

Politechnika Warszawska
Wydział Elektroniki i Technik Informatycznych

DZIEKAN I RADA WYDZIAŁU ELEKTRONIKI I TECHNIK INFORMATYCZNYCH
POLITECHNIKI WARSZAWSKIEJ

zawiadamiają o

PUBLICZNEJ OBRONIE ROZPRAWY DOKTORSKIEJ

mgr. inż. Marka Adamka

która odbędzie się w dniu 19 listopada 2019 r. o godzinie 11.00

na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej
Gmach im. Janusza Groszkowskiego, Warszawa ul. Nowowiejska 15/19, w sali nr 116

Tytuł rozprawy doktorskiej: „Rozwój aplikacji logik temporalnych w zastosowaniach informatycznych”

promotor: prof. dr hab. inż. Jan Mulawka Wydział Elektroniki i Technik
 Informatycznych Politechniki Warszawskiej

recenzenci: prof. dr hab. Wojciech Penczek Instytut Podstaw Informatyki Polskiej
 Akademii Nauk,

 prof. dr hab. Andrzej Szałas Wydział Matematyki, Informatyki i Mechaniki
 Uniwersytet Warszawski.

Na stronie internetowej wydziału www.elka.pw.edu.pl/Wydzial/Rada-Wydzialu/Harmonogram-obron-doktorskich-streszczenia-i-recenzje znajdują się streszczenie rozprawy oraz recenzje, jak również dostęp do tekstu rozprawy umieszczonej w Bazie Wiedzy Politechniki Warszawskiej.

Dziekan



prof. dr hab. inż. Krzysztof Zaremba

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH

Rozprawa doktorska

mgr inż. Marek Adamek

**Rozwój aplikacji logik temporalnych
w zastosowaniach informatycznych**

Promotor
prof. dr hab. inż. Jan Mulawka

WARSZAWA 2019

Streszczenie

Logiki temporalne są rozszerzeniem logiki klasycznej, umożliwiającym rozważanie zależności czasowych bez jawnego wprowadzania czasu. Formalizmy te są od prawie 40 lat wykorzystywane w informatyce. Jednym z nich jest logika LTL (Linear-Time Temporal Logic), która została już gruntownie przebadana, w szczególności opracowano dla niej szereg metod formalnego wnioskowania oraz metod weryfikacji modelowej. Wiele z nich ma wysokiej jakości implementacje, które są wykorzystywane w przemyśle informatycznym i telekomunikacyjnym.

Niniejsza praca poświęcona jest analizie logiki temporalnej LNC (Logic Next Change), będącej fragmentem LTL. Chociaż ogólnie ekspresywność LNC jest mniejsza niż w przypadku LTL, to jednak ciągle istnieje szerokie spektrum problemów, dla których siła wyrażania LNC jest wystarczająca. W pracy prowadzone są rozważania nad problemami badania spełnialności oraz weryfikacji modelowej w logice LNC, z których wynika, że klasy złożoności tych problemów są niższe (NP-zupełne) względem LTL (PSPACE-zupełne). Jako uzupełnienie tych rozważań zaproponowano efektywne algorytmy badania spełnialności i weryfikacji modelowej w logice LNC.

Innym zagadnieniem poruszonym w tej pracy jest związek pomiędzy formułami LNC a pewną klasą struktur Kripkego. Formuły te można interpretować jako modele poddawane weryfikacji w metodzie weryfikacji modelowej. Dla takiej reprezentacji istnieje efektywna i symboliczna metoda weryfikacji modelowej wykorzystująca redukcję do problemu badania spełnialności w LTL.

Oprócz rozważań teoretycznych praca zawiera wyniki przeprowadzonej weryfikacji eksperymentalnej pod kątem wydajności i poprawności zaproponowanych algorytmów (implementację wykonano w języku Scala). W znacznej większości przypadków opracowane metody uzyskały dużo lepsze wyniki w porównywaniu z dostępnymi obecnie narzędziami dedykowanymi logice LTL. Wybrane narzędzia implementują różne techniki badania spełnialności i weryfikacji modelowej.

Praca prezentuje również możliwe, praktyczne zastosowania logiki LNC — jak przykładowo: język wykonywalnej specyfikacji wymagań, metoda planowania czy generacja, weryfikacja i korekcja ciągów binarnych — a także perspektywy dalszych badań w zakresie logiki LNC i opracowanych dla niej metod. Oprócz części zasadniczej, w pracy zamieszczono dwa Dodatki zawierające szczegółowy opis implementacji logiki LNC w języku Scala oraz przykłady wykorzystania tej implementacji do rozwiązywania wybranych problemów.

Słowa kluczowe:

logika, logika temporalna, LTL, LNC, spełnialność formuł logicznych, SAT, weryfikacja modelowa, MC, struktura Kripkego

Abstract

Temporal logic systems are extensions of classical logic calculus, which allow to express time related dependencies where time not given explicitly. Such logics have been used in computer science over the last 40 years. One of them — which is highlighted in the dissertation is LTL (Linear-Time Temporal Logic) — which have already been deeply studied. In particular, there is a number of methods developed for LTL-based formal inference and model checking. Some of them have high quality implementations which are used in computer science and telecommunication industry.

The main focus of the dissertation is on temporal logic LNC (Logic Next Change), which is a LTL fragment. Expressiveness of LNC is not as wide in comparison to LTL, however there still exists a significant number of problems for which LNC expressiveness is sufficient. The work provides analysis of boolean satisfiability and model checking problems in context of LNC. The conclusion is that the complexity of these problems (NP-complete) is lower in comparison to LTL (PSPACE-complete). In addition to theoretical discussion efficient algorithms for boolean satisfiability and model checking problems has been designed and implemented.

Another interesting feature of LNC logic discussed in this dissertation is the relationship between LNC formulas and a subclass of Kripke structures. A LNC formula can be interpreted as a model under verification in model checking method. Such representation can be used for implementation of efficient and symbolic LTL model checking method based on reduction to LTL boolean satisfiability problem.

To support claims stated in the thesis an experimental verification has been conducted. The tests were focused on efficiency and correctness of proposed algorithms and methods. In great number of cases these methods present much better performance in comparison to existing tools used for LTL. These tools were selected to cover wide range of different methods for boolean satisfiability and model checking problems.

The work also discusses possible practical applications of LNC logic — for example: executable specification language, planning or generation, verification and correction of binary streams — and indicates directions for further research in context of LNC and developed methods. In addition to the main part of the dissertation there are two Appendices which present implementation details of LNC in Scala and its usage for solving selected problems.

Key words:

logic, temporal logic, LTL, LNC, boolean satisfiability problem, SAT, model checking, MC, Kripke structure

RECENZJA ROZPRAWY DOKTORSKIEJ
DLA RADY WYDZIAŁU ELEKTRONIKI I TECHNIK INFORMACYJNYCH
POLITECHNIKI WARSZAWSKIEJ

Autor recenzji: prof. dr hab. Andrzej Szalas

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

1 września 2019

AUTOR ROZPRAWY: mgr inż. Marek Adamek

TYTUŁ ROZPRAWY: „Rozwój aplikacji logik temporalnych w zastosowaniach informatycznych”

0. WPROWADZENIE

Przedmiotem recenzji jest druga wersja rozprawy (z 2019r.), znacznie poprawiona i uzupełniona w stosunku do pierwszej. Podejmuje ona zagadnienia informatycznych zastosowań logiki temporalnej LNC (*Logic Next Change*),¹ w której czas jest dyskretny i liniowy. Rozszerza ona klasyczny rachunek zdań o dwie wzajemnie definiowalne modalności N (*Next*) oraz C (*Change*). Choć formuły odnoszą się do potencjalnie nieograniczonego czasu, a ich prawdziwość jest uzależniona od wszystkich stanów przyszłości, modalności te mają „lokalny” charakter w tym sensie, że dotyczą najbliższego otoczenia czasowego danego stanu. Z punktu widzenia konkretnego stanu/momentu czasu bardziej oddalone przyszłe punkty czasu są dostępne poprzez zagnieżdżanie tych modalności.

W informatyce od prawie 40 lat są intensywnie stosowane bardziej ogólne logiki czasu liniowego, w tym warianty LTL (*Linear-Time Temporal Logic*), mające istotnie większą siłę wyrażania niż LNC. Logiki te zostały gruntownie zbadane, w szczególności opracowano dla nich szereg metod formalnego wnioskowania (w tym automatycznego) oraz metod weryfikacji modelowej (*model checking*). Metody te są niejednokrotnie bardzo wyrafinowane i wiele z nich ma wysokiej jakości implementacje i jest formalną podstawą w poważnych zastosowaniach, np. w przemyśle informatycznym i telekomunikacyjnym.

Rozprawa jest poświęcona przede wszystkim metodom wnioskowania w logice LNC przy zastosowaniu badania spełnialności i weryfikacji modelowej.² Implementacje wybranych metod stworzono w języku SCALA. Pokazano też zastosowanie logiki LNC do generowania i weryfikacji ciągów binarnych. Praca liczy osiem rozdziałów. Po wstępie (stanowiącym rozdział 1) następuje wprowadzenie do logiki LNC i jej zanurzenia w LTL (oznaczonego przez LTL_{LNC}). Rozdział 3 jest poświęcony omówieniu istniejących metod wnioskowania i ich algorytmicznych aspektów. Dwa kolejne rozdziały dotyczą odpowiednio problemu spełnialności oraz weryfikacji modelowej. W rozdziale 6 pokazane zostały wyniki przeprowadzonych eksperymentów, zaś w rozdziale 7 – wybrane zastosowania badanego rachunku logicznego. Rozdział 8 podsumowuje rozprawę i prezentuje wynikające z niej wnioski. Załącznik A zawiera wybrane fragmenty implementacji

¹Logika LNC została wprowadzona przez K. Świątorzeczką w 2008 roku.

²W dalszej części recenzji przez $SAT(L)$, $MC(L)$ oznaczam odpowiednio problem spełnialności i weryfikacji modelowej dla logiki L .

LNC w języku SCALA. Załącznik B prezentuje zastosowania opracowanych metod dla trzech scenariuszy wybranych z literatury.

1. JAKIE ZAGADNIENIE NAUKOWE JEST ROZPATRZONE W PRACY (TEZA ROZPRAWY) I CZY ZOSTAŁO ONO DOSTATECZNIE JASNO SPRECYZOWANE PRZEZ AUTORA? JAKI CHARAKTER MA ROZPRAWA (TEORETYCZNY, DOŚWIADCZALNY, INNY)?

W pracy postawiono zagadnienie wykazania następujących tez (por. str. 17):

1. Problem SAT(LNC) jest NP-zupełny.
2. Problem MC(LNC) jest NP-zupełny.
3. Problem MC(LNC) można zastąpić problemem SAT(LTL).
4. Formuły LNC mogą być generatorami i walidatorami strumieni binarnych.

Powyższe problemy są interesujące. Dwa pierwsze z nich mają charakter teoretyczny, dwa kolejne wymagały opracowania odpowiednich algorytmów i ich eksperymentalnej weryfikacji.³

Dla osadzenia prowadzonych badań w kontekście literatury, rozważany jest także uzupełniający problem:

5. Logika LNC jest fragmentem LTL.

Problem ten jest także rozwiązany w rozprawie. Jednak jego merytoryczny status jest niższy niż problemów 1–4 (problem jest nietrudny do rozwiązania i Autor słusznie traktuje go jako uzupełniający). Posłużył on także eksperymentalnemu porównaniu LNC z wybranymi metodami wnioskowania w LTL.

Należy stwierdzić, że:

- zagadnienia naukowe i problemy postawione w rozprawie są jasno sformułowane;
 - część rozprawy ma charakter teoretyczny, zaś część – eksperymentalny, co uważam za zaletę rozprawy, bowiem w ten sposób wykazuje solidny warsztat badawczy Autora.
2. CZY W ROZPRAWIE PRZEPROWADZONO W SPOSÓB WŁAŚCIWY ANALIZĘ ŹRÓDEŁ (W TYM LITERATURY ŚWIATOWEJ, STANU WIEDZY I ZASTOSOWAŃ W PRZEMYSŁE) ŚWIADCZĄCY O DOSTATECZNEJ WIEDZY AUTORA? CZY WNIOSKI Z PRZEGLĄDU ŹRÓDEŁ SFORMULOWANO W SPOSÓB JASNY I PRZEKONYWUJĄCY?

W rozprawie zostało zacytowanych 87 źródeł. Przeglądowi istniejących metod jest poświęcony rozdział 3, w którym zostały omówione wyniki dotyczące:

- złożoności LTL;
- najważniejszych metod badania spełnialności (metoda tablic analitycznych dla klasycznego rachunku zdań (KRZ) oraz LTL, automaty Büchiego, solvery dla problemu SAT(KRZ));
- metod weryfikacji modelowej (w tym automatowych, symbolicznych i diagramów BDD).

³Z czysto teoretycznego punktu widzenia zastąpienie $MC(p)$ przez $SAT(LTL)$, o którym mowa w trzecim problemie, jest dość oczywiste. To co jest nieoczywiste i weryfikowane w rozprawie, to pytanie na ile takie zastąpienie jest skuteczne w zastosowaniach z wydajnościowego punktu widzenia, w tym przy wyborze adekwatnego fragmentu LTL.

Autor nie odniósł się do literatury związanej z zastosowaniami logik temporalnych w takich dziedzinach, jak systemy wieloagentowe, czy teorie akcji i zmiany (*action and change*) ale zapewne uznał, że leżą na tyle daleko od głównego nurtu jego badań, że ich omówienie nie jest konieczne. Choć rozprawa z omówieniem tych dziedzin byłaby bogatsza, można uznać taki punkt widzenia. Analiza źródeł jest więc przeprowadzona w wystarczająco szerokim zakresie i świadczy o dostatecznej wiedzy Autora.

3. CZY AUTOR ROZWIĄZAŁ POSTAWIONE ZAGADNIENIA, CZY UŻYŁ WŁAŚCIWEJ DO TEGO METODY I CZY PRZYJĘTE ZAŁOŻENIA SĄ UZASADNIONE?

Poniżej odnoszę się do poszczególnych postawionych problemów.

* PROBLEM 1: NP-ZUPEŁNOŚĆ PROBLEMU SAT(LNC)

W pracy wykazano NP-zupełność SAT(LTL_{LNC}), co jest znaczącym obniżeniem złożoności w stosunku do fragmentu LTL z operatorami G, X , którego problem spełnialności jest PSPACE-zupełny.⁴ Dowód, że SAT(LTL_{LNC}) jest w klasie NP został przeprowadzony przez jego redukcję do SAT(KRZ) (rozdział 4.2).

W rozdz. 4.3 rozważona jest także metoda badania spełnialności poprzez wykrywanie cykli w tablicach analitycznych.

* PROBLEMY 2-3: NP-ZUPEŁNOŚĆ PROBLEMU MC(LNC) I ZASTĘPOWALNOŚĆ MC(LNC) PRZEZ SAT(LTL)

Aspektem teoretycznym i algorytmicznym problemów 2-3 jest poświęcony rozdział 5 rozprawy. Zasadnicza idea polega tu na redukcji MC(LTL_{LNC}) do problemu SAT(LTL_{LNC}). Okazuje się, że weryfikację modelową można ograniczyć do głębokości odpowiadającej maksymalnemu zagnieżdżeniu operatora X w badanej formule. Problemem okazuje się tu wykładnicza eksplozja liczby ścieżek. Aby radzić sobie z tym problemem, Autor rozważa odpowiednie kodowanie stanów.

Weryfikacji eksperymentalnej i jej wynikom poświęcony jest rozdział 6. Na jej potrzeby została zaimplementowana metoda tablic analitycznych. Dla problemu SAT(KRZ) użyto kilku znanych narzędzi: Minisat, Glucose oraz Lingeling wraz z konwerterem formuł do koniunkcyjnej postaci normalnej bool2cnf (działającym w deterministycznym czasie wielomianowym). Do porównania użyto znanych narzędzi: pltl (tablice analityczne dla LTL), TRP (rezolucja dla LTL), Aalta (automaty Büchiego) oraz NuSMV (dla logiki CTL, bogatszej od LTL). Wyniki są tu bardzo obiecujące. Choć bogato zilustrowane wykresami, ich opis pozostawia jednak niedosyt, zwłaszcza w części dotyczącej SAT(LTL_{LNC}), np.:

- jakie zasady stały za doбором formuł do eksperymentów?
- dlaczego część badań dotyczy wyłącznie wspomnianych narzędzi obcych (por. np. tabele/rys. 6.5, 6.6)?
- brak szerszej dyskusji wniosków wynikających z eksperymentów.

Dużo lepiej prezentują się pod tym względem opisy eksperymentów dotyczących MC(LTL_{LNC}).

* PROBLEM 4: ZASTOSOWANIE FORMUŁ LNC JAKO GENERATORÓW I WALIDATORÓW STRUMIENI BINARNYCH

Problem ten jest rozwiązany w rozdziale 7.3. Rozdział ten jest zresztą szerszy – zarysowuje także możliwe zastosowania badanego formalizmu jako języka specyfikacji wykonywalnych oraz planowania (rozdz. 7.1, 7.2, uzupełnione dodatkiem B).

⁴Zakładając, że $NP \neq PSPACE$, co choć jest stawianą hipotezą – jak wiadomo – jest otwartym problemem.

Autor pokazał interesujące zastosowanie LNC w generowaniu ciągów binarnych (odpowiadających modelom formuł) oraz ich weryfikacji. Zbadał także czas i skuteczność wykrywania przekłamań. Ponadto opracowany został w rozprawie algorytm automatycznej korekcji i zbadana jego skuteczność.

PODSUMOWANIE

Ponieważ problem 5 również został rozwiązany (rozdział 2.4 rozprawy), postawione w rozprawie problemy zostały przez Autora rozwiązane przy użyciu adekwatnych metod.

4. NA CZYM POLEGA ORYGINALNOŚĆ ROZPRAWY, CO STANOWI SAMODZIELNY I ORYGINALNY DORÓBEK AUTORA, JAKA JEST POZYCJA ROZPRAWY W STOSUNKU DO STANU WIEDZY CZY POZIOMU TECHNIKI REPREZENTOWANYCH PRZEZ LITERATURĘ ŚWIATOWĄ?

Oryginalność rozprawy polega na zbadaniu fragmentu LNC logiki temporalnej LTL i wykazaniu, że fragment ten:

- pozwala na opracowanie znacząco wydajniejszych algorytmów badania spełnialności i weryfikacji modelowej od tych, które są opracowane dla pełnych wersji logik temporalnych;
- ma siłę wyrazu pozwalającą na wykorzystanie w praktyce do wielu problemów o ograniczonym horyzoncie czasowym.

W celu wykazania powyższych punktów Autor wykazał teoretycznie klasy złożoności badanych problemów, opracował algorytmy oraz przeprowadził badania eksperymentalne pokazujące na skuteczność zaproponowanych rozwiązań. Wszystkie te aspekty stanowią także wkład autorski.

5. CZY AUTOR WYKAZAŁ UMIEJĘTNOŚĆ POPRAWNEGO I PRZEKONYWUJĄCEGO PRZEDSTAWIENIA UŻYSKANYCH PRZEZ SIEBIE WYNIKÓW (ZWIĘŻŁOŚĆ, JASNOŚĆ, POPRAWNOŚĆ REDAKCYJNA PRACY)?

Autor wykazał opanowanie warsztatu badawczego, w tym umiejętność poprawnego i przekonującego przedstawiania swoich wyników.

6. JAKIE SĄ SŁABE STRONY ROZPRAWY I JEJ GŁÓWNE WADY?

Główną, wspomnianą już wcześniej słabością, jest zbyt mało wyczerpująca dyskusja badań eksperymentalnych dotyczących problemu spełnialności. Podobnie – omówienie literatury odnoszące się do innych dziedzin informatycznych zastosowań logik temporalnych wzbogaciłoby rozprawę.

Można też żałować, że wyniki pracy nie są opublikowane w źródłach zajmujących się logikami temporalnymi jako głównym lub jednym z głównych zagadnień. Na pewno warto by Autor podjął wysiłki zmierzające w tym kierunku.

7. JAKA JEST PRZYDATNOŚĆ ROZPRAWY DLA NAUK TECHNICZNYCH?

Zbadanie logiki LNC pod kątem jej zastosowań w informatyce jest przydatne dla nauk technicznych biorąc pod uwagę:

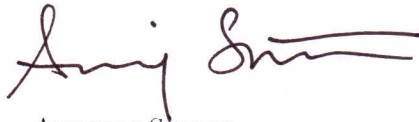
- znaczącą poprawę wydajności stosowanych technik w stosunku do metod znanych w literaturze przedmiotu, wynikającą z ograniczenia siły LNC w stosunku do logik temporalnych badanych dotychczas;

- ważne obszary zastosowań LNC w informatyce, w szczególności pokazanie, że ograniczenie do modalności N i C nadal umożliwia weryfikację ciekawej podklasy problemów specyfikowanych w LNC i weryfikowanych w LTL_{LNC} ;
- zarysowaną w rozprawie możliwość zastosowań LNC w planowaniu (wymagająca jednak dalszych badań).

8. DO KTÓREJ Z NASTĘPUJĄCYCH KATEGORII RECENZENT ZALICZA ROZPRAWĘ:

- (A) NIE SPELNIĄCA WYMAGAŃ STAWIANYCH ROZPRAWOM DOKTORSKIM PRZEZ OBOWIĄZUJĄCE PRZEPISY
- (B) WYMAGAJĄCA WPROWADZENIA POPRAWEK I PONÓWNEGO RECENZOWANIA
- (C) SPELNIĄCA WYMAGANIA
- (D) SPELNIĄCA WYMAGANIA Z WYRAŹNYM NADMIAREM
- (E) WYBITNIE DOBRA, ZASŁUGUJĄCA NA WYRÓŻNIENIE

Mimo wskazanych drobnych niedoskonałości, rozprawa zawiera ciekawe wyniki teoretyczne i eksperymentalne. Uważam, że spełnia ona wymagania ustawowe i zwyczajowe stawiane przed rozprawami doktorskimi. Zaliczam ją więc do kategorii C i wnoszę o jej dopuszczenie do dalszych faz przewodu doktorskiego.



ANDRZEJ SZALAS



INSTYTUT PODSTAW INFORMATYKI
POLSKIEJ AKADEMII NAUK
ul. Jana Kazimierza 5,
01-248 Warszawa

tel.: (48-22) 380-05-00

fax: (48-22) 380-05-10

Prof. dr hab. Wojciech Penczek
Dyrektor IPI APN

penczek@ipipan.waw.pl
<http://www.ipipan.waw.pl/~penczek>

Warszawa, 12.09.2019

**Kwestionariusz-Recenzja Rozprawy Doktorskiej Dla Rady
Wydziału Elektroniki i Technik Informatycznych
Politechniki Warszawskiej**

Tytuł Rozprawy: Rozwój aplikacji logik temporalnych w zastosowaniach informatycznych

Autor Rozprawy: mgr inż. Marek Adamek

Jest to ponowna recenzja rozprawy doktorskiej mgra Marka Adamka. Rozprawa została istotnie poprawiona i rozszerzona. Dziedzina rozprawy nie uległa zmianie, dlatego jej opis w pierwszym punkcie tej recenzji jest podobny do poprzedniego z mojej recenzji z dnia 15.08.2018 roku.

1. Dziedzina rozprawy doktorskiej

Logiki temporalne czasu liniowego i rozgałęzionego wykorzystywane są do specyfikowania i weryfikowania systemów informatycznych, których poprawność jest często krytyczna dla zdrowia i życia użytkowników. Do najbardziej popularnych logik temporalnych zalicza się dwa następujące formalizmy: LTL (liniowa logika temporalna) i CTL (logika temporalna czasu rozgałęzionego), które pozwalają na specyfikowanie pełnej i częściowej poprawności systemów oraz programów komputerowych jak również braku blokady i zakleszczenia. Weryfikacja modelowa (ang. model checking) jest najczęściej używaną metodą do weryfikacji systemów przy użyciu logik temporalnych. Istota tej metody polega na reprezentacji systemu przez jego model (strukturę Kripkego), zapisaniu własności w postaci formuły logiki, a następnie sprawdzeniu czy ta formuła jest prawdziwa w stanach początkowych modelu. Problem weryfikacji modelowej jest trudny, tzn. należy do klasy problemów NP-zupełnych lub PSPACE-zupełnych w zależności od reprezentacji modelu i skomplikowania formuły. Podobnie trudny jest problem spełnialności. Z tego powodu rozwój

heurystycznych metod testowania spełnialności jak również heurystycznych metod weryfikacji modelowej jest ważnym i aktualnym problemem.

Rozprawa doktorska mgra Marka Adamka poświęcona jest zagadnieniom wykorzystania logiki LNC (Logic Next Change) w zastosowaniach informatycznych. LNC rozszerza klasyczną logikę zdaniową o dwa operatory C i N, które opisują zmiany pomiędzy danym stanem i jego następnikiem w sekwencji stanów. W rozprawie zawarte są wyniki dotyczące testowania spełnialności oraz weryfikacji modelowej dla LNC, poparte wynikami eksperymentalnymi dla zaimplementowanych algorytmów.

2. Tezy rozprawy doktorskiej w kontekście wyników na świecie

Autor rozprawy doktorskiej stawia cztery poniższe tezy:

1. Problem spełnialności dla logiki LNC jest NP-zupełny.
2. Problemu weryfikacji modelowej dla logiki LNC jest NP-zupełny.
3. Formuły logiki LNC są symbolicznym zapisem modeli w problemie MC (ang. model checking, po polsku: weryfikacja modelowa). Problem MC dla tych modeli można zastąpić problemem badania spełnialności w logice LTL.
4. Formuły logiki LNC mogą być generatorami i walidatorami strumieni binarnych.

Teza 1 i 2 jest bezpośrednią konsekwencją faktu, że LNC jest fragmentem logiki LTL, dla którego to fragmentu takie wyniki zostały już wcześniej pokazane. Jednakże autor wykorzystuje w swojej rozprawie ten fakt, żeby zaprezentować nowe metody testowania spełnialności i weryfikacji modelowej specyficzne dla tego fragmentu. Ten fakt nadaje pracy pewną oryginalność.

Teza 3 jest ciekawą obserwacją, która jest również prawdziwa dla logiki LTL. Natomiast Teza 4 jest odpowiednikiem Tezy 3 z poprzedniej wersji rozprawy. Dotyczy ona pewnego specyficznego zastosowania LNC.

Reasumując, postawione tezy dają podstawę do zaproponowania nowych metod do testowania spełnialności oraz do weryfikacji modelowej dla logiki LNC.

3. Analiza literatury światowej i stanu wiedzy w dziedzinie rozprawy

W porównaniu z poprzednią wersją rozprawy analiza literatury światowej jest zdecydowanie bardziej kompletna. W Rozdziale 3 rozprawy doktorskiej autor dokonuje analizy wyników dotyczących złożoności problemów spełnialności i weryfikacji modelowej oraz stosowanych

algorytmów dla LTL i fragmentów LTL. Nadal jednak brakuje mi na liście bibliografii następujących pozycji:

- Edmund M. Clarke, Orna Grumberg, Doron A. Peled: Model checking. MIT Press 2001, ISBN 978-0-262-03270-4, pp. I-XIV, 1-314
- Christel Baier, Joost-Pieter Katoen: Principles of model checking. MIT Press 2008, ISBN 978-0-262-02649-9, pp. I-XVII, 1-975
- Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, Roderick Bloem: Handbook of Model Checking. Springer 2018, ISBN 978-3-319-10574-1

4. Ocena wyników uzyskanych w rozprawie

Rozprawa liczy wraz z bibliografią i dodatkami 203 strony formatu A5. Zatem jest rozmiar zwiększył się w stosunku do poprzedniej wersji o ponad 100 stron. Rozprawa składa się z 7 rozdziałów, które kolejno omawiam.

Rozdział 1. „Wstęp” liczy 8 stron i dotyczy przeglądu logik temporalnych i ich zastosowań, oraz sformułowania problemu i celu pracy. Na bazie analizy złożoności problemów spełnialności i MC, autor uzasadnia, że warto zajmować się fragmentem LTL, dla którego złożoność jest NP-zupełna a nie PSPACE-zupełna. Wiadomo, że klasa NP zawiera się w PSPACE, więc można liczyć na to, że jeśli te klasy nie są równe, to problemy z klasy NP mogą mieć heurystyczne rozwiązania, które będą bardziej efektywne niż dla tych z klasy PSPACE. Ponieważ, dla LTL oba problemy są w PSPACE, a dla niektórych fragmentów w NP, to warto zajmować się tymi fragmentami, przynajmniej z punktu widzenia ich niższej złożoności obliczeniowej.

Rozdział 2. „Logiki temporalne LNC i LTL_LNC” liczy 10 stron i dotyczy składni i semantyki logiki LTL i LNC. Zostało pokazane, że LNC jest fragmentem LTL, a dokładniej zdefiniowano logikę LTL_LNC, która jest fragmentem LTL i każda formuła LNC może być w niej wyrażona.

Rozdział 3. „Przegląd istniejących metod” liczy 10 stron i dotyczy złożoności problemu SAT i MC dla fragmentów LTL, oraz metod testowania spełnialności i weryfikacji modelowej dla

LTL. Autor opisuje metody tabel semantycznych, automatowe, wykorzystujące SAT-solvery i Boolowskie diagramy decyzyjne (BDD).

Rozdział 4. „Problem spełnialności w logice LTL_LNC” liczy 10 stron i opisuje dwie oryginalne metody autora badania spełnialności dla formuł logiki LTL_LNC. Pierwsza z tych metod polega na redukcji problemu spełnialności do problemu SAT formuły prefiksowej (Definicja 4.6), a druga na detekcji cykli formuły bazowej (modyfikacja metody tabel semantycznych). Formuła f jest bazowa jeśli Gf jest formułą LTL_LNC (Definicja 2.20). Pesymistyczna złożoność obliczeniowa obu metod jest wykładnicza. Jednakże jak zostało pokazane w Rozdziale 6. wyniki eksperymentalne potwierdzają dużą skuteczność metody detekcji cykli.

Rozdział 5. „Weryfikacja modelowa w logice LTL_LNC” liczy 14 stron i opisuje oryginalny algorytm autora do weryfikacji modelowej dla logiki LTL_LNC. Problem ten jest redukowany do problemu SAT. Ponadto, w rozdziale tym zostały opisane dwa algorytmy syntezy struktury Kripkego z formuły w logice LTL_LNC. Złożoność obu algorytmów jest wykładnicza.

Rozdział 6. „Weryfikacja eksperymentalna” liczy 42 strony i opisuje wyniki eksperymentalne uzyskane dla zaimplementowanych algorytmów. Ten rozdział dowodzi, że algorytmy dla logiki LTL_LNC opisane w Rozdziale 4. i 5. zachowują się lepiej niż algorytmy, zaprojektowane dla logiki LTL a zastosowane do formuł logiki LTL_LNC. W podrozdziale 6.2 zostały przeprowadzone eksperymenty porównawcze dla badania spełnialności formuł LTL_LNC. Wyniki eksperymentalne dla metod autora nazywanych: „SAT formuły prefiksowej” i „detekcja cykli” zostały zestawione z wynikami dla następujących narzędzi: PLTL (implementacja metod tabel semantycznych w C++), TRP++ (implementacja algorytmu rezolucji dla logiki temporalnej w języku OCAML), Aalta v0.2, NuSMV 2.6.0. Eksperymenty pokazują, że metoda „SAT formuły prefiksowej” jest bardziej efektywna niż pozostałe metody lub narzędzia. Również metoda „detekcji cykli” jest najczęściej bardziej efektywna (ale nie zawsze) od innych metod dla formuł bazujących na operatorze C. Ciekawą obserwacją jest niezbyt efektywne zachowanie metody tabel semantycznych dla formuły prefiksowej.

W podrozdziale 6.3 zostały przeprowadzone eksperymenty porównawcze dla weryfikacji modelowej formuł LTL_LNC. Wyniki eksperymentalne dla metody autora nazwanej LNC

MC zostały zestawione z wynikami dla następujących narzędzi: SPOT (metoda automatowa) i NuSMV. Metoda LNC MC polega na kolejnym sprawdzaniu modeli formuły bazowej weryfikującej model. W przeprowadzonych eksperymentach metoda LNC MC okazała się najlepsza.

Rozdział 7 „Zastosowania rachunku LNC” liczy 17 stron i prezentuje zastosowanie LNC do generowania i weryfikacji ciągów binarnych. W tym rozdziale zaproponowano również metodę kodowania wiadomości przy wykorzystaniu LNC. Myślę, że warto byłoby omówić również inne metody, które można wykorzystać dla weryfikowania ciągów binarnych i porównać ich efektywność z metodą z rozprawy.

Rozdział 8 „Zakończenie i wnioski” liczy 5 stron i omawia uzyskane wyniki i inne możliwe zastosowania logiki LNC.

5. Uwagi krytyczne

Nowa wersja rozprawy doktorskiej jest zdecydowanie bardziej precyzyjnie zredagowana niż poprzednia. Ale i w niej nie wszystkie definicje są poprawnie sformułowane. Przykładowo Definicja 4,4 jest trudna do zrozumienia i bez zobaczenia przykładów wymaga zgadnięcia co autor miał na myśli. Rola niektórych fragmentów rozprawy nie jest do końca jasna, np. podrozdział 5.3.3 opisuje kodowanie w postaci BDD, ale autor nie używa tych diagramów w swoich implementacjach. W Rozdziale 6 wystąpiły problemy edycyjne związane z opisami rysunków, np. Rysunku 6.1. Ponadto, nie wszystkie wyniki umieszczone na wykresach zostały poddane analizie. Odrębnym problemem jest użyteczność logiki LNC w zastosowaniach informatycznych, która jest dość ograniczona a sama logika jest raczej niewykorzystywana w literaturze światowej. Jednakże, Autor przedstawił takie zastosowania na kilku przykładach w Rozdziale 7.

W pracy jest bardzo dużo drobnych problemów formalnych i redakcyjnych. Poniżej lista kilku przykładowych takich problemów:

1. Str. 20: dlaczego funkcja π_{PLTL} jest tak zdefiniowana, że przypisuje wartości podzbiorom AP a nie elementom ?
2. Składnia logik temporalnych nie zawiera symboli dla prawdy i fałszu, ale są one używane w formułach,
3. Formuła 2.32 jest niepoprawna bo operatora X nie stosuje się do zbiorów,

4. Formuła 4.2: brakuje kwantyfikatora ogólnego: dla każdego t należącego do N .
5. Definicja 4.4 jest niezrozumiała. Co to znaczy, że literały temporalne mogą przyjąć dowolne wartości logiczne ?
6. Czy w Lemacie 4.1 nie wystarczyłoby zachowanie spełnialności ?
7. Formuła 4.14: czy formuła θ jest dowolna ?
8. Algorytm 5.1: co oznacza $\text{BuildFormula}(v, K, d)$?
9. Rysunek 6.2.1: konwencja przypisania literałów do stanów jest niezrozumiała.
10. Str. 148: referencja [51] jest niepełna.

6. Podsumowanie i konkluzja

Główny wynik rozprawy doktorskiej polega na podaniu nowych heurystycznych algorytmów do testowania spełnialności i do weryfikacji modelowej dla logiki LNC. Efektywność opracowanych algorytmów została porównana z innymi algorytmami, które istnieją w literaturze światowej. Wyniki eksperymentalne wykazały lepszą efektywność nowych algorytmów. Ponadto, Autor dokonał oceny złożoności obliczeniowej zaproponowanych metod i uzasadnił ich poprawność. Uważam, że poprawiona wersja rozprawy doktorskiej zawiera oryginalne elementy dla rozwoju aplikacji logik temporalnych w zastosowaniach informatycznych a zatem może być przydatna dla nauk technicznych, w zastosowaniach logiki LNC.

Reasumując, stwierdzam, że rozprawa doktorska mgr inż. Marka Adamka spełnia wymagania stawiane rozprawom doktorskim w aktualnie obowiązującej Ustawie o Stopniach Naukowych i Tytule Naukowym, i wnoszę o dopuszczenie Autora do dalszych etapów przewodu doktorskiego.



Wojciech Penczek