

dr hab. Bożena Woźna-Szcześniak, prof. UJD
Uniwersytet Humanistyczno-Przyrodniczy
im. Jana Długosza w Częstochowie
ul. Waszyngtona 4/8, 42-200 Częstochowa

RECENZJA

rozprawy doktorskiej

Tytuł rozprawy: Metody generowania etykietowanych systemów przejść dla języka Alvis

Autor rozprawy: mgr inż. Michał Wypych

Recenzja wykonana jest na zlecenie Rady Dyscypliny Informatyki Technicznej i Telekomunikacji, Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie, pismo z dnia 18 października 2021r.

Recenzowana rozprawa doktorska napisana została pod kierunkiem promotora prof. dr hab. Marcina Szpyrki (Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, AGH). Praca lokuje się w dyscyplinie informatyka techniczna i telekomunikacja.

1. *Cel, zakres i charakter rozprawy.*

Rozprawa doktorska mgra inż. Michała Wypycha dotyczy metod formalnej weryfikacji modeli systemów współbieżnych oraz rozproszonych wyspecyfikowanych w języku *Alvis*.

Automatyczna weryfikacja systemów realizowana przez analizę modeli jest bardzo ważnym tematem badań naukowych. Wynika to z rosnącego zapotrzebowania na weryfikację systemów o krytycznym znaczeniu dla bezpieczeństwa, których awaria może powodować dramatyczne konsekwencje dla ludzi i urządzeń. Do takich systemów zaliczają się systemy sterowania ruchem kolejowym/tramwajowym/lotniczym, kontrolery hamulcowe, systemy planowania lotów i wiele innych.

Problemem większości popularnych narzędzi do weryfikacji formalnej systemów współbieżnych oraz rozproszonych jest ich dostępność dla przeciętnego inżyniera oprogramowania. Narzędzie *Alvis* jest przeznaczone do formalnej specyfikacji i weryfikacji takich systemów, ale w sposób przystępny dla przeciętnego inżyniera oprogramowania. Oceniana rozprawa zawiera nowe rozwiązania umożliwiające formalną weryfikację modeli zaprojektowanych w języku *Alvis* i w ten sposób istotnie przyczynia się do rozszerzenia dotychczasowych funkcjonalności narzędzia *Alvis*.

Formalny cel rozprawy jest jasno i precyzyjnie sformułowana przez Autora. Mianowicie, głównym celem rozprawy jest zaproponowanie, zaprojektowanie i wdrożenie narzędzia umożliwiającego automatyczne, szybkie i nienadzorowane generowanie etykietowanych systemów przejść (ang. Labelled Transition System, LTS) z formalnego opisu modelu systemu w nieczasowej wersji języka *Alvis* z warstwą systemową α^0 .

Aby osiągnąć postawiony cel, Autor sformułował następujące cele pośrednie:

- Opracowanie zestawu algorytmów, które przekształcą formalny model zapisany w języku Alvisa na jego postać pośrednią w języku Haskell (ang. Intermediate Haskell Representation, IHR).
- Implementacja zaproponowanych algorytmów w postaci kompilatora Alvis – narzędzia, które automatycznie weryfikuje poprawność syntaktyczną projektu modelu zapisanego w języku Alvis oraz generuje postać pośrednią w języku Haskell odzwierciedlającą zaprojektowany model.
- Opracowanie i implementacja wydajnego algorytmu będącego częścią pośredniej reprezentacji w języku Haskell, który pozwala na wygenerowanie jawnego etykietowanego systemu przejść.

Ponieważ rozprawa ma rozbudowaną stronę aplikacyjną i w zasadzie nie zawiera strony analitycznej, a wyniki opisane w pracy bazują na znajomości i umiejętności stosowania metod formalnych oraz wyspecjalizowanych języków programowania, pracę zaliczam do rozpraw o charakterze aplikacyjnym.

2. *Struktura i zawartość rozprawy.* Opiniowana rozprawa doktorska została przygotowana w języku angielskim i składa się ze wstępu, pięciu rozdziałów, podsumowania oraz jednego dodatku. Zawiera ponadto stronę tytułową w językach polskim i angielskim, stronę z podziękowaniami, streszczenia w językach polskim i angielskim, spis treści, spis listingów oraz bibliografię. Praca liczy łącznie 244 strony. Bibliografia obejmuje 184 pozycje, w tym pozycje książkowe, artykuły w czasopismach naukowych oraz recenzowanych materiałach konferencyjnych, a także raporty techniczne, prace dyplomowe, rozprawy doktorskie i dokumentacje techniczne. Treść poszczególnych rozdziałów daje syntetyczny pogląd na przedstawioną w nich zawartość merytoryczną. W szczególności:

- Rozdział 1 stanowi wstęp do rozprawy, zawiera motywację do prowadzonych badań oraz cel badań. W rozdziale tym omówione zostały krótko wybrane metody weryfikacji oprogramowania poprzez testowanie, które jest najczęściej stosowanym podejściem w celu eliminacji błędów oprogramowania i utrzymania niskiego ryzyka jego awarii. Opisano również przykłady problemów obliczeń współbieżnych, które ze swojej natury są trudne do weryfikacji poprzez testowanie: *problem ucztujących filozofów*, *problem czytelników i pisarzy* oraz *problem producenta i konsumenta*. Na ich przykładzie scharakteryzowano klasyczne mechanizmy synchronizacji procesów (wątków) współbieżnych wykorzystujące algorytmy z blokadami oraz algorytmy nieblokujące. Wprowadzono pojęcie *własności bezpieczeństwa i żywotności* – cech typowych dla systemów współbieżnych. Wprowadzono również logiki temporalne (LTL, CTL, μ -rachunek) jako formalizm, który z powodzeniem od lat 80. XX wieku jest stosowany do specyfikacji i weryfikacji systemów współbieżnych. Zdefiniowano pojęcie weryfikacji modelowej oraz krótko opisano techniki unikania wykładniczej eksplozji stanów – typowego problemu, z którym zmagają się wszystkie algorytmy automatycznej weryfikacji modelowej. Na końcu rozdziału zamieszczono zwięzłą historię narzędzia Alvis oraz tezę rozprawy.
- Rozdział 2 stanowi przegląd istniejącego stanu wiedzy w zakresie, którego dotyczy rozprawa. W szczególności zawiera zwięzłe omówienie wybranych języków formalnych (sieci Petriego, algebra procesów, TLA+) oraz narzędzi (nuXmv, CADP, Uppaal, PRISM, SPIN), które można porównać z narzędziem Alvis. Porównaniu towarzyszą przykładowe modele, które pozwalają lepiej zobrazować różnice między Alvisem a omawianymi formalizmami.
- Rozdział 3 zawiera opis elementów formalnego języka modelowania Alvis, które są niezbędne do zrozumienia dalszych części rozprawy. W szczególności, podrozdział 3.1 opisuje graficzną warstwę modelu, podrozdział 3.2 opisuje warstwę kodu, a podrozdział 3.3 opisuje warstwę systemową. Warstwy te pozwalają na projektowanie systemu współbieżnego/rozproszonego, ale nie są odpowiednie do wykonywania automatycznej weryfikacji modelowej. W tym celu w podrozdziałach 3.4 i 3.5 wprowadzono formalną definicję modelu Alvis; pojęcie stanu modelu opisano w podrozdziale 3.4, a zasady rządzące zachowaniem modelu - tranzycje -

opisano w podrozdziale 3.5. Stany i tranzycje modelu Alvis są połączone w strukturę podobną do grafu, zwaną *Etykietowanym Systemem Przejść* (ang. Labelled Transition System, LTS), którego rozważana w rozprawie definicja została wprowadzona w podrozdziale 3.5.

Rozdział 3 nie jest tylko krótką prezentacją dawnej pracy zespołu prof. dr hab. M. Szpyrki [163, 169] na temat syntaktyki i semantyki języka Alvis, ale opisem jego zaktualizowanej wersji, w której wprowadzono kilka modyfikacji zestawu instrukcji Alvisa, takich jak: *nieblokujące instrukcje komunikacyjne*, usunięcie instrukcji *if* (jako nadmiarowe w stosunku do instrukcji *select*), uproszczenie semantyki instrukcji agenta pasywnego przez wymaganie, aby każda procedura kończyła się jawnie instrukcją *exit*.

- Rozdział 4 przedstawia główny wynik rozprawy, czyli algorytmy i struktury danych, które umożliwiają przekształcenie modelu zaprojektowanego w języku Alvis (przy użyciu warstwy graficznej, warstwy kodu i warstwy systemowej α^0 w wersji bez czasu) na jego reprezentację pośrednią w języku Haskell (ang. Intermediate Haskell Representation, IHR). Prezentowane algorytmy stanowią implementację zasad formalnych (reguł umożliwiania tranzycji - **E** i reguł odpalania tranzycji - **F**) wprowadzonych w rozdziale 3. Reprezentacja pośrednia grafu LTS w języku Haskell to kod źródłowy programu, który umożliwia generowanie jawnego LTS w różnych formatach i bazuje na dwóch funkcjach: **enable** oraz **fire**, które szczegółowo omówiono w podrozdziałach 4.3 i 4.4. Przed wprowadzeniem funkcji **enable** oraz **fire** w podrozdziale 4.1 omówiono szczegółowo struktury danych reprezentujące stan systemu w IHR, a w podrozdziale 4.2 omówiono szczegółową reprezentację tranzycji w IHR modelujących dynamikę systemu. Podrozdział 4.5 zawiera opis funkcji pomocniczych wykorzystywanych w funkcjach **enable** oraz **fire**. Podrozdział 4.6 wprowadza definicję pełnego diagramu komunikacji, który rozszerza pojęcie diagramu komunikacji wprowadzonego w podrozdziale 3.1.
- Rozdział 5 zawiera omówienie metody projektowania i implementacji algorytmu, wykorzystującego funkcje **enable** i **fire**, generującego jawny etykietowany system przejść (jawny LTS) na podstawie reprezentacji pośredniej zapisanej w języku Haskell (IHR). Podrozdział 5.1 stanowi wprowadzenie i uzasadnienie do metod opisanych w podrozdziałach 5.2 i 5.3. Podrozdział 5.2 omawia algorytm przeszukiwania niejawnego etykietowanego systemu przejść (niejawnego LTS) zdefiniowanego w IHR w celu utworzenia jego jawnego LTS oraz przedstawia analizę złożoności obliczeniowej tego algorytmu. Podrozdział 5.3 prezentuje problem ograniczania przestrzeni stanów przed jej eksploracją oraz pokazuje jak ewentualnie można dodać redukcje częściowo porządkowe do głównego algorytmu generującego LTS z reprezentacji pośredniej w IHR, bez modyfikacji algorytmu przeszukiwania opisanego w podrozdziale 5.2.
- Rozdział 6 zawiera opis pomiaru wydajności algorytmu zaproponowanego w rozdziale 5. Wydajność algorytmu ma ogromne znaczenie, ponieważ bardziej wydajny algorytm pozwala na weryfikację bardziej złożonych i zaawansowanych systemów.
- Rozdział 7 zawiera streszczenie rozprawy. Oprócz podsumowania głównych osiągnięć Autora przedstawionych w rozprawie, omawiane są plany na przyszłość i możliwe rozszerzenia narzędzia Alvis.

Prezentacja materiału przedstawionego w pracy dokonana jest w sposób relatywnie poprawny z językowego punktu widzenia. W pracy trafiają się literówki i drobne błędy gramatyczne oraz stylistyczne. Inne uwagi techniczno-redakcyjne przedstawiono w dalszej części recenzji.

3. *Czy w rozprawie przeprowadzono w sposób właściwy analizę źródeł (w tym literatury światowej, stanu wiedzy i zastosowań w przemyśle) świadczącej o dostatecznej wiedzy Autora. Czy wnioski z przeglądu źródeł sformułowano w sposób jasny i przekonujący?*

Motywacja dla podjęcia tematu rozprawy wynikała z dobrze przeprowadzonej przez Autora analizy źródeł w zakresie automatycznej weryfikacji modelowej i narzędzi z nią związanych. Dzięki erudycji autora w pełni odzwierciedlony został aktualny stan wiedzy w zakresie narzędzi inżynierskich do automatycznej weryfikacji modelowej systemów współbieżnych. Zabrakło może jedynie

odniesienia do narzędzia MCMAS (<https://vas.doc.ic.ac.uk/software/mcmas/>), przy pomocy którego można weryfikować systemy wieloagentowe.

Analizy wykonane na podstawie źródeł świadczą o dość obszernej wiedzy Autora. Wnioski z tej analizy są sformułowane w sposób jasny i przekonujący, pozwalający Autorowi właściwie zdefiniować tezę rozprawy, odpowiadającą na rzeczywiste zapotrzebowanie w inżynierii oprogramowania na przyjazne dla użytkownika narzędzia do automatycznej weryfikacji modelowej systemów współbieżnych i rozproszonych.

4. *Na czym polega oryginalność rozprawy, co stanowi samodzielny i oryginalny dorobek Autora, jaka jest pozycja rozprawy w stosunku do stanu wiedzy czy poziomu techniki reprezentowanych przez literaturę światową?*

Rozwiązując postawione w tezie rozprawy zadanie Autor uzyskał następujące nowe rezultaty, stanowiące jego samodzielny i oryginalny dorobek. Osiągnięcia te zawarto w rozdziałach od trzeciego do szóstego:

- opracowanie zbioru wydajnych algorytmów, które przekształcają model systemu zapisany w języku Alvis do jego postaci pośredniej zapisanej w języku Haskell (model Alvis \implies IHR);
- implementacja i wdrożenie narzędzia wspomagające przetwarzanie i analizę modeli Alvisa zgodnie z teoretycznymi zasadami rządzącymi semantyką modeli Alvisa;
- opracowanie oraz implementacja kompilatora Alvis – narzędzia, które automatycznie weryfikuje poprawność składniową projektu modelu Alvis i generuje postać pośrednią w języku Haskell odzwierciedlającą zaprojektowany model;
- rozszerzenie języka Alvis o nieblokujące instrukcje komunikacyjne;
- opracowanie oraz implementacja algorytmów i struktur danych umożliwiających automatyczne generowanie jawnego etykietowanego systemu przejść dla danego systemu z jego reprezentacji pośredniej zapisanej w języku Haskell (IHR \implies LTS);
- opracowanie testowych, skalowalnych modeli systemów współbieżnych w celu oceny wydajności zaproponowanych algorytmów;
- implementacja środowiska testowego do uruchamiania proponowanych modeli testowych i różnych algorytmów generowania LTS z wiarygodnymi pomiarami ich wydajności;
- ocena wydajności.

Warto również podkreślić całkiem dobry współczynnik Hirscha ($h=4$, wg. bazy danych <https://www.scopus.com>), który stanowi dodatkową ilustrację osiągnięć naukowych Autora, głównie tych zawartych w rozprawie. Oceniana rozprawa z powodzeniem wpisuje się w nurt aktualnych badań światowych dotyczących rozwoju nowych metod automatycznej weryfikacji modelowej, a uzyskane wyniki upoważniają do stwierdzenia o wysokiej pozycji rozprawy w relacji do stanu wiedzy w skali krajowej oraz umiarkowanej pozycji w relacji do stanu wiedzy w skali światowej.

5. *Czy Autor rozwiązał postawione zagadnienia, czy użył właściwej do tego metody i czy przyjęte założenia są uzasadnione?*

Autor w pełni rozwiązał postawione w tezie rozprawy zadanie stosując właściwe dla przedmiotu badań metody. Zastosowana metodologia to: (1) analiza dostępnych narzędzi do automatycznej weryfikacji modelowej systemów współbieżnych i rozproszonych; (2) aktualizacja syntaktyki i semantyki języka *Alvis* celem jej uproszczenia i wyeliminowania skomplikowanego przypadku kaskadowego zakończenia procedur agenta pasywnego w łańcuchu wywołań; (3) opracowanie i wdrożenie narzędzia, które automatycznie weryfikuje poprawność składniową projektu modelu Alvis (zgodnie z teoretycznymi zasadami rządzącymi semantyką modeli Alvisa) i generuje postać pośrednią w języku Haskell odzwierciedlającą zaprojektowany model; (4) opracowanie oraz implementacja algorytmów i struktur danych umożliwiających automatyczne generowanie LTS z postaci zapisanej w IHR; (5) opracowanie testowych, skalowalnych modeli systemów współbieżnych w celu oceny wydajności zaproponowanych algorytmów; (6) implementacja środowiska

testowego do uruchamiania proponowanych modeli testowych i różnych algorytmów generowania LTS z wiarygodnymi pomiarami ich wydajności.

Zastosowana metodologia postępowania jest moim zdaniem słuszna i okazała się efektywna. Założenia przyjęte do realizacji zadań postawionych w tezie rozprawy są również prawidłowe i merytorycznie uzasadnione.

6. *Czy Autor wykazał umiejętność poprawnego i przekonującego przedstawienia uzyskanych przez siebie wyników (zwięzłość, jasność, poprawność redakcyjna rozprawy)?*

Jak wspomniano w punkcie 2 niniejszej recenzji, rozprawa składa się z 244 stron, podzielonych na wstęp, podsumowanie, 5 rozdziałów prezentujących cel rozprawy i rozwiązujących postawioną tezę, spis treści, spis listingów, bibliografię oraz 1 załącznik. Treść pracy odpowiada tematowi określoneemu w tytule. Treść kolejnych rozdziałów jest logicznie powiązaną całością.

Struktura rozprawy jest wynikiem przyjętego planu pracy oraz podporządkowana jest stopniowemu realizowaniu przyjętych celów. Prezentowane zagadnienia tworzą harmonijną całość, dzięki czemu Autor rozprawy konsekwentnie realizuje zamierzone zadanie badawcze. Eksperymenty zostały odpowiednio zaprojektowane i metodycznie przeprowadzone, a uzyskane wyniki dobrze przeanalizowane.

Rozprawa jest napisana przekonująco i poprawnie. Zastosowana terminologia i symbole nie budzą większych zastrzeżeń. Należy jednak zauważyć, że:

- w definicji syntaktyk LTL i CTL: (1) a nie jest zdefiniowane - powinno być, że $a \in AP$; (2) nie ma również potrzeby używania ϕ_1 i ϕ_2 ;
- w semantyce LTL i CTL zastosowanie $j > i \geq 0$ w definicji operator U zmniejsza jej czytelność;
- w definicji semantyki μ - calculus dla formuł akcyjnych zamiast $Sat(\neg\alpha) = Act \setminus \{a\}$ powinno być $Sat(\neg\alpha) = A \setminus \{\alpha\}$;
- NuXmv nie jest językiem modelowania, a narzędziem do automatycznej weryfikacji modelowej.
- rysunek 3.1 nie jest oryginalnym opracowaniem autora (był już w rozprawie dr Jerzego Biernackiego), należałoby zatem wskazać źródło;
- na rys. 3.6 (b) instancje agenta B zawierają port q , który nie jest uwzględniony w jego reprezentancie rys. 3.6 (a);
- opis instrukcji `in (t) p`; oraz `in (t) p x`; jest taki sam;
- w opisie instrukcji `in (time) p {...}` zamiast *time* jest t ;
- w opisie instrukcji `in (time) p x {...}` zamiast *time* jest t ;
- w opisie instrukcji `out` naprzemiennie (bez uzasadnienia) stosowane jest *value* oraz *signal*
- w opisie instrukcji `out (time) p {...}` zamiast *time* jest t ;
- w opisie instrukcji `out (time) p x {...}` zamiast *time* jest t ;
- „`in(p)` – When a passive agent is in the *wainting* mode (...)” – zamiast *wainting* powinno być *waiting*;
- „`out(p)` – When a passive agent is in the *wainting* mode (...)” – zamiast *wainting* powinno być *waiting*;
- „*By analysing all rules from \mathbf{R} set*” - zbiór \mathbf{R} nie jest zdefiniowany;
- Definicja 3.6.2: zbiór \mathcal{T} nie jest zdefiniowany;
- „*The algorithm generating the duration function was presented in Listing 4.31.*” - ale „*Listing 4.31 depicts a generic template used by the compiler to generate the type assuming that the modelled system consists of four agents, namely FirstAgent, SecondAgent, ThirdAgent, and ForthAgent*”(str. 91).

- referencja [172]: „*M. Szpyrka, P. Matyasik, M. Wypych, J. Biernacki, and L. Podolski. Alvis Modelling Language, 2017*” nie zawiera informacji o tym, gdzie publikacja została wydana.

Pod względem edytorskim oceniana rozprawa doktorska jest napisana dobrze. Niemniej jednak Autor zapomniał o dodaniu do rozprawy tak istotnych z punktu widzenia czytelnika elementów jak: spis tabel, spis rysunków, czy też spis algorytmów. Ponadto:

- rozmiar czcionki użytej do przedstawienia algorytmów i listingów jest zdecydowania za mały, przez co utrudnia swobodne czytanie rozprawy w wersji papierowej;
- wielkość rysunku 1.2 sprawia, że jest on całkowicie nieczytelny na papierze;
- można było zadbać o to, aby Definicja 3.1.1 była umieszczona na jednej stronie zamiast na trzech;
- można było zadbać o to, aby większość listingów mieściła się na jednej stronie, np. 1.3, 1.4, 2.3, 3.2, 4.1, 4.13, 4.34, itd.
- można było zadbać o to, aby zwiększyć czytelność algorytmów poprzez ich odpowiedni zapis i rozłożenie na mniejsze bloki. Algorytm zapisany na kilku stronach jest trudny do zrozumienia i zweryfikowania, np. 4.3 i 4.4;
- brak jest konsekwencji w podpisywaniu listingów. Raz Autor stosuje opis nad listingiem, a innym razem pod listingiem;
- brak jest konsekwencji w odwoływaniu się do rysunków. Raz Autor stosuje „Fig. x.x”, a innym razem „Figure x.x”;
- wielokrotnie w rozprawie można spotkać nietypowe czerwone zakrzywione strzałki (np. str. 62, 75, 77, 79, 90, itd.).
- nietypowe i mylące odwołania do rozdziałów:
 - code layer 3.2;
 - in the enable function 4.3;
 - by the function fire 4.4;
 - itd.
- str. 86, niedokończone zdanie: „*Currently, as it was mentioned previously, the Intermediate Haskell Representation of agent’s variables(...)*”;
- wskazana byłaby profesjonalna korekta językowa, lub przynajmniej zastosowanie narzędzia typu Grammarly. Pozwoliłoby to na wyeliminowanie błędów interpunkcyjnych, stylistycznych oraz tych związanych z przedimkami *a* oraz *the*.

7. Jakie są słabe strony rozprawy i jej główne wady?

Rozprawa ma pewne istotne usterki merytoryczne, które wymagają komentarza w trakcie obrony pracy.

- (a) Autor w tezie rozprawy pisze: „*Primary goal of the thesis is to provide a tool that allows automated generation of Labelled Transition System from Alvis formal model for the untimed Alvis language version with α^0 system layer.*” W związku z powyższym nasuwa się pytanie o cel rozważania instrukcji oraz tranzycji związanych z czasem w rozdziale 3 wraz ze specjalnymi przypadkami, kiedy czasu nie ma, a następnie w rozdziałach 4 i 5 brania czasu pod uwagę w projektowanych algorytmach i strukturach danych. Czy instrukcje i tranzycje związane z czasem nie mają istotnego wpływu na czas generowania i rozmiar IHR, a następnie czas generowania i rozmiar LTS ?
- (b) przedstawiony w rozdziale 4 zbiór algorytmów implementujących funkcje **enable** oraz **fire** powinien zostać opatrzony dowodami ich poprawności, a takowe w rozprawie są pominięte. Wydaje się, że poprawność tych algorytmów jest oczywista, ale pewien komentarz byłby mile widziany, zwłaszcza, że brak jest również odniesienia do opublikowanej pracy Autora, w której takie dowody poprawności można byłoby znaleźć.

Wymienione tutaj usterki nie mają istotnego wpływu na pozytywną ocenę ogólną osiągnięć naukowych opisanych w rozprawie.

8. *Jaka jest przydatność rozprawy dla dyscypliny Informatyka Techniczna i Telekomunikacja ?*

W ocenianej rozprawie Autor zaproponował i zaimplementował zestaw wydajnych algorytmów, które przekształcają model systemu zapisany w zaktualizowanym języku Alvis do jego postaci pośredniej zapisanej w języku Haskell, a następnie generują dla niego etykietowany system przejść. Autor stworzył również środowisko testowe do uruchamiania różnych algorytmów generowania LTS w celu wykonania pomiarów ich wydajności. Realizacja tych zadań wymagała doskonałej znajomości metod formalnych (m.in., systemów tranzycyjnych, logik temporalnych), wysokich umiejętności w zakresie programowania, w tym programowania funkcyjnego, oraz znajomości zasad i technik pisania kompilatorów. Ponadto, zaproponowane algorytmy istotnie rozszerzają funkcjonalność narzędzia *Alvis* opracowanego i rozwijanego na Wydziale Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej w AGH przez zespół badawczy prof. dr hab. Marcina Szpyrki. Dlatego uważam, że realizację powyższych zadań należy uznać za istotny wkład w rozwój dyscypliny Informatyka Techniczna i Telekomunikacja, o czym świadczą również publikacje Autora na warsztatach i konferencjach o zasięgu międzynarodowym (m.in., *the 22nd International Workshop on Concurrency, Specification and Programming - 2013*; *International Conference on Artificial Intelligence and Soft Computing - 2017* [20 pkt]) oraz w czasopiśmie o zasięgu międzynarodowym (*Computing and Informatics - 20 pkt*; *IEEE Access - 100 pkt*).

9. *Konkluzja*

Ze względu na wagę podjętego w rozprawie problemu badawczego i znaczenie praktyczne, a przede wszystkim na pozytywną ocenę merytoryczną i metodologiczną, stwierdzam, że recenzowana rozprawa doktorska spełnia wymagania Ustawy stawiane rozprawom doktorskim. Wnioskuje zatem o dopuszczenie mgr inż. Michała Wypycha do dalszych etapów postępowania w przewodzie doktorskim.

dr hab. Bożena Woźna-Szcześniak, prof. UJD

