawdziwa w systemie tranzycyjnym z semantyką gęstą dla danego automatu czasowego wtedy i tylko wtedy, gdy odpowiadająca jej formuła HLTL (formuła po translacji) jest prawdziwa w systemie tranzycyjnym z semantyką dyskretną dla tego automatu. Udowodnione twierdzenie jest podstawą do zastosowania metody BMC do weryfikacji własności wyrażonych w MITL poprzez ich translację do formuł HLTL z następującym zastosowaniem metody BMC do formuł logiki HLTL poprzez odpowiednie rozszerzenie standardowej metody BMC dla liniowej logiki temporalnej.

Podsumowanie

Jednym z najważniejszych praktycznych problemów weryfikacji modelowej jest wykładniczy wzrost, zależącej od liczby komponentów modelowanego systemu, liczby stanów systemu tranzycyjnego. Edmund M. Clarke, współtwórca metody weryfikacji modelowej, wielokrotnie podkreślał [D16–D18], iż problem przezwyciężenia wykładniczej eksplozji przestrzeni stanów był jednym z najważniejszych problemów badawczych jakimi zajmował się od momentu powstania weryfikacji modelowej. Przezwyciężenie problemu wykładniczej eksplozji przestrzeni stanów wymaga w szczególności, aby stosowane metody i algorytmy charakteryzowały się jak największą efektywnością. Omawiając artykuły, które zaliczyłem do mojego osiągnięcia habilitacyjnego, przedstawiłem swój wkład zarówno w zwiększanie efektywności metod ograniczonej weryfikacji modelowej, jak również w rozszerzenie stosowalności ograniczonej weryfikacji modelowej do różnych klas systemów współbieżnych.

W artykułach [H1, H6, H7] rozwinąłem i znacząco udoskonaliłem istniejące już metody BMC, dzięki czemu istotnie zwiększył się zakres ich praktycznej stosowalności. W artykule [H2] rozszerzyłem metodę BMC rozstrzygania problemu osiągalności dla automatów z warunkami diagonalnymi. W artykułach [H4, H5], wspólnie z ich współautorkami, zaproponowałem i rozwinąłem metody pozwalające na stosowanie ograniczonej weryfikacji modelowej dla automatów czasowych ze zmiennymi dyskretnymi oraz dla programów wielowątkowych w Javie. Natomiast w artykułach [H3, H8], wspólnie z ich współautorką, zaproponowałem metody ograniczonej weryfikacji modelowej dla logik TECTL-G oraz MITL.

Każda z zaproponowanych metod ograniczonej weryfikacji modelowej została zaimplementowana przeze mnie w języku C++ jako niezależny moduł. Moduły zaimplementowane przed 2010 rokiem zostały dołączone do systemu VerICS, który jest narzędziem służącym do weryfikacji poprawności systemów czasu rzeczywistego, systemów wieloagentowych oraz systemów specyfikowanych w językach takich jak Estelle, Promela, Java i UML. System VerICS jest rozwijany pod kierunkiem profesora Wojciecha Penczka w Instytucie Podstaw Informatyki PAN w Warszawie. Moduły weryfikacyjne systemu VerICS stosują metodę ograniczonej weryfikacji modelowej, która wykorzystuje translację do problemu spełnialności. W roku 2010 system VerICS został laureatem konkursu JAKOŚĆ ROKU.

Do systemu VerICS zostały także dołączone, zaimplementowane przeze mnie, moduły związane z powstałymi metodami ograniczonej weryfikacji modelowej, które nie zostały opisane w artykułach zaliczonych przeze mnie do mojego osiągnięcia habilitacyjnego. Metody te zostały natomiast opisane w innych pracach, których jestem współautorem. Ponadto, w skład systemu VerICS wchodzą moduły realizujące metody ograniczonej weryfikacji modelowej opisane w pracach, których nie jestem współautorem. Większość z tych modułów jest jednak modyfikacją i rozszerzeniem modułów, których byłem jedynym wykonawcą. Moduły powstałe po 2010 roku zostaną włączone do, planowanej do realizacji w bieżącym roku, kolejnej wersji systemu VerICS.

5. Omówienie pozostałych osiągnięć naukowo - badawczych (artystycznych):

Moje pozostałe osiągnięcia naukowo-badawcze zostały wyszczególnione w załączniku nr 3: „Wykaz opublikowanych prac naukowych lub twórczych prac zawodowych oraz informacja o osiągnięciach dydaktycznych, współpracy naukowej i popularyzacji nauki”

Tutaj wymienię jedynie wybrane prace [D35, D42, D43], których byłem współautorem, i które dotyczą weryfikacji modelowej systemów wieloagentowych [D22, D38, D39]. Własności systemów wieloagentowych formułowane są w rozszerzeniach logik temporalnych zawierających operatory wiedzowe.

Prace [D42] oraz [D43] zostały już pokrótce scharakteryzowane w zasadniczej części mojego autoreferatu.

W pracy [D35] razem ze współautorami zdefiniowaliśmy metodę ograniczonej weryfikacji modelowej dla systemów wieloagentowych modelowanych przez przeplotowe systemy inter-

pretowane (ang. interleaving interpreted systems, w skrócie IIS) oraz dla egzystencjalnego fragmentu liniowej logiki temporalnej rozszerzonej o operatory wiedzowe (ELTLK). Udowodniliśmy w tym artykule równoważność semantyki ograniczonej i nieograniczonej oraz poprawność translacji problemu weryfikacji modelowej dla ELTLK do problemu SAT. Opisana w omawianym artykule metoda została zaimplementowana w postaci modułu BMC4ELTLK, który został użyty do przeprowadzenia eksperymentów. Wyniki tych eksperymentów potwierdziły praktyczną stosowalność wprowadzonej metody ograniczonej weryfikacji modelowej oraz pozwoliły wybrać jeden z trzech możliwych wariantów translacji jako najbardziej efektywny. Jednak jak się okazało, jeszcze bardziej efektywna jest translacja wykorzystująca wprowadzoną przeze mnie w artykule [D7] translację z ECTL* do SAT. Translacja ta została wykorzystana w pracy [D43].

W wymienionych pracach, których byłem współautorem, a które dotyczą ograniczonej weryfikacji modelowej systemów wieloagentowych oraz własności wyrażonych w wybranej logice temporalnej rozszerzonej o operatory wiedzowe wykorzystywana była semantyka przeplotowa, która ogranicza wykonania w systemie tranzycyjnym dla danego systemu wieloagentowego do takich, w których w jednym kroku może się wykonać tylko jedna akcja, przy czym jeśli jest to akcja wspólna dla co najmniej dwóch agentów, to musi ona zostać wykonana w danym kroku przez wszystkich agentów, w których akcja ta występuje.

Obecnie biorę udział w pracach nad artykułem, w którym wykorzystywana jest również semantyka nieprzeplotowa, nie ograniczająca wykonań systemu w opisany powyżej sposób. Jednak, analogicznie jak w semantyce przeplotowej, każda akcja, która jest wspólna dla co najmniej dwóch agentów, musi zostać wykonana w danym kroku przez wszystkich agentów, w których akcja ta występuje. Wykonana przez mnie implementacja wykorzystująca semantykę nieprzeplotową pozwoliła eksperymentalnie stwierdzić, że wykorzystująca translację do SAT ograniczona weryfikacja modelowa systemów wieloagentowych z semantyką nieprzeplotową jest znacznie efektywniejsza niż wykorzystująca translację do SAT ograniczona weryfikacja modelowa systemów wieloagentowych z semantyką nieprzeplotową. Stworzony na potrzeby wspomnianej pracy moduł zostanie również dołączony do systemu VerICS.

Literatura cytowana

[D1] P. A. Abdulla, P. Bjesse, N. Eén. Symbolic Reachability Analysis Based on SAT-Solvers. *Proc. of the 6th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'00)*, wolumen 1785 serii *LNCS*, strony 411–425. Springer-Verlag, 2000.

[D2] Rajeev Alur, David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[D3] Rajeev Alur, Tomás Feder, Thomas A. Henzinger. The Benefits of Relaxing Punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[D4] C. Baier, J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[D5] Gerd Behrmann, Alexandre David, Kim Guldstrand Larsen, Paul Pettersson, Wang Yi. Developing uppaal over 15 years. *Softw., Pract. Exper.*, 41(2):133–142, 2011.

[D6] A. Biere, A. Cimatti, E. Clarke, M.Fujita, Y. Zhu. Symbolic Model Checking Using SAT Procedures instead of BDDs. *Proc. of the ACM/IEEE Design Automation Conference (DAC'99)*, strony 317–320, 1999.

[D7] A. Biere, A. Cimatti, E. Clarke, Y. Zhu. Symbolic Model Checking without BDDs. *Proc. of the 5th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99)*, wolumen 1579 serii *LNCS*, strony 193–207. Springer-Verlag, 1999.

[D8] A. Biere, K. Heljanko, T. A. Junttila, T. Latvala, V. Schuppan. Linear Encodings of Bounded LTL Model Checking. *Logical Methods in Computer Science*, 2(5), 2006.

[D9] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, Yunshan Zhu. Bounded Model Checking. *Advances in Computers*, 58:117–148, 2003.

[D10] Patricia Bouyer. Untameable Timed Automata! *STACS*, wolumen 2607 serii *Lecture Notes in Computer Science*, strony 620–631. Springer, 2003.

[D11] Patricia Bouyer. Model-checking Timed Temporal Logics. *Electr. Notes Theor. Comput. Sci.*, 231:323–341, 2009.

[D12] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, James Worrell. The Cost of Punctuality. *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, strony 109–120. IEEE Computer Society, 2007.

[D13] Patricia Bouyer, Nicolas Markey, Joël Ouaknine, James Worrell. On Expressiveness and Complexity in Real-Time Model Checking. *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, wolumen 5126 serii *Lecture Notes in Computer Science*, strony 124–135. Springer, 2008.

[D14] E. Clarke, A. Biere, R. Raimi, Y. Zhu. Bounded Model Checking Using Satisfiability Solving. *Formal Methods in System Design*, 19(1):7–34, 2001.

[D15] E. Clarke, D. Kroening, J. Ouaknine, O. Strichman. Completeness and Complexity of Bounded Model Checking. *Proc. of the 5th Int. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI'04)*, wolumen 2937 serii *LNCS*, strony 85–96. Springer-Verlag, 2004.

[D16] Edmund M. Clarke. The birth of model checking. *25 Years of Model Checking - History, Achievements, Perspectives*, wolumen 5000 serii *Lecture Notes in Computer Science*, strony 1–26. Springer, 2008.

[D17] Edmund M. Clarke. Model checking - my 27-year quest to overcome the state explosion problem. *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings*, wolumen 5330 serii *Lecture Notes in Computer Science*, strona 182. Springer, 2008.

[D18] Edmund M. Clarke. My 27-year quest to overcome the state explosion problem. *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, strona 3, 2009.

[D19] Edmund M. Clarke, E. Allen Emerson, A. Prasad Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. *Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 1983*, strony 117–126. ACM Press, 1983.

[D20] Edmund M. Clarke, Orna Grumberg, Doron Peled. *Model Checking*. MIT Press, 2001.

[D21] Dariusz Doliwa, Wojciech Horzelski, Mariusz Jarocki, Artur Niewiadomski, Wojciech Penczek, Agata Połrola, Maciej Szreter, Andrzej Zbrzezny. PlanICS - a Web Service Composition Toolset. *Fundamenta Informaticae*, 112(1):47–71, 2011.

[D22] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[D23] Michal Knapik, Wojciech Penczek, Maciej Szreter, Agata Półrola. Bounded Parametric Verification for Distributed Time Petri Nets with Discrete-time Semantics. *Fundamenta Informaticae*, 101(1-2):9–27, 2010.

[D24] Ron Koymans. Specifying Real-time Properties with Metric Temporal Logic. *Real-Time Systems*, 2(4):255–299, 1990.

[D25] Kim Guldstrand Larsen, Paul Pettersson, Wang Yi. Uppaal in a nutshell. *STTT*, 1(1-2):134–152, 1997.

[D26] Alessio Lomuscio, Wojciech Penczek, Bożena Woźna. Bounded Model Checking for Knowledge and Real Time. *Artif. Intell.*, 171(16-17):1011–1038, 2007.

[D27] P. Merlin, D. J. Farber. Recoverability of Communication Protocols – Implication of a Theoretical Study. *IEEE Trans. on Communications*, 24(9):1036–1043, 1976.

[D28] Artur Niewiadomski, Wojciech Penczek, Agata Półrola, Maciej Szreter, Andrzej Zbrzezny. Towards Automatic Composition of Web Services: SAT-Based Concretisation of Abstract Scenario. *Fundamenta Informaticae*, 120(2):181–203, 2012.

[D29] Joël Ouaknine, James Worrell. On the Decidability and Complexity of Metric Temporal Logic over Finite Words. *Logical Methods in Computer Science*, 3(1), 2007.

[D30] Joël Ouaknine, James Worrell. Some Recent Results in Metric Temporal Logic. *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, wolumen 5215 serii *Lecture Notes in Computer Science*, strony 1–13. Springer, 2008.

[D31] W. Penczek, A. Półrola. *Advances in Verification of Time Petri Nets and Timed Automata: A Temporal Logic Approach*, wolumen 20 serii *Studies in Computational Intelligence*. Springer-Verlag, 2006.

[D32] Wojciech Penczek, Agata Półrola, Andrzej Zbrzezny. Towards Automatic Composition of Web Services: A SAT-Based Phase. *ACSD/Petri Nets Workshops*, wolumen 827 serii *CEUR Workshop Proceedings*, strony 453–473. CEUR-WS.org, 2010.

[D33] Wojciech Penczek, Bożena Woźna, Andrzej Zbrzezny. Bounded Model Checking for the Universal Fragment of CTL. *Fundamenta Informaticae*, 51(1-2):135–156, 2002.

[D34] Wojciech Penczek, Bożena Woźna, Andrzej Zbrzezny. Towards Bounded Model Checking for the Universal Fragment of TCTL. *Proc. of the 7th Int. Symp. on Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT'02)*, wolumen 2469 serii *LNCS*, strony 265–288. Springer-Verlag, 2002.

[D35] Wojciech Penczek, Bożena Woźna-Szcześniak, Andrzej Zbrzezny. Towards SAT-based BMC for LTLK over Interleaved Interpreted Systems. *Fundamenta Informaticae*, 119(3-4):373–392, 2012.

[D36] Artur Rataj, Bożena Woźna, Andrzej Zbrzezny. A Translator of Java Programs to TADDs. *Fundamenta Informaticae*, 93(1-3):305–324, 2009.

[D37] Wolfgang Reisig. *Petri Nets: An Introduction*, wolumen 4 serii *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1985.

[D38] Wiebe van der Hoek, Michael Wooldridge. Model checking knowledge and time. *Model Checking of Software, 9th International SPIN Workshop, Grenoble, France, April 11-13, 2002, Proceedings*, wolumen 2318 serii *Lecture Notes in Computer Science*, strony 95–111. Springer, 2002.

[D39] Michael J. Wooldridge. *An Introduction to MultiAgent Systems (2. ed.)*. Wiley, 2009.

[D40] B. Woźna. ACTL* Properties and Bounded Model Checking. *Fundamenta Informaticae*, 63(1):65–87, 2004.

[D41] Bożena Woźna, Andrzej Zbrzezny, Wojciech Penczek. Checking Reachability Properties for Timed Automata via SAT. *Fundamenta Informaticae*, 55(2):223–241, 2003.

[D42] Bożena Woźna-Szcześniak, Agnieszka Zbrzezny, Andrzej Zbrzezny. The BMC Method for the Existential Part of RTCTLK and Interleaved Interpreted Systems. *Progress in Artificial Intelligence, 15th Portuguese Conference on Artificial Intelligence, EPIA 2011, Lisbon, Portugal, October 10-13, 2011. Proceedings*, wolumen 7026 serii *Lecture Notes in Computer Science*, strony 551–565. Springer, 2011.

[D43] Bożena Woźna-Szcześniak, Andrzej Zbrzezny. SAT-Based BMC for Deontic Metric Temporal Logic and Deontic Interleaved Interpreted Systems. *10th International Workshop, DALT 2012, Valencia, Spain, June 4, 2012, Revised Selected and Invited Papers.*, wolumen 7784 serii *LNCS*, strony 170–189. Springer-Verlag, 2013.

[D44] Andrzej Zbrzezny. A Boolean Encoding of Arithmetic Operations. Raport instytutowy 999, ICS PAS, Ordona 21, 01-237 Warsaw, Styczeń 2007.

[D45] Wieslaw Zielonka. Notes on Finite Asynchronous Automata. *ITA*, 21(2):99–135, 1987.